CRANFIELD INSTITUTE OF TECHNOLOGY

INNOVATION & TECHNOLOGY ASSESSMENT UNIT
(SCHOOL OF MANAGEMENT)

Ph.D. THESIS

Academic Years 1987-90

Peter D. Holden

A Multiple Perspective Approach Towards the Assessment and Development of Expert Systems in Manufacturing

VOLUME I.

Supervisors: Professor M. Cordey-Hayes
Dr. J.G. Towriss

February 1991

ABSTRACT

Current approaches to technology innovation often fail because they are conceived and assessed from a single perspective or dimension. Thus, current considerations in expert systems development are characterised by a strong focus upon the technology and technical issues without a prior process of wider appraisal and technology assessment. A central theme of this study is that the business, organisational and human factors, which determine how effectively the technology will be used in practice, must be an integral part of the assessment process. The thesis describes a 'multiple perspective approach' to technology assessment applied to expert systems innovation in a large manufacturing organisation.

This research therefore embraces detailed technical, organisational and individual perspectives of expert systems assessment and development and describes how each perspective adds new concepts, methods and tools. In practice, this has meant modelling activities and information flows in a two-site manufacturing organisation, the identification of a variety of potential areas for expert systems development, the narrowing down and selection of particular areas according to technical, organisational, business and personal criteria, and the eventual design, development, 'operationalisation' and evaluation of a single application. This study is placed in a wider context by complementary analyses of other manufacturing users and suppliers of expert systems. The work aims to contribute towards an understanding of expert systems innovation and to improved methodologies for technology assessment and technology transfer.

Acknowledgements

To Professor Cordey-Hayes for his innate ability of unveiling the simplicity in even the most complex; to Roger Seaton for his freshness of thought; and to John Towriss for his technical assistance. I would like to thank all members of the Total Technology Panel that steered me a path over the two years, and to all my friends and colleagues in the Client Company.

Particular thanks go to Martyn for the vision of 'INTA', of which I am grateful to be a part, and to all students past and present for making it what it is now.

Finally, my deepest thanks go to my parents for their unceasing love and support, and to other special people in my life who have made all things possible.

LIST OF CONTENTS

continued/...

Chapter Five: Organisational Modelling and the Role of IDEFo

Chapter Six: Determining Appropriate Expert Systems Project: From Problem Identification to Application Selection

continued/...

continued/...

Chapter Eight: Enlarging the Context of Expert Systems Evaluation

Chapter Nine: Summary, Conclusions and Further Research

LIST OF FIGURES

(continued /..)

LIST OF TABLES

List of Appendices in Volume II.

Chapter 1.

Introduction: Motivation and Nature of the Research

1.1. Backcloth to the Study

The focus of research in expert systems is very much orientated towards the merits of the technology and future capabilities, especially in its associations with artificial intelligence. This in itself has proved damaging because it leads to false expectations over what the technology is and how it might be exploited in manufacturing. Moreover, current analysis of the actual use of expert systems is limiting because it is undertaken from the point of view of the supplier (whether supplier of products or information), therefore technical issues tend to predominate. The limited research which shows how user organisations are making use of Expert Systems (or 'ES') reveals a number of organisational and human problems; however, there is little rationale or explanation of how and why such problems arise. Where such information is available, through industry experiences, there is a reluctance by organisations to share this for reasons of commercial confidentiality. Thus, there has been no comprehensive analysis which defines the determinants of successful ES innovation, nor has this been consolidated to discern an accepted approach or 'best-practice' method. Although some studies have defined a number of critical success factors of ES development, such studies are often associated with very large 'show-case' or pioneering developments which are not representative of the types of system actually being developed in manufacturing. One conclusion which may be drawn from reported developments however, is that the most important problems experienced in development and implementation were of an organisational and personal nature rather than of a technical one, or through generic deficiencies in the technology itself.

a) Technology Assessment

It is recognised that expert systems technology has ingredients to provide a number of organisational benefits, such as improved competitive advantage or increased effectiveness and efficiency at all levels in a company. However, manufacturing organisations have had difficulty in exploiting this potential. One reason is that there is uncertainty over how to select appropriate problems for development in the first instance. A critical first stage in expert systems innovation therefore, must be in defining methods which allow these companies to assess the business and organisational potential of the technology. Clearly it is important to assess whether applications are technically feasible, however, as with many other technology developments, assessment is wholly biased towards technical issues alone: this takes the focus of assessment towards specific design and development issues and away from an initial understanding of organisational needs. In many cases therefore, technology assessment is described as a 'technology fitting' exercise rather than one which is shaped by the organisation .

For these reasons, literature which describes development experiences from a user-organisation's perspective shows a confusion over what the technology is and how it might be exploited. It also a vindication of the chasm that has emerged between state-of-the-art research and development at a supply level on the one hand, and the actual diffusion and utilization of the technology, and associated knowledge and ideas, on the other. Indeed, the take up in manufacturing has been very low, not because the technology does not offer competitive advantage or the prospects of 'adding-value' to company operations, but because organisations are having genuine difficulty in applying the technology into organisational settings. This has implications at both macro and micro levels in the technology transfer process.

At a macro level, it is evident that the structure of the ES supply industry is orientated towards the marketing and presentation of the technology, whilst overlooking 'delivery' issues of how it might be applied within organisations, thus addressing the processes of technology transfer beyond the sales and service level. The market still remains strongly product orientated with associated services biased towards the technical development of these products and not the evaluation of their organisational potential. However, it is not just in the supply of technology from vendors where problems are experienced, but in the supply of knowledge on the methods of assessment and development from the research community. Criticism may be directed at this quarter for a concentration of research almost exclusively upon the development of the technology, tools for knowledge acquisition, tools for programming, new methods of validation for instance, whilst failing to address the 'delivery issues' of how the technology is assessed, how the business value of expert systems may be evaluated, how technology assessment and development should be organised in an organisation and how these systems should be costed and maintained. Indeed methods which claim to do this have a distinct 'scientific' flavour without showing a realistic approach to the problems of how they can be applied within time and budget constraints in a commercial manufacturing organisation.

At a micro level too, there are indications that manufacturing organisations are failing to undertake initial technology assessment as a formal organisational programme, thereby linking evaluation to business and company needs in a top-down process of enquiry. Instead, the initiative is being taken by the 'hobbyist' who is interested in the technology and will apply it in an uncoordinated way in order to solve local (i.e. departmental or functional) problems in a characteristically bottom-up, technology driven manner. This also reflects why certain problems are being experienced in development, such as failure to gain management support, failure to gain interest in the technology and a failure to develop systems beyond the prototype stage.

There is a distinct lack of research which addresses *how* expert systems are actually being used in manufacturing other than the 'show case' applications and experimental projects which do not mirror the reality of most commercial developments. This is because development is conceived as a technological innovation process and not as an organisational innovation also, in which social, cultural, business and human processes of change determine how effectively the technology is used in the company. Therefore in considering an approach towards expert systems innovation, the tools and processes used must, necessarily, not only be multi-disciplinary but also multiple perspective ; the latter indicates *how* they are used and the development context in which they are applied, whilst the former merely indicates in which field of study they lie.

Such sensitivity towards the development context requires an implicit understanding of the organisation's culture, values and needs at the stage of problem identification rather than at the design level. Again however, this raises further research questions as to how this might be achieved ? In other areas of study, knowledge of the organisation has been attained by modelling it in some way according to the specific objectives of the modelling exercise. For instance soft approaches have been used to define organisational problems and reach a consensus between individuals in a group. By contrast, at lower levels in the organisation, computing disciplines apply entity models to define the information and data requirements of a process. Thus a hierarchy of modelling techniques may be identified spanning the abstract level of thoughts and perceptions through to a highly defined, low level series of data relationships. Despite the range of approaches, modelling in the expert systems field is a design and development activity used most frequently during knowledge acquisition and representation tasks rather than at earlier stages of technology assessment and problem conceptualisation where modelling the organisation may be of the greatest benefit. There is the necessity of research to consider how useful modelling techniques are in defining the organisational potential for expert systems, by mapping functions and business activities for example where ES technology is both feasible and appropriate.

A final stage of technology assessment is estimating the costs and potential benefits of proposed applications as part of the selection process. However, there is little work on how expert systems should be costed systematically, whether current business and economic tools are appropriate in the case of expert systems or whether new costing models are required. A popular justification for expert systems is that they 'add-value' to organisational activities: however, again, there has been little analysis on precisely how this is achieved and therefore how development justifications should be prepared, especially since it is more than likely that they will operate in organisations which adopt traditional accounting models in defining costs and benefits. It is also important that developers are aware of the sensitivity of costs and benefits to changes in assumptions: here, there are a great number of uncertainties. The following provide some indication of the estimating and quantification difficulties faced by organisations:-

i) What is the cost of expertise to the company ?,
ii) How much will maintenance cost a year ?
iii) What is the accounting lifetime of an expert system ?
iv) Should a company allocate contingencies because of the uncertainties of prototyping ?
v) How does a company define development effort ?
vi) What performance criteria should be used ?
vii) How does a company measure the knowledge engineering requirements ?

The distinct lack of feed-back on experiental or 'process' issues such as these arise through company secrecy, described earlier, but also because the feed-back mechanisms from user organisations to those that define methods and undertake research are not in place.

b) Development Methods and Tools

As a new technology, expert systems have gone through the now established innovation life-cycle of initial hype and overstated claims over its capabilities, to the present state of realism in which the market has at last begun to downplay expert systems as being 'artificially intelligent' systems and adopt a 'mainstream' strategy of marketing the technology as another useful computing tool. Although there are similarities in design and development approach between expert systems and existing information technologies, there are also important differences. However, no clear practice of expert systems development as 'best practice guide-lines' or methodologies have emerged and it remains the case that user organisations are struggling between two extremes. At one end is the simple unstructured notion of a development life-cycle which emerged from early experiences in developing experimental systems. These were characterised by being developed in an experimental and research environment. Such lifecycles provide no assistance in how to undertake development and lacked the detail to plan and manage expert system projects. At the other extremes are large, bureaucratic methodologies which impose a discipline of development but require substantial resources to implement and are therefore targeted beyond the current level of commitment and innovation which is currently experienced in manufacturing. Furthermore, they are highly prescriptive and provide little scope for the user organisation to craft the methodology about its specific needs and settings.

Industry clearly requires a middle ground which allows organisations to apply tools, often those which are already in use in the company, in a way which fulfils task requirements and technical feasibility, but also is complementary to the development context and organisational situation. As with the technology itself, development methods and tools both affect and are affected by the organisational characteristics, working practices and culture of the company in which they are applied. Thus in order to take account of the specific needs of an organisation, methods and tools should mirror these. This requires that tools are flexible enough to be applied in different ways. It also suggests that there can be no normative method of expert systems development but rather an approach emerges which is shaped by the development context and characteristics of the organisation. To attain this level of understanding requires that research effort should focus upon providing an eclectic and multidimensional 'self-assessment framework' in order to allow organisations to determine for themselves the most appropriate blend of formal and informal tools and methods.

If development is to be driven by organisational needs rather than technology itself, it is important to understand how organisational and human factors tools which have been applied in other fields should be applied to expert systems specifically. For example, Mumford describes how participation techniques were used in a very large, pioneering expert systems project (1989). How then can these skills, and those deriving from other socio-technical concepts, be transferred to the 'layman'? Bearing in mind that such a person is most likely to be a technologist, rather than an organisational specialist, and operating within severe cost and time restraints as well as under less overt cultural and political forces . This is a significant challenge, and yet, has received only limited coverage in the expert systems field.

c) Technology Evaluation

As well as a neglect of pre-development issues, and consideration to the processes of development, a final shortcoming of the present situation is that there is a disregard for post-implementation issues such as testing, validation, maintenance and evaluation of an application. Very little is known about maintenance costs and needs for instance, despite being an over-riding factor which determines the economic viability of an application over its lifetime.

The few available studies which look at how systems have been implemented and used shows repeatedly that barriers to implementation were 'infrastructural', in other words concerned with how a company organised and managed the innovation and technology transfer process, rather than through specific shortcomings of the technology itself. The implication is that during evaluation, it is not possible to bound the situation to the level of the technology or the project group, but that external and internal influences and pressures at any stage in the transfer process, from initial negotiation with the technology supplier through to end-user acceptability , will affect the outcome. Despite this, the evaluation of expert systems takes place at a product level, often in terms of technical viability, occasionally in terms of economic payback, but very seldom in social, human and organisational terms.

1.2. Study Objectives

In response to this backcloth of research needs, this study represents a multiple perspective analysis to consider how manufacturing organisations should go about introducing and exploiting expert systems technology. This approach brings together a diverse range of disciplines and concepts previously unrelated to the expert systems field. However underpinning the study are three principal themes: re-defining the conceptual basis of technology *assessment*; providing a methodological framework for the *development* and implementation of expert systems; and providing an assessment framework for expert systems *evaluation*. Furthermore, it adopts two levels of research: at an external level, it makes use of surveys to investigate the extent of the use of expert systems in manufacturing organisations, the assessment and development approaches utilised and operational experiences, and the problems and successes encountered. It also adopts an internal focus in which the author spent two years in a manufacturing company defining a framework for the assessment and development of expert systems, culminating in the design and implementation of a full-scale operational system. Details of this arrangement are outlined in the next section. It is intended that both levels of analysis will help to fill the gap in the literature, indicating how 'real life' requirements (such as experts' needs, company politics, budget and time) influence the development process and also to demonstrate the importance and value of multiple perspective analysis in expert systems innovation. It is also intended to address some of the research questions which have been generated by viewing the innovation process from organisational and personal perspectives. These define a series of study objectives which may be expressed as research questions:-

i) what is the overall level of use and impact of expert systems upon manufacturing organisations ?

ii) how is the organisational and business value of expert systems defined and assessed ?

iii) how is assessment and development actually undertaken and what methods and tools are used ?

iv) what are the problems and barriers in the innovation process ?

v) how do organisations manage the process of change ?

vi) how should expert systems be evaluated ?

As well as these main objectives, the study has also generated a further set of research questions which derive from consideration of the processes of research: for instance, how does a person brought into an organisation rapidly identify its problems, needs and culture as a precursor to technology assessment ? There was also a third set of practical tasks and deliverables which were required by the client organisation. To resolve each set of questions, requires a balanced approach in the process of investigation. Different perspectives of the same problem are valuable because they not only identify the full range of issues that need to be addressed, but also because each embodies a theory of what causes the problem and what needs to be done to prevent or correct it.

Although this is not a purely technical study as such, it does have a significant technical component, raising a question of to whom is this study directed ? As a broadly based study, it is relevant to readers of diverse disciplines who are interested in expert systems methods and approaches, and more generally technology transfer and innovation, the management of change, organisational assessment and business planning, and the human and social effects of technology. Although it is assumed that the reader has some understanding of expert systems concepts, this is not essential since definitions and an introduction to the technology and its uses are provided in the main text and covered in greater detail in Volume II.

1.3. Industrial Collaboration

As well defining the basic themes to this study, it is also important to outline the nature of the research. The author undertook a three year Total Technology Doctoral Programme based at Cranfield Institute of Technology. This had two practical implications: firstly, the doctorate was required to be interdisciplinary and therefore the study was undertaken in a wider context than might be expected for a 'conventional' Ph.D. Secondly, the study was based on a problem of direct relevance to a sponsoring industrial company and the author spent the first two years within this organisation. This also had secondary implications in that the direction of research was often constrained by the work requirements and organisational settings of the client company. The author was also expected to attend short courses on finance, marketing, and management. Furthermore, as a technology which was new to the author, time was allowed to attend technical courses specifically on expert systems and participate in a series of lectures on the subject within the KBS school at Cranfield. Assessment and steering of the study during this period in the client organisation was managed by a panel group of inter-faculty members at Cranfield together with the technical director and company computing manager of the company.

The Total Technology scheme was initiated as a broadly based research degree to cover all stages of the technological process in industry, its interrelationships and management. The focus of analysis therefore is problem-centred rather than based on a single discipline or function. A consequence of this is that the structure of the thesis does not have a single theme or assessment, but instead pursues a course of generality in which the objective is to achieve a well balanced syntheses of all functions, disciplines and perspectives. Furthermore, where it is necessary to describe a research activity in detail, much of this work, though relevant to the study, has been appended.

Chapter 2.

Research Overview and the Positioning of Expert Systems Literature

2.1 Introduction

Where the last chapter provided a 'back-cloth' to this study and the organisation of work through the Total Technology Doctoral Programme, this chapter looks in more detail at the study's components and themes. It begins therefore by giving an account of the thesis structure and provides a breakdown of each of the chapters and how they are related to the three principal themes of technology assessment, technology development and technology evaluation. As a multiple-perspective study, chapters inevitably call upon a diverse range of literatures, both within the expert systems field and also in other areas previously unassociated with this technology. In order to retain clarity and direction in the study therefore, analysis of these literatures is apportioned to those chapters which cover the respective issues in detail. Thus for example, an analysis of Expert System (ES) development methodologies is discussed in Chapter 4, and a review of models of technology transfer in Chapter 8. This study also makes significant use of Appendices, all of which are located in Volume II.

Common to the three themes of assessment, development and evaluation is the technology itself, therefore following a mapping of the study, this chapter seeks to provide a working definition of 'expert systems'. As an empirical study, greater importance is attached to definitions which describe precisely how this technology is being used in industry rather than reflecting current research ideas and expectations. In doing this, it is necessary to distinguish expert systems from 'artificial intelligence' at one extreme and conventional or traditional programming at the other.

From this basis, this chapter provides an overview of the extant literature on ES. It should be reiterated that the purpose of this chapter is not to provide an exhaustive 'literature review', as a multiple-perspective study this is not feasible, but rather to provide a framework for cataloguing different types of literature and research viewpoints which have emerged since the early 'pioneering' expert systems and thus place this study within a research context.

Although this study centres upon an analysis of expert systems specifically, it is a strongly held belief that most of the concepts and tools used in this study (such as the conceptual model in chapter 3, the development framework in chapter 4, organisational modelling in chapter 5, and technology transfer in chapter 8) are valuable irrespective of the technology or organisational settings. For this reason, there has been a tendency to place the more esoteric detail of expert systems development and company specific activities in the Appendices in Volume II. It should be noted however, that they are of central importance and direct relevance to this particular study.

2.2. A Mapping of This Study

An overview of the structure of the study and a mapping of primary research issues is given in Figure 2.1. It shows that the study begins with the question 'how should we view technology assessment ? ' To answer this first requires an understanding of what the technology is, and more importantly, what user-organisations understand it to be. Thus Section 2.3. of this chapter provides a working definition of expert systems and Section 2.4. evaluates reported experiences in dealing with the technology. Chapter 2 identifies a strong bias in research and professional literature towards expert systems technology, tools and techniques and their potential

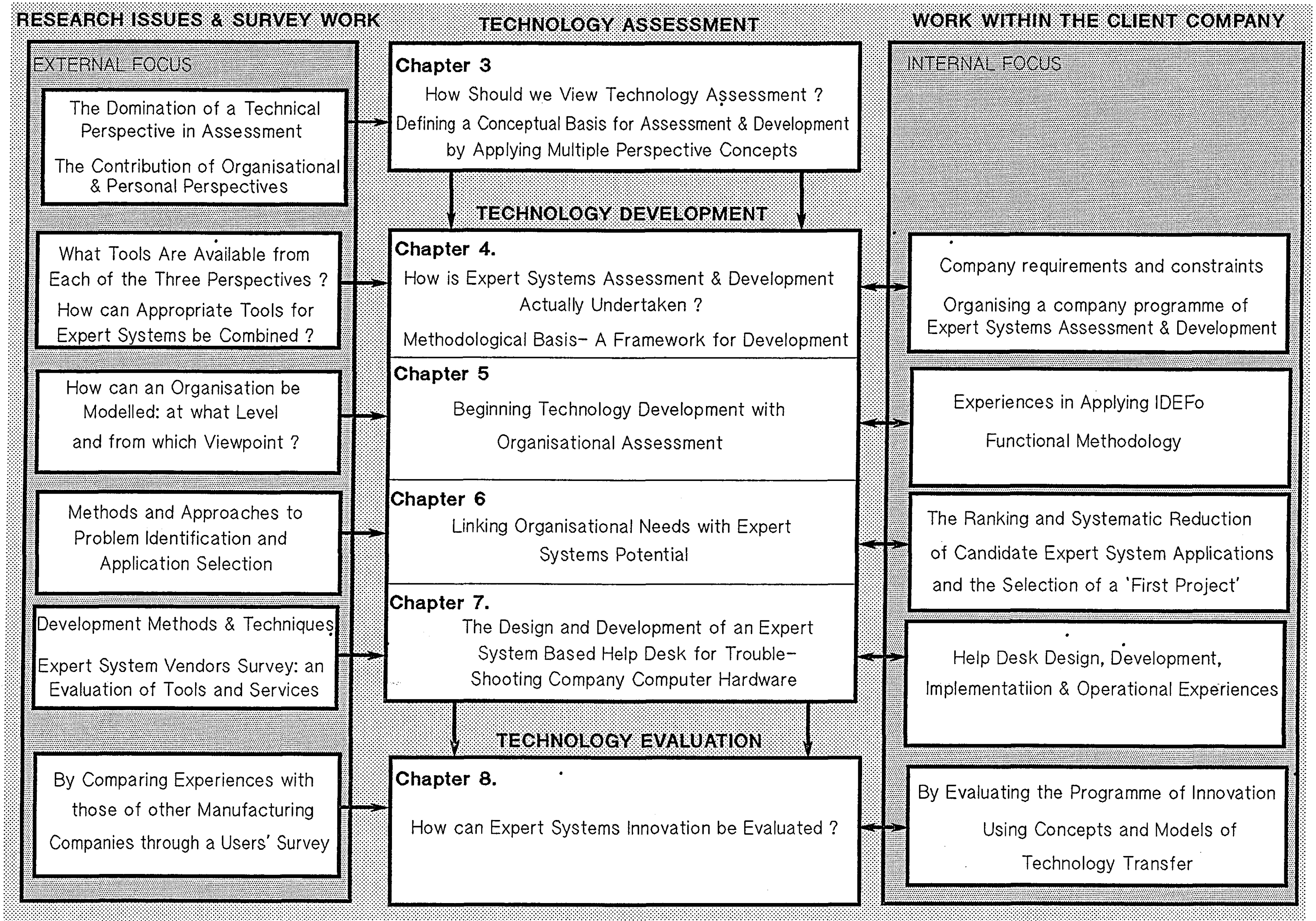


Figure 2.1 Structure of the Study and Mapping of Research Issues

capabilities, with little analysis of the actual processes of assessment and development which determine, ultimately, how successfully the technology will be in a company. This issue is explored further in **Chapter 3**, which shows that current approaches towards technology assessment operate within the constraints of a one dimensional, technology focus which, though necessary, is not sufficient in describing the total process of change. Therefore Chapter 3 calls for a re-assessment of the problem from not just a technical perspective, but from organisational and personal perspectives. In viewing expert systems innovation from an organisational perspective for example, the question becomes how can manufacturing companies go about assessing the organisational value and need for expert systems technology rather than the pervading technical focus upon 'where in the organisation can expert systems be applied' ("or a solution looking for a problem"). By considering organisational and personal perspectives, new concepts, values and disciplines are added to the assessment process. As with a technical perspective though, Organisational and Personal perspectives also have their limitations and for a complete analysis, it is necessary to mix these perspectives in some way. Chapter 3 concludes by describing an applied use of Multiple Perspective Concepts (Linstone:1981) as a means of combining perspectives in a way which reflects the relative importance of each at any given stage of assessment or development. The idea of combining perspectives also gives rise to the use of different processes of investigation, from the formal and 'hard' elements of a technical perspective, to the 'soft' and implicit elements of a personal perspective, for instance.

Chapter 4, as Figure 2.1. shows, progresses from a conceptual level to a methodological level by explaining how methods, tools and processes derived from each of the three main perspectives, technical , organisational, and personal, may be of use in the assessment and development of Expert Systems (ES). Current methods of ES development fall between two extremes: the first states that expert systems are unique and distinct from other computing approaches and therefore require new development concepts such as iterative prototyping and evolutionary design. By contrast the second 'school of thought' argues that expert systems are just another computer software and therefore many of the tools of software engineering may be applied. Chapter 4 adopts a central view which states that both models provide useful methods and should be combined in a 'hybrid approach'. Although this offers considerable scope for improvement at a design and developmental level, such enhancements are limiting in the sense that they focus upon technical issues only and fail to consider pre-development issues such as problem selection, business planning and technology transfer which can only be understood from an organisational perspective. Similarly, Individual perspectives define how technical designs should be shaped according to human needs and how the uncertainty during the processes of development may be reduced by involving people affected by the design to participate in shaping its outcome. Where MPC was used in Chapter 3 to combine perspectives, Chapter 4 concludes by suggesting a development framework by which each of the different processes of assessment and development may be combined. The development framework assumes three things:-

> i) that the choice of tool or process at any given stage in the development life-cycle is based upon its suitability to the development context as well as its ability to perform a specific function,
>
> ii) that care must be taken to consider how the tool will be applied (formally / informally; hard process/soft process for instance), according to the development context.
>
> iii) that it is necessary to understand how the organisation operates and appreciate its problems, culture and needs for the framework to be used correctly.

The development framework thus provides a means by which to combine tools and approaches which have their conceptual roots at each of the three main perspectives in a way which is central to the application context. The necessity of understanding the 'emergent properties' of the organisation suggests that it is appropriate, especially from the author's point of view as an 'outsider', to model the organisation in some way. The value of modelling is outlined, but as Figure 2.1. notes, the difficulty is in deciding at what level in the organisation this should take place and from which viewpoint ? Examples are given of business modelling down to very detailed information and entity modelling based around the specific data needs of information systems. From an analysis of current modelling approaches, five organisational viewpoints are identified, all of which are potentially useful at some stage in the development life-cycle. **Chapter 5** therefore considers ways in which each of these viewpoints may be combined, if at all, within a single modelling technique. It is concluded that for this to be possible, firstly, the modelling process should be functionally (i.e. activity) based; and secondly, that it should be flexible enough to be applied both formally and informally in order to accommodate each of the five viewpoints satisfactorily. A short-list of functional models was made from which IDEFo, a functional methodology, was chosen and applied in the client organisation. Consistent with its use in the company, the model and the modelling process were evaluated in both formal and informal terms.

The modelling exercise of Chapter 5 provided an understanding of the organisation and its problems and needs. This is a useful exercise whatever its intended role. **Chapter 6**, by contrast is concerned specifically with how the client organisation should go about assessing the potential contribution that expert systems could make in resolving these problems. Critical to this process is the ability to identify a set of problems whose characteristics matched those of the technology: but also to provide some measure of the organisational, personal and business value of the proposed application. Current approaches to ES selection however, concentrate upon almost exclusively upon issues of technology feasibility. It was therefore necessary to place the selection process within a wider context of assessment; IDEFo was used along with a number of business tools in order to rate application ideas according to their strategic value and organisational impact as well as resource requirements.

A second, more detailed analysis is undertaken in Chapter 6 in order to reduce candidate applications down to three. Central to this filtering process was the need to define 'first project' criteria in which applications had to satisfy specific organisational constraints, such as low risk, low organisational impact, high exposure etc., arising from the fact that this would be the first expert system to be developed in the organisation. Evaluation prototypes were also built from which the decision was made to construct a fully operational expert system based help-desk for trouble-shooting computer hardware faults. The evaluation prototype provided a significant amount of information, as did earlier processes of technology assessment, and these findings were consolidated using a development suitability check-list to provide a 'requirements specification'. The philosophy behind the use of the check-list was that although it is not possible to specify all development needs beforehand, a great deal of information may still be acquired and used in the planning, design and implementation of the expert system.

This information could also be used, as **Chapter 7.** shows in providing performance standards, such as project phase completion targets and cost estimates, by which to constrain the prototyping element of development. Chapter 7 itself is divided into two parts: Part I looks at all pre-development issues and is therefore an extension of the requirements specification generated in Chapter 6. Particular emphasis has been placed upon justification and costing of the help desk since this is an area least well understood and covered by the literature. The output to Part I is a functional design and outline of desirable features, together with project planning guide-lines, cost and performance estimates and a mapping of individual responsibilities and scope for participation at various stages of development.

From this, Part II of Chapter 7 looks at the actual development, implementation and eventual evaluation of the completed project. Significant technical detail, as before, has been appended. Midway through the development process it became evident that a number of the time estimates and design features had been too ambitious and therefore two significant changes were made to the design: firstly, the scope of the project was reduced to what was considered achievable in the remaining time; and secondly, the organisational role for the help-desk was simplified. The structure and design of the revised system are described from which a more detailed account of the operations of the help-desk is given. Following testing and validation, this chapter describes the implementation and operational experiences in using the help-desk over the first six months. The chapter concludes by providing a cost-benefit analysis, comparing initial costs and predicted benefits with those actually achieved, and also evaluating its organisational and personal effects.

Evaluation at this level is limiting because it looks specifically at the success of a particular end-product rather than the total processes of innovation and the framework of assessment and evaluation that were constructed. This study concludes therefore by asking the question in **Chapter 8**, 'how should experts systems innovation be evaluated ? ' Two approaches are adopted in this study. The first, as *Figure 2.1.* shows, is to compare the approach and results of assessment and development experienced in the client company with those of other manufacturing organisations. Few insights could be gained from the literature and therefore the author collaborated in a European survey of manufacturing users who were at some stage in the development life-cycle, whether it was initial problem conceptualisation or the operation of a completed system. This provided a great deal of information about how other companies had undertaken development: both similarities and fundamental differences in approach were noted.

A second stage of evaluation was to place experiences in the client organisation within a framework of technology transfer and knowledge diffusion. This made it possible to evaluate the author's own role as 'transfer agent' and also that of the organisation in terms of how 'receptive' it was to expert system ideas and to the help-desk application itself. In thinking in these terms, coupled with the main findings of the user survey, Chapter 8 concludes by defining a model of knowledge transfer for expert systems innovation. An outline of its proposed use is given, particularly how it might be used to impress upon organisations the importance of attending to issues of 'delivery' (i.e. how can knowledge transfer take place in accordance to company needs and what are the appropriate processes and mechanisms of transfer ?), rather the present structure of the ES industry which is based upon the marketing and presentation of ES technology and knowledge.

2.3 Understanding Expert Systems

It is not the aim of this section to provide a detailed account of the technology, in terms of design, construction and alternatives for example, as this is done elsewhere (see the technical reference to the technology in Appendix I and subject headings in other chapters). Rather, it's aim is first to provide a working definition of Expert Systems (ES) based on how they are being used; second, to distinguish between ES, Artificial Intelligence (AI) and Knowledge Based Systems (KBS). Figure 2.2 shows the relationship between AI, KBS and ES. The terms are all in use and often interdependently. However there are important distinctions between them. Finally, it should be mentioned how ES and conventional computer programs differ.

2.3.1. What are Expert Systems ?

Berkins (1986) suggests that there are important misconceptions in industry over precisely what expert systems are and are not. Moreover, he argues that if there is a problem in understanding the nature of the concept, then there is the possibility that any such use of the technology will result in disappointment. This view tends to be verified by the survey findings in Chapter 8 which reveal that a failure, of management particularly, to understand the technology and its capabilities actually prevented implementation. Yet, as D'Agapeyeff and Hawkins (1988) point out, 'there is no commonly accepted definition of expert systems. Indeed, a standard definition is now a forlorn hope'. These authors, conclude that given the wide scope in definitions and viewpoints from which these definitions were made, expert systems appear to 'mean whatever anyone chooses they should mean'.(p189).

Despite this view, a collection of some of the many definitions was gathered in the hope that commonalities could be identified. A useful division of definitions was provided by Pederson who distinguishes between ES by 'what they do' from 'how they do it'(1989). Defined on a what they do basis, expert systems are computer programs which :-

> a) captures expert's performance and duplicate it in a chosen area (Harmon & King: 1985),
>
> b) emulates an experts problem solving, decision-making or reasoning processes (Feigenbaum *et al*: 1988)
>
> c) addresses problems which are demanding enough to require human expertise (SEAI :1988)
>
> d) advise, analyse, categorise, communicate, consult, design, diagnose, explore, forecast, form concepts, identify, interpret, justify, manage, monitor, plan, present, retrieve, test and tutor (Michaelson *et al*: 1985)

Other authors choose to define ES on a how-they-do-it basis:-

> a) uses complex inferential reasoning to perform tasks which a human expert could do (Welbank: 1983),
>
> b) using a computer model of heuristics and facts in order to reach the same conclusions as the expert (Alty and Coombs:1984),
>
> c) symbolically represent expert knowledge to attain high levels of performance in narrow problem areas (Waterman:1986),
>
> d) 'allow modifications to include new knowledge and new contexts of application, and have the ability to explain and justify its actions or line of reasoning' (Buchanan :1985),
>
> e) makes use of declarative and symbolic programming rather than normal computing's focus upon procedural programming (d'Agepeyeff and Hawkins: 1987),
>
> f) 'a collection of 'IF-THEN' rules for drawing inferences: IF such -and-such is true, THEN assume that so-and-so is true. By way of these rules, the program embodies some of the theoretical knowledge and "rules-of-thumb" used by human experts' (Boden: 1989).

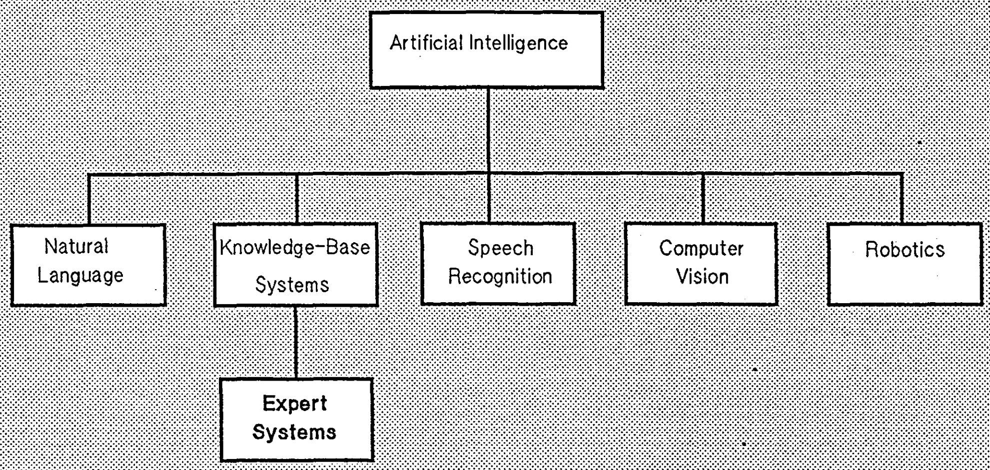
From the foregoing, a picture of the characteristics of an expert system emerge. The above list is not taken as rigidly defining an expert system because no such definition exists; however it does reflect how they are used and is flexible enough to accommodate variances. For example, a program developed using conventional programs like 'C' or Fortran can constitute an expert system if it is not analogue, that is, the knowledge base is separated from the means of controlling the program. From these definitions, it is also possible to state what expert systems are **not**. Firstly, they are not general problem solvers; nor are they intelligent or 'clever' because they are restricted to providing knowledge which has already been programmed into the knowledge-base. Thus, as Boden (1989) notes they have no 'non-monotonic' reasoning capabilities, in other words if the system is programmed to assume a statement is true and it happens to be false, it cannot adapt or change accordingly. Similarly, 'true' remains so until it is manually reprogrammed and ES cannot adapt reasoning to a new context which is analogous to the old one (sometimes called 'common-sense' reasoning or reasoning by analogy - *see Appendix I for details on the construction of ES*). In short, expert systems can only do a few of the things that human experts can do. Therefore, as Gill (1986) states, they should not replace human beings in the sense of making them no longer necessary .The latter is a qualitative proviso which most commentators now seem to affix to their definitions.

2.3.2. The Differences Between Artificial Intelligence and Expert Systems

Artificial Intelligence, refers to research since the 1940s which look towards simulating the actions of the human brain and thereby creating a 'synthetic intelligence' (Pople:1977). In order to do this, it was also necessary to replicate the sensory capabilities of humans such that research work in vision systems, speech recognition and natural language interfacing, robotics and knowledge manipulation necessarily accompanied studies of cognitive science. AI is thus a generic name for all these research activities. Early research in AI looked towards creating a 'General Problem Solver' which combined symbolic rather than real world reasoning with learning systems in order to simulate human thought (Ernst and Newell: 1969). Current work on AI is less ambitious and in terms of knowledge processing, has tended to concentrate upon specific advanced methods of representation and inferencing such as neural connectivity (Judd:1990), non-monotonic or non-standard methods of reasoning (Ginsberg; 1987) and object orientation (Stefik and Bobrow:1986). Much of this work has been made more possible through the improved capabilities of computer hardware and use of new computing techniques such as parallel processing and blackboard architectures (see Appendix I). Although there are examples of commercial knowledge, robotics and vision systems, these are subsets, and Artificial Intelligence *per se* remains very much a research issue.

Knowledge-Based Systems(KBS) and Expert Systems from *Figure 2.2.* can be seen to be only one ingredient of AI. Where AI concentrates on creating systems with general problem-solving capabilities so, like humans, they can develop broad classes of problems, AI methods and techniques were modified in the 1970's and used in more specialised programs which addressed specific rather than general problem domains. These were called 'expert systems'. Rather than develop complete problem-solving programs, expert systems adopt specific approaches to *representation* which express knowledge or human expertise in a systematic way; and 'search' or inferencing techniques which allow the program to find the most efficient path through knowledge in the system in order to resolve a problem. Bramer argues that Expert Systems (ES) are not 'intelligent' systems because they are unable to employ techniques which improve their performance, such as learning (1984). A further distinction between the two according to Bramer is that intelligent systems are able to recognise patterns associations and relationships from unstructured and unconnected items of knowledge. By contrast, knowledge in ES has to be defined and organised in a predetermined way.

Figure 2.2: The Relationship Between AI, KBS & ES.



2.3.3. Knowledge-Base Systems or Expert Systems ?

Although most commentators use these two terms interdependently, there is some debate over whether the term expert systems actually represents the developments taking place. At a systems level, Wiig (1990) notes that KBS tend to include explicit knowledge about some domain; this knowledge may be high-level or 'shallow' knowledge from an expert or from more procedural knowledge and information sources. By contrast, ES are a sub-set of KBS which deal exclusively with expert domain knowledge.

Winograd and Flores (1989) argue against the use of the term 'expert systems' at a human level because they claim that it misrepresents what the technology is actually capable of achieving. As they put it: 'There is a danger inherent in the label "Expert Systems." When we talk of a human expert we connote someone whose depth of understanding serves not only to solve specific well-formulated problems, but also to put them into a larger context. We distinguish between experts and idiot savants. Calling a program an "expert" is misleading in exactly the same ways as calling it "intelligent" or saying it "understands"....This can lead to inappropriate expectations by those who attempt to use them' (p132.). Dreyfus and Dreyfus (1986) argue similarly that expertise has both explicit and tacit forms of knowledge with only the former being possible to represent in any computer system.

For the above reasons, the term knowledge based systems is often used instead of expert systems to focus attention on the knowledge the systems carry, rather than the question of whether or not such knowledge constitutes 'expertise' (Davis:1986). However, in retaining an empirical focus, d'Agepeyeff and Hawkins note that those systems that are actually in use in manufacturing are 'expert systems' rather than KBS for the reason that most systems in use are 'in the form of rules articulated by the experts or through interviews about their skills' (1988.p188.). This view is endorsed by Bobrow *et al* (1988) who note from experience that KBS have flexibility in the use of knowledge (through integration and interfacing or on-line feed-back, maintenance front-ends etc.). By contrast, ES have built-in commitments as to how the knowledge embedded in them is to be used, in most cases this is because of the limited input/output behaviour of the tool rather than the knowledge it contains. These authors therefore achieve a satisfactory 'middle ground' in using the term

'knowledge-based expert systems'; these have some of the characteristics of KBS, namely the focus upon the explicit representation of knowledge, whilst achieving expert-level performance in specified areas. For the purposes of brevity they are referred to simply as **expert systems** in the rest of this study.

2.3.4. The Differences between ES and Conventional Systems

An expert systems can differ from more conventional computer programs in a number of ways. Before these are discussed though, the rejoinder of Sviokla (1986) should be noted that the traditional divisions between the two are less apparent as continual advancements in 'conventional' programming are made and tools are developed which combine both of these computer programming approaches.

a) Symbolic versus Algorithmic Processing

Expert systems make use of *symbolic processing* in order to model non-mathematical and usually ill-structured problems. 'Symbols' are names which designate a process or concept: in ES, these are facts about the problem; routines which provide for a problem solution; and interpretations of the solution. Symbols usually take the form of expressions which, when combined, can be used to define more complex objects or concepts.

By contrast, traditional software programming methods are based on mathematics and statistics and are better suited to solve well structured, non-symbolic tasks which aim to convert known procedures into code. Thus, the techniques used in traditional programming are called *algorithmic* because there exists computational routines (algorithms) that permit a solution to be found in actual numeric terms and specified in well-defined terms, e.g. maximisation of profit, optimisation of efficiency, and so on. For ill-structured problems, algorithmic methods are either inappropriate or attempt to find structure by rigourously testing each possible combination and permutation of the problem. For example, in order to locate a fault in an integrated circuit, an algorithmic program would have to check every individual circuit, component and their respective interactions. Furthermore, the program would have to distinguish between symptoms and the root cause of the fault. Although this may be feasible, it would require excessive amounts of computing power and time in order to provide a solution. Where the 'decision-space' or size of the domain is large, as with the above example, the number of fault possibilities may be so great that the scale of the problem may escalate to fantastic proportions - this is called a 'combinatorial explosion'.

By using an expert systems approach, such problems may be resolved more easily because symbolic methods allow facts and expressions of problem solving itself to be declared explicitly. These expressions mainly take the form of 'heuristics' or 'rules-of-thumb': these are simply short-cuts in solving problems which evolve when an individual gains experience in a particular domain and becomes skilled or 'expert' in problem solving. Heuristics greatly simplify the decision-making process by reducing the decision space. In the example above, a heuristic might be:-

If	No Signal Output From Main Circuit
And	Component 214 Has Blown
And	Circuit 12 Is By-passed
Then	Re-solder Circuit 19 To Specification BS9160.

By using heuristics in this way, it is possible to emulate explicit aspects of expert decision-making in narrow domains. Heuristics may be of a more complex nature by defining levels of uncertainty associated with each expression. For example,

If	Car Does Not Start
Then	Fault With Starter Motor (0.76)
Or	Fault With Battery (0.60)
Or	etc...

A particular type of uncertainty is used in the above example called *certainty factors* ; the numbers indicate the likelihood of various causes of the car not starting, with a probability of 1 indicating full certainty and 0 indicating no certainty. Certainty factors are described in more detail in Appendix I. Experiences in developing an expert system described in Chapter 7., and those of other manufacturing companies described in Chapter 8 show that in most cases, heuristics may be expressed explicitly without the need to define levels of uncertainty. Furthermore Towriss (1988) argues over the mathematical integrity of defining, and the empirical difficulties of quantifying, uncertainty in this way.

From the above examples, it can be seen that the most natural way of expressing and representing heuristics is as IF-THEN rules as Boden stated in her definition above. Indeed the Vendor survey of Chapter 7 shows this to be the most popular form of representation in the UK. However for more complicated problems where the structure of knowledge is 'deep' or tacit, more complex means of representation are required such as Frames and Semantic Networks.

b) Structure

In an ES, there is a separation of general knowledge about the problem (the rules forming a knowledge base) and methods for applying the general knowledge to the problem (Williams:1986). Thus, in an expert system, the program itself is only an interpreter or general reasoning mechanism and ideally it is possible to change the system by adding or removing rules or other structured representations from the knowledge base. In a conventional computer program by contrast, knowledge pertinent to the problem and methods for utilizing the knowledge are all intermixed.

2.3.5. A Classification of Expert Systems

Equally important to defining and understanding the differences between technologies is an understanding of how they are intended to be used. An indication of this, reflecting not only the role of ES but also the principal viewpoint of the systems developer, is in how ES are classified. The predominant focus of classifications has been from a technology viewpoint; expert systems have been classified on the basis of their construction, for example van Koppen (1988) identifies categories of systems according to methods of inferencing and interface; Turban (1988) makes a distinction between software types(shells. AI toolkits etc). ES may also be classified from an organisational viewpoint. Holroyd *et al* (1985) for instance, look at the physical characteristics of organisational problems-structured/unstructured, level of judgement required and so on. Similarly, Luconi *et al.*(1986) define a hierarchy of components of problems according to data, procedures, goals and constraints, and strategies: strategies for example determine which procedures to apply to achieve certain goals. Curnow (1987) considers the structural value and impact that ES will have upon the organisation. This author identifies classes of ES which involve routine tasks where non-experts should be involved and discretionary tasks where experts are used. The theme of organisational value may be expressed in business terms also, and Chapter 7 looks in detail at how business models may be applied to define the 'strategic' value of potential ES applications.

It is also valuable to attempt to classify applications from an end-user viewpoint. Wensley(1989) for instance define four classes of end-user role ranging from *intelligent assistant* where the individual is an expert in the domain to an *advisory aid* where the individual has no detailed knowledge of the domain. Ostberg (1988) adopts a similar classification but looks more closely at the possible skills and personal impacts that each class carries with it. Ostberg concludes from his analysis that most commercial ES are promoted as being 'transactional' where the function of the system is to take overall responsibility and perform the tasks through a 'human facilitator' when in fact in most user organisations, ES adopt a 'commentary' role where the end-user performs the task independently and only uses the ES when in need of a second opinion.

Classifications are an important part of problem identification and application selection, as Chapter 5 shows. However, it is important to understand their limitations as a means of defining and representing ES. In most cases classifications are made from a single viewpoint and because they neglect other viewpoints they make generalisations about the effects of the technology. This can only be resolved, as indeed Chapter 6 has done, by combining classifications from different viewpoints within a 'multi-perspective' framework .

2.4. Placing Expert Systems Literature in Context

It should be reiterated that this thesis is structured so that appraisal and critical reviews of the extant literature on expert systems is divided by topic so that, for example, a critique on current development methodologies is given in Chapter 4, whilst problem selection techniques are discussed in Chapter 5 and so on. The purpose of this section is to place all extant literature on the development and use of expert systems within a wider context in order to show changes in viewpoint in its coverage over the last twenty years. In doing this, four broad perspectives emerge from the literature: pioneering or 'founding father' work; leading-edge developments and research work; the forecasting of social and organisational impacts; and the emerging 'reality' of expert systems in practical use.

2.4.1. Pioneering Literature on Expert Systems

From the 1970's to the mid 1980s the ES literature was awash with detailed accounts of a few pioneering expert systems. These systems had an important role in establishing ES as a new field of computing and were effective in communicating both the concepts and the potential for this 'new technology'. The following systems are perhaps the most well known and have provided the technical basis for the majority of ES currently in use:

a) **MYCIN**: a system for diagnosing blood infections (Shortcliffe: 1976). It is the most widely studied and is seen as the flagship of ES because it was one of the first to make use of production rules and employ a backward chaining inference method which most current ES utilize. This system was not used commercially.

b) **DENDRAL:** a system for inferring molecular structure from experimental data provided by mass spectroscopy and nuclear magnetic resonance (Buchanan & Feigenbaum: 1985).

c) **PROSPECTOR**: this system is based on the technology of MYCIN and was designed to help geologists locate ore deposits (Duda *et al.*1978). Its knowledge base contains expertise on the geology of ore deposits and the classification of various types of rocks and minerals. Given data about a particular are, this system is able to estimate the chance of finding various types of mineral deposits. This system was not used commercially however.

d) **XCON**: originally called R1, this system was designed to help technicians configure VAX computer systems. It received a customer purchase order and determined what, if any, substitutions and additions were needed to make the order complete (McDermott: 1981). The system produced diagrams showing the spatial and logical relationship between components to be configured. The system took 50 man-years to develop and is made up of 3500 rules. The system is able to configure over 97% of orders, is continually maintained and remains in commercial use.

e) **Dipmeter Adviser**: this system helps to determine whether an oil well will contain oil or be a dry hole. Oil explorers lower specialised logging instruments, known as dipmeters, into oil boreholes to provide information about the geology of the subsurface formations being pierced (Smith: 1984). This information is used by the dipmeter Adviser to make a prediction.

There are a number of common characteristics of these pioneer systems:

a) the applications were in very specialised areas such as medical diagnosis, mineral prospecting and research chemistry.

b) all systems were very large and took between 20-50 man years to develop (Sviokla: 1986).

c) most of the systems were designed to be used by professionals in their domain rather than by lay people (OU:1987).

d) the systems were developed in a strong academic/research environment and culture rather than in an industrial setting.

The style of commentary is also significant. Literature tends to be biographical rather than analytical and fails to describe how the systems were being implemented or used. Ostberg for instance identifies the euphoria with which the technical capabilities of ES are reported: and yet, from his analysis of 'pioneer' systems, it is significant that of the previously mentioned expert systems, none are in use but for XCON. Even here, Durham(1987) argues that 'its size and complexity have grown to a point where the system can no longer be maintained effectively'.

As first generation expert systems, with a clear function being to communicate the technology's concepts, methods and capabilities, it was inevitable that a strong 'technology push' would ensue. However, this effort was restricted to certain areas only: Rifkin (1985) argues that the actual diffusion of these pioneer systems in manufacturing was very low with the focus of effort concentrated in scientific and engineering applications in 'controlled' environments. As Ostberg comments, 'There is obviously a serious gap between claims and reality of ES applications. Thousands of articles have been published about development work in progress and about the potential use of ES. Only very few systems are in actual use outside the laboratory environment , and evaluation reports are non-existent.'

A consequence of this, was that a hype was generated by the media and popular computing press which have had, as Ovum (1988b.) note in retrospect, lasting damaging effects. Stevenson noted three in particular (1989). First, it portrayed expert systems as being completely different from anything else; the following quote epitomises this, 'as programs that make computers appear to think, reason and use human-like judgement, expert systems represent a revolutionary branch of technology' (Kehoe: 1986). Secondly, it presented a powerful expert replacement focus, " An expert system does not have many of the short-comings found in human experts. Unlike humans they do not forget, they cannot be headhunted and they are always available...they enable computers to move into an area not normally addressed by computer systems, the replacement of human experts' (Greenwood: 1985). Thirdly, the media and vendor literature tends to exaggerate the level of use and benefits of the technology, "This book is about expert systems and how they will change the world of business...expert systems will also help America to solve its productivity problems" (Harmon & King: 1985).

2.3.2. Technology Capabilities, Design and Development

Although developments in the manufacturing sector lagged behind other areas, the direction of research and coverage in the literature in this sector retains many of the characteristics of the 'pioneer systems' with a strong focus upon technical excellence and associations with Artificial Intelligence. This is apparent from current research and development themes exemplified by recent collaborative research projects in Industry (ACME: 1990). As examples, these include :-

a) intelligent ESs to provide design and manufacturing data for forging
b) the intelligent selection of manufacturing control systems
c) applying intelligent systems to made-to-order manufacturing
d) intelligent planning systems in advanced manufacturing technologies
e) an ES approach for the control and sequencing of robotic assembly
f) knowledge based CAD and design for economic manufacture
g) an ES for the automatic generation of process plans for rotational parts

Two observations may be drawn from this. The first is clarified by a recent Government report (MI:1989) which shows that the bulk of research and development work is in three areas of manufacturing: design (with a particular focus upon design for manufacture); planning (particularly process planning and shop scheduling); and process control. A second observation is that in the research projects above, there is a strong focus upon 'intelligent' design and planning systems suggesting a turnaround back to the concepts of Artificial Intelligence (AI) explored prior to the development of 'pioneering' expert systems. This is endorsed by Partridge (1988) who considers the future importance of A.I. in manufacturing but combined with, rather than distinct from, software engineering in manufacturing.

Although this objective has been made much more possible by advanced techniques such as object orientation, neural connectivity and parallel processing and other improved methods of representation , inferencing and processing, research and ES literature appears to have by-passed evaluations of actual use, processes of implementation and developments need. The effects of this bias towards technical potential rather than understanding the requirements of the user community is expressed by d'Agapeyeff(1984), 'Expert systems in business are not as they are said to be. They bear little similarity to impressive projects found in the literature on AI. They are not inherently complex, demanding, risky and expensive. They are not built by implementors with doctorates and world-ranking expert knowledge. Instead they are as they have to be: simple in form, limited in aims and built by rather ordinary people with local and relative expertise.'

The contrasts between the scientific focus of research against the actual use of the technology may account for the findings in Chapter 8 which show confusion over what ES is, how it may be used in manufacturing and how such systems are actually developed. Moreover, it could also explain recent criticisms that research into methods of development is oblivious to the actual constraints and circumstances of development (O'Neill & Morris : 1989).

2.4.3. Forecasting of Social and Organisational Impacts

This body of literature emerged as a natural reaction to the strongly deterministic, technology driven models of expert systems use presented by developers of pioneer systems and more recently by advocates of 'intelligent systems'. Commentators stress the potentially adverse social and human impacts of expert systems. For example, using the case of early expert systems, Gullers (1988) argues that there is a strong human replacement focus in their use. Smith (1987) describes a 'human EPROM' or throwaway workforce in which expertise is endowed upon the user for a specific function and then erased and 'reprogrammed' as demand for expertise requires. Indeed, in a previous publication by this author, an analogy is drawn between the mechanisation of physical tasks, which characterised early industrial modes of production (specifically 'Taylorism' and the 'Principles of Scientific Management'), with the potential for expert systems to automate human cognitive tasks (Holden:1989). From such claims, persuasive social theoretic arguments emerged on the nature of knowledge and the essential differences between man and machine. For instance, Goranzon (1988), Gill (1986), Rosenbrock (1988) and Gullers (1988) all argue in various ways that current approaches to ES development assume that human expertise is predictable and can be explicitly and logically described in a formalised language, when in fact, this overlooks the tacit dimension of human expertise which includes intuition, creativity and a practical knowledge of how to do things all of which cannot be replicated by expert systems. Brodner (1986) and Johanessen (1988) argue further that rather than focus upon replacing human knowledge and skills using ES or indeed other technologies, their role should be one of 'human facilitator' in which the function of ES is to enhance personal roles.

Often without much evidence, such writers are little more than "social soothsayers". However, the value of this literature was that it opened up the field of ES to consider the human and organisational effects and requirements of development and use. For example, the studies above conclude that there can be no aggregated or normative model of human behaviour, reflecting that the design of an expert system's user interface must therefore be highly customised to personal needs. It also brought in other social disciplines which began to consider the processes of change as well as simply the impacts, such as applying Mumford's participative design and Checkland's 'SSM', as Chapter 4 discusses. Finally, it has provided a vehicle by which to nurture and develop new concepts- for instance the author describes the opportunities for human-centred or 'anthropocentric' design in ES development (described in detail in by the author elsewhere, Holden:1990b, and summarised in Chapter 9). The value of this literature therefore must be judged not solely upon its effectiveness in applying individual social and organisational methods or techniques, but also by the way in which the extant literature on ES has been enlarged, even in what were previously technical circles, to incorporate organisational and human viewpoints on the value and use of ES. This process has been assisted by a fourth category of literature discussed below which shows the 'reality' of ES development and reveals that these viewpoints are of paramount importance to development success.

2.4.4. Actual Developments and Commercial Use

A fourth body of literature has emerged more recently, although it remains scarce, which looks at precisely how expert systems are being developed and used from a user organisation perspective rather than a market or research point of view. This is constrained as d'Agapeyeff and Hawkins (1987) point out by the reluctance by these users to divulge their approach and by a lack of operational systems which can be analysed and evaluated. Where such experiences are documented though, they provide a valuable source of information for the following reasons:-

i) They identify a 'realism' from the 'hype'

The extent of the 'sensationalism' is made clear by Stevenson (1989) in a study of financial expert systems. The author found evidence of: unverified sales claims by suppliers; unsupported, elaborate claims of benefits and applications; unsubstantiated optimistic claims in the literature; and undisputed, exaggerated consultants' claims. This body of literature however, although not immune, has shown that it is able to rationalise the hype of the market and suppliers of ES tools, hardware and development capabilities, as the reality of use transpires. This is demonstrated by a concluding remark in a study by Andrews (1988) on the practical value of ES in business, 'It would..be totally wrong to suppose that expert systems are a panacea for every business problem. Applications must be chosen with discrimination and executed with care.' (p83.).

Empirical studies by D'Agapeyeff and Hawkins (1987,1988) amongst others, have also been successful in downplaying the connotations associated 'artificial intelligence' and the suggestions that expert systems have to be large and complex to be of value by showing from survey results that most systems in commercial use are in fact small scale and simple in construction. They also indicate a number of practical lessons learnt from project successes and failures in developing ES. Four factors in particular seem to be cited quite frequently although in most cases no suggestions on how such lessons may be applied is given. These are:-

a) the need to integrate ES within 'mainstream' Information Technology (IT) and company computing department activities (Worden: 1988)

b) the need to incorporate strategic planning (Sviokla:1986) and 'business analysis' (Harbridge:1989) into ES development.

c) the need to manage the 'change process' and address issues of implementation and operations (Leonard-Barton: 1984).

d) the need to establish individual project roles and characteristics; such as whether there is a project champion (Pederson:1989), whether there is expert support (Turban:1988), and end-user participation (Mumford: 1989) for instance.

ii) They identify limitations and development problems

There are a number of organisational difficulties in developing expert systems in addition to purely technical problems. For instance Bramer (1988) describes the lack of management commitment, business secrecy, poor organisation and a fear of the nature and costs of the technology as prime barriers which prevent implementation. Similarly, Ackroff *et al* (1990) describe improper budgeting, difficulties in retaining management support, a lack of attention to training and a failure to integrate ES project development within the prevailing management, control and administrative systems of the organisation as the main reasons why , in the authors experience, projects failed to make the transition from prototype to full-scale operations. At the point of entry into an organisation, Turban notes the difficulties of communicating the benefits of expert systems to a company, gaining management support and of estimating time and resource requirements. This clearly depends on the approach to technology adopted and the choice of problems addressed. Sell(1987) defines an 'uncertainty' in problem and application selection experienced by companies and the lack of an approach towards identification.

The political problems of introducing a new technology should not be underestimated either. For example, Sviokla describes the conflicts between those in an organisation charged with development and those that actually make use of the technology. Milne (1990) suggests that such problems arise often because the expectations of management as to what a project will achieve differ from the understanding of the problem by the development team.

iii) Reveal actual as opposed to 'potential' benefits

Early literature exalting the benefits of expert systems did so by making comparisons against the weaknesses of human experts - they don't have lapses, they don't suffer from bad days and so on (Harmon and King: 1985). As well as an implicit bias towards replacing the expert, this literature also defined 'potential benefits' such as securing, duplicating and distributing human expertise (SEAI:1988), providing a pool or knowledge-bank of experience (Waterman: 1986) and improving upon human performance and reliability (Hayes-Roth *et al* 1983) without any such evidence through commercial use.

As actual implementations were reported from an end-user perspective, the performance objectives and benefits achieved were distinctly less ambitious than potential benefits. Foremost, is that the purpose of the systems were seldom to replace the expert but rather to relieve him or her of the more routine problems which could then be undertaken by less skilled people. However, the expert was still required to solve the more complex problems. Two recent examples of applications reported in the literature reflect this change: the first is in order processing and the second in fault diagnosis - both are in current use.

i) **OCEX**: Hermann (1990) gives an account of an expert system for the order clearing of medical products to provide instructions on how to proceed with each order. This author defined benefits in terms of quality improvements, time saving in order processing and saving the expert time to devote more attention to more important 1% of problems which were passed on. It is significant too that the expert was made responsible for the expert system and for maintenance in particular.

ii) **RBEST**: Braunwalder et al (1990) describe an expert system which determines the cause of failure in disk drives during manufacturing stress testing. The system was used by the specialist in order to reduce the time taken diagnose and fix each failing disk drive and thereby increase the production rate twenty times over. The system also led to fewer 'no faults found' units being resubmitted for test with no satisfactory explanation for failure.

These examples first show that expert systems can provide significant returns for a company. There has been a tendency perhaps in this critique to overlook this fact. Secondly though, the nature of these benefits, the way in which they are expressed, and their effects are not standard. Rather they differ according to the problem domain, type of application and the organisation involved amongst other factors. Furthermore, benefits will have an impact at different levels in a company . Chapter 6 shows for instance that this can be at a business level, by directly improving competitive advantage for example, or as in the examples above, at an organisational level by improving organisational effectiveness. The impact of benefits may also be highly localised, improving the efficiency of a particular operation for example.

As well as levels of impact, it is also important to expand upon the definition of 'benefit' to consider the wider positive effects which materialise, such as improved awareness and dissemination of ES ideas. For example, the benefits of the Aries project (one of the 'club' developments of the UK Alvey programme) were expressed by Butler & Chamberlin (1988) in terms of dispelling fears, uncertainty and doubt and communicating the concepts of the technology. The actual commercial return on investment was of secondary importance. Thus 'benefits' should be measured in relation to their organisational objectives. In practice, as

Chapter 8 will show, this means broadening the evaluation process to consider not just the return on investment from implementing the technology, but also the total 'added-value' which accrues from the technology transfer process.

In terms of developing an approach in the client organisation, this fourth category of literature was by far the most important. However, as with all valuable things, it is also the most occasional and rare. The predominant focus of literature in 1990/1991 is still pitched strongly towards the technology, its technical merits and potential capabilities with very little coverage from a user's perspective. This has a self-validating effect since the main source of information for user organisations is from the technical literature through the market and media. Therefore as D'Agapeyeff and Hawkins observe, the 'myths', uncertainty and misconceptions about the technology are perpetuated. This sustains a strong 'technology-push' culture in industry which as Crispin *et al* (1989) note is exacerbated by a lack of skills, methods and qualified staff to design 'good' expert systems which are of organisational value.

It seems then that ensuring an organisation's requirements are met, whilst simultaneously evaluating the potential contribution of expert systems, in a climate which strongly favours the technology and technical issues is a forlorn hope. One possible way out of this dilemma though, is to shift focus away from the technology foremost to consider instead the organisation's problems and needs. At a conceptual level, this requires a new framework in which to undertake technology assessment and this is the principal concern of the next chapter.

2.5. Conclusions to Chapter Two

This chapter has provided three things. Firstly, a mapping of the thesis defining research direction and linkages between chapters. Second, it has provided a series of definitions of expert systems with weight added to how they are actually being used rather than what the market aspire to. Thirdly it has provided an overview of the coverage of expert systems in the research and development literature.

A review and cataloguing of expert systems literature reveals firstly the confusion over what expert systems are and how they should be used. This reflects the varied viewpoints of expert systems commentators; four basic viewpoints in development are identifiable. A first phase of 'pioneering' literature which focused upon a few, very large ' show case' developments to impress upon industry the potential for this technology. These systems were mainly experimental however and failed to distinguish between what was conceivable in a controlled research environment and what was achievable in practice. A second body of literature which retains many of the characteristics of the first, is state-of-the-art research and looks at how Artificial Intelligence (A.I.) and advanced expert systems techniques may be applied to specific industrial problems. A reaction to both the above phases of literature is a third which forecasts the potentially adverse social and human impacts of expert systems, which although 'alarmist' in retrospect, did have a value in enlightening the expert systems research community of the centrality of human and organisational factors in development.

Perhaps of most use to this study however is a fourth, emerging body of literature, although still infrequent, which describes **how** expert systems are being used, not from a supplier's or technology research viewpoint, but from the perspective of user organisations themselves. Such studies describe the processes of innovation and development and the problems and barriers to implementation. They also indicate the organisational role of expert systems and, according to needs and preferences, how they may be developed and utilised most effectively. The scarcity of such information, together with the uncertainty of the technology's role and equivocation over methods of assessment, development and evaluation, are the main motivations for this study.

Classifying the literature in this way is useful because it shows discrete phases in the maturation of expert systems, from the initial euphoria and conception of ES as being ‘revolutionary’, often with the misguided connotations associated with artificial intelligence, to the actuality of its use in industry. It also highlights how different viewpoints of the same situation can lead to drastically different assumptions being made. This is an useful lesson and precursor to the next chapter, the main theme of which is that in order to fully comprehend a situation or problem, particularly during initial stages of technology assessment, it is necessary to observe it from all viewpoints or ‘multiple perspectives’. This is critical to achieving a true understanding of problem and situational needs and thus subscribing to an appropriate solution.

Chapter 3

Defining a Basis for Technology Assessment and Development

3.1. Introduction

This chapter is concerned with how manufacturing companies should go about assessing the organisational value and need for expert systems technology. It is less concerned with tools and techniques, which is the theme of the next chapter, but rather in understanding the underlying viewpoints of those that perform such roles in these companies. Clearly though, how a problem is perceived will define the solution approach.

A review of the extant literature in the last chapter identified a number of short-comings in the development and use of expert systems (ES). It was apparent from this analysis that many of the problems experienced arose through a failure to understand the problem context, processes and organisational settings in which to apply the technology. Furthermore, this was demonstrated at all levels of development, from initial problem conceptualisation through to evaluation, and not solely at the level of technology design. These difficulties are compounded further by the fact that those charged with expert systems development have a strong technical background; for instance Chapter 8 shows that project responsibility is often under a computer specialist or analyst. This confuses the technical skills required to develop a system with the management and organisational skills necessary to make the system work in the company. This sentiment is embodied in the statement of Bobrow and Stefik,

'...Remember that you are not designing a computer system, but are putting a process in place in a user organisation' (1985).

Thus, although a technology focus is clearly *necessary* in the development of ES, it is not *sufficient* in understanding and managing the total change process. The significance of this point may be eluded by technical personnel because of a inherent preference to view problems and solutions in technical terms alone. This probably simplifies the organisational complexities and intricacies of the technology assessment process in companies, but it does highlight an underlying concern that expert systems are managed and implemented with a strong bias towards ES technology and technical issues. This concern however is not specific to expert systems, but part of a wider 'scientific culture' which predominates in technology developments and predisposes the 'technical specialist' to determine and control a particular philosophy of design and implementation.

A mapping of the chapter is shown in Figure 3.1. It begins by underlining the importance, and the difficulties, of re-addressing attitudes and viewpoints of technology assessment at a conceptual level, particularly in dispelling the focus upon a technical perspective . It shows that there is a wide body of literature available which define specific concepts adopting either principal organisational or individual perspectives which may be applied to ES assessment. Each have their strengths and weaknesses. For example, organisational concepts are valuable in highlighting the importance of understanding problems from more than a single technical viewpoint, stress the necessity to involve end-users in development and consider *how* development processes should be undertaken. However they over-emphasize the centrality of the organisation and adopt a simplistic, aggregated notion of the 'individual'. Therefore although current conceptual approaches contribute towards understanding and defining organisational and individual needs and also in identifying the deficiencies of a technical perspective alone, they suffer the same limitation in focusing from a single dimension.

This analysis leads to the conclusion that all three perspectives should be combined in some way. The literature again provided some pointers towards way of achieving this which raise questions about whether a conceptual model or framework should be used; at what level it should be applied; and with respect to what . The idea of combining perspectives in assessment also raised a number of inherent conflicts, most notably the dialectic between formal, hard(external) and prescriptive processes and soft, informal(intrinsic) and pro-active processes. In order to reconcile these differences it is first necessary to create a common understanding by placing a context upon the use of perspectives. Two studies were helpful here: the first was by Markus and Robey (1988) who define an 'emergent perspective'. This states that change emerges from an 'unpredictable interaction' between technology and its human and organisational users. The second concept is Linstone's Multiple-Perspective Concept (MPC) which provides a enquiry framework for understanding the balance of perspectives according to settings, viewpoints and processes (1981). The chapter concludes with a detailed account of MPC and projections on its use in the assessment and development of expert systems.

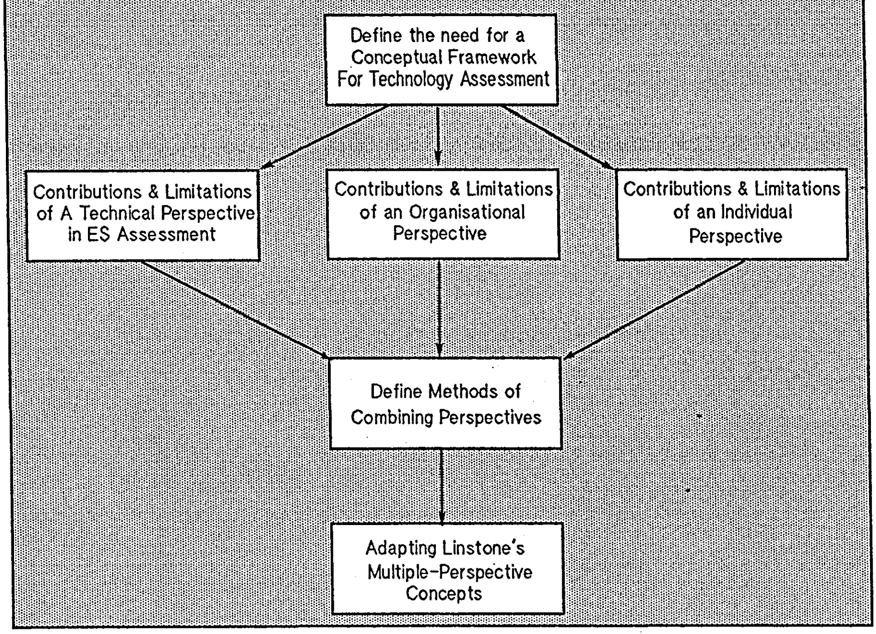


Figure 3.1: A Mapping of Chapter 3.

3.2 The Predominance and Limitations of 'Technology-centred' Assessments

The predominance of a technical perspective in information technology development has been described in general terms as being implicit in western industrial culture and attitudes towards the organisation of work. This view is exemplified by Hoos (1979) who speaks of a 'scientific culture' and Winograd and Flores (1983) who add, 'Current thinking about computers and their impact upon society has been shaped by a rationalistic tradition that needs to be re-examined and challenged as a source of understanding'. More focused analyses consider the influence of this paradigm from a particular organisational context. For instance, Leonard-Barton *et al.* (1988) define it as an expression of technological determinism, in which external market relationships between vendor (producer or supplier of software) and user organisations shapes the course and direction of development.

At the organisational level, Linstone (1981) and Checkland (1981) define the inadequacies of a Technical(T) perspective in defining socio-technical problems. The bulk of criticism of the technical perspective however lies in the choice of development tools used (Whitaker and Ostberg:1988). Rauner *et al.* (1990) for instance defines a 'technical rationality' in the shaping of production systems in which the functioning of technology is based upon an inner logic which defines a one-best way to develop technology using optimisation techniques. Mumford(1988) refers to an overemphasize upon formal computational methods and scientific techniques; and Markus(1984) refers to the engineering emphasis upon data and models through analysis and compartmentalisation. In the latter two cases, criticism is directed at the scientific process of full specification prior to design itself; Finally, Hirschheim(1987) notes that the evaluation process itself may favour and therefore promote a 'scientific rationale'. She adds, " Precisely how evaluation is undertaken is largely dependent upon the available evaluation tools and techniques. Those currently available have a decidedly quantitative, rational and technical orientation to them which leads to a particular type or form of evaluation."

At the individual level, Cooley (1989) and Gill (1986) define the human impact of 'technocentric' and 'machine-centred' systems which are defined as technologies which marginalise and downgrade human skills and potential. These authors focus upon the use of technical rather than human measures during evaluation and the use of models and techniques which preclude the inclusion of subjectivity. Other authors focus upon the capacity of individuals to shape or perpetuate a scientific approach in the organisation. Clegg (1989) for instance defines a technical focus in the selection of technical staff to manage and design systems and thereby define the outcome of the project.

All these theories mention similar facets which constitute a 'technical perspective' but define impacts and effects which are peculiar to a particular context or focus of analysis. This suggests that alternatives to a scientific approach should therefore be sensitive not only to other perspectives, but also to the level and setting in the organisation at which they are being considered. Furthermore, although these theories relate to information technology in general terms, it is necessary to consider the case for expert systems as being unique on the basis of its purpose, design and operational use. The following discussions therefore considers evidence which relates to the ways in which a technical model is present in ES development. It focuses the discussions in particular at three levels: the way in which the external vendor-user organisation relationship may be described as being deterministic; the predominance of a scientific design philosophy in ES development; and the potential 'machine-centred' use of ES and theoretical debates on the implications upon skills and work organisation.

3.2.1. Vendor-User Organisation Relationships

The notion of external influences upon an organisation is often seen in terms of a unidirectional process of sales rather than as an interactive process of collaboration. The technical perspective predominates not only in the way that ES technology is presented to user organisations, but also as a response to this, in the way that the transaction is managed. Moralee (1987) points to the need to develop the right development 'culture' based on not whether the latest ES technology is being used but that it is appropriate to the needs of the organisation. Brodner (1986) too, describes the predominance of a scientific culture at the market level, but as a response to changes in the structure of the manufacturing sector rather than the software sector. This suggests a demand-pull rather than technology push scenario and arises through changes in the structure of competition. Under these new conditions, the ability to adopt products to customer requirements and yet guarantee short delivery times is considered more important than price and quality. In response to this, manufacturing companies, according to Brødner, have focused on the extensive use of technology to computerise design and manufacturing operations with the role of expert systems being to widen the range of tasks which can be computerised and be integrated into a central computer system. The implicit assumption of this 'technology-centred' approach is that it is technically feasible to computerise empirical knowledge of production, but also that by computerising these tasks, they will make operations more effective. The demands placed upon the vendors therefore are for large, highly complex embedded expert systems customised to user organisation's needs. Indeed this sentiment in the mid- 1980's was reflected by a prominent academic (Feigenbaum:1986) who advocated a "go-for-broke" policy in that expert systems had to be big and ambitious to be worthwhile.

In fact, as Chapter 8 shows, the trend is quite the opposite. A large share of the market for ESs is for low-cost standard products, which are stand-alone (non-integrated) and have limited functionality. Moreover, an increasing share of the market is being taken up by application-specific software. It is both these areas where there is a strong technology push by vendors. Ovum for instance speak of the "hype" generated through the over-estimation and over-specification of the software by vendors (1988a, 1988b.). This situation was made worse by a number of characteristics of the ES market identified by these authors: many small vendors reliant upon a single product; expert systems marketed as 'leading-edge' and Arificial Intelligence (A.I.) based; and little evidence of operational systems such that users were dependent upon vendors for technical information. Although Ovum refer to a 'second wave' or second generation of expert systems in the 1990s which are more functional and offer better support, the basic structure of relationships between market and vendor remains the same.

As the last chapter observed, the hype of the market is perpetuated rather than restrained by the bulk of ES literature. Stevenson (1989) asserts for example, that commentary on the ES market is strongly technology and product based rather than from the viewpoint of the customer's needs, and that this reflects the general direction of knowledge and expertise from the vendor to the user. The technical bias apparent in ES literature and discussions on ES is widespread. Their technological sophistication has in itself become a marketing item. Hirschheim(1985) found with Office Automation technology that this was partly due to the enamouring effect of the technology, partly due to the type of person who becomes involved in their development and implementation (the 'technocrat') and partly to the reasons for considering its use (to make the organisation more effective through technological means). The ability of the user organisation to make a free choice of product is thus constrained by its technical understanding of the product and the internal capability to evaluate and monitor its performance and the performance of the vendor. Markus(1984) views this relationship not as a deterministic one, but as a political process in which each party has the opportunity to achieve its own goals and objectives at the expense of the other. Markus therefore sees it as the responsibility of the user organisation to manage the assessment process correctly rather than attribute blame on the structure of the market. Similarly, Mumford (1987) shifts the weight of responsibility upon the user organisation for a failure to understand the social context of ES development for example and thereby take account of the various inputs and outputs during the design *process*. Thus the implication from these authors is that, assuming that the market transfer process remains technology driven, the initiative for a balanced process of technology assessment must come from the user organisation, which means in practice providing an appropriate technology assessment infrastructure.

3.2.2. Technical Design Philosophy and Systems Analysts

A number of commentators have noted that conventional software engineering (SE) approaches are based upon a 'rationale' for development which exhibit certain universal scientific characteristics. For instance Markus refers to the complete, one-time specification of design features before execution based on the assumption of predictability and repeatability. The development lifecycle itself is intended to rationalise and routinize the process of building systems by imposing a discipline upon the developer and on the development process. Friedrich(1990) refers to a trend towards the standardization of software production in which development has moved from being intuitive to becoming a logical process. Furthermore, this process of objectivisation and depersonalisation is perceived by management, Friedrich claims, as an improvement in programming efficiency. The notion of structured programming also fits better into the formal organisation structure of the Computer Department or IT function since standardisation of programming reduces the dependency of the enterprise from the IT specialist.

The more recent development of CASE (Computer-Aided Software Engineering) tools further rationalises the human role in software development, so that through the combined use compilers, debuggers, and program libraries, a "software development machine" is generated (Ackermann:1989).The standardisation of the software development process has also been followed by the standardisation of the software product itself. The development of standard software like text processing, and off-the-shelf expert systems, is based on the logic that there is a 'one-best way' of the work-place. In a sense, this has been extended into the expert systems domain with the trend towards 'off-the-shelf' systems. These, as Boden (1986) argues, presuppose that knowledge is a 'commodity' which may be transplanted from setting to setting without understanding of the context to which it is applied. In these arguments , a central theme is that the choice of technology or development technique is influenced by an original conception of the problem. Thus, the design orientation of the participants is important in understanding why a technical paradigm exists. Linstone (1981) defines a 'rational actor model' of the individual that discounts subjectivity and political orientation and assumes a rational and objective role in the formulation of problems and choice of formal methods by which to solve the problem. This author also notes that the rational actor model is the strongest in software engineers and analysts who in turn tend to control and make up the bulk of members in project assessment teams. He adds, 'The technologist's preference for the T perspective commonly mirrors a natural bent; he feels more comfortable dealing with objects than people' (1988: pg69).

There are certain attributes and characteristics of the present ES development process which may be defined as being scientific in approach. Kelly(1986) for instance has described ES development as a process of 'scientific experimentation' and Ostberg(1988) describes the use of scientific measurement techniques to codify human expertise. Raveden *et al* (1988) point out that a basic concentration upon the scientific component of development leads to the use of technical criteria in assessment and the corresponding allocation of 'formal and hard' techniques. Similarly, Rauner *et al.* define hard techniques in terms of sequential processes in which the technical aspects of a work system are designed first and other non-scientific elements are made to fit or adapt to the technical system. Furthermore, the system is appraised using formal techniques which measure 'quantifiable and tangible' benefits and define failure in technical terms. The basic implication from all of these accounts is that as well as being implicit in the design and marketing of ES, a technical rationale is also prominent in the methods and tools of development.

3.2.3. The Consequences of a Technology Perspective at the Individual Level.

The most graphic manifestation of a technology perspective is at an individual level. Perrolle(1986) for instance speaks of 'intellectual assembly lines' in which "mental labor is subjected to both the rationalisation of its knowledge and the gradual automation of its productive activity". This raises two issues; the technical feasibility of automating human cognitive processes, and the social desirability of doing so. Gill

(1986) defines the effects of a T perspective in sustaining and supporting a *machine-centred* model of individual behaviour in the design and use of expert systems. Underlying a machine-centred approach is the philosophy that it is possible to use expert systems to automate human expertise because of the formal and objectivised nature of knowledge. A machine model of the individual assumes that knowledge and cognitive skills can be studied scientifically as if they were the mechanical movements of a machine and therefore components of expertise can be logically described in a formalised language. Rauner *et al.* call this a computer based 'Neo-Taylorism' because the emphasis upon the mechanisation of knowledge is analogous to the way in which Taylor (1911) mechanised shop-floor activities at the turn of the century (1990). A number of other writers also note similar personal and social consequences such as de-skilling (Cooley:1989), the fragmentation and objectivisation of tasks (Guller:1988) and the separation of design knowledge and centralised control from manufacturing activities (Brodner:1988).

Machine-centred approaches to ES development adopt an explicit model of individual behaviour in which humans are perceived as information processors that calculate according to rules and data. Gill comments that this strategy of rationalisation prescribes the 'correct' behaviour as precisely as possible and thereby restricts human expression and initiative which are considered unquantifiable and therefore unimportant (1986). The user subordinates the sensory and actuating portions of their work to the machine. Information is aggregated before the humans involved make any decisions. It is therefore a variety reducing exercise where the user is trained on how to use the expert system rather than learn the general principles and concepts behind the operation. Cooley (1988) regards this as a process in which the technology retains control of the individual to the effect that "humans are rendered more passive as technology becomes more active." Furthermore Brodner commented that when problems arise in this man-machine relationship, they are addressed by an extended use of computers in order to reduce labour and gain better control of production. Where the tasks are too complex for the machine, they are decomposed into simple tasks or 'bundles of expertise' in order to facilitate their automation.

Despite the uncertain ground of projecting the human consequences of a T-perspective in ES assessment and development, it is important to consider these issues in order to understand the possible diversity in motivations and assumptions made by those who use or deal with expert systems whether vendors or end-users; the span of theoretical debates is a reflection of the span of individuals involved. There is no substantial evidence that expert systems *per se* have an automatically single machine-centred or mechanistic (or indeed enhancing) effect upon the individual. However being aware of the possible effects of expert systems through discussions of machine-centred systems as the'worst-case' scenario, is a necessary first step in planning for more desirable human consequences. It must be appreciated though, that a principal failing of machine-centred theories is that they rationalise impacts from a predominantly individual perspective. Their inherent weaknesses stem from the coverage of socio-cultural and technical issues from exclusively within this perspective without due consideration to other viewpoints.

3.2.4. The Contribution of a Technology Perspective in ES Assessment and Development

The effects of a T perspective have been described in this chapter at a market level in defining the relationships between vendors of ES products and user organisations; at the organisational level through discussions on the way technology is assessed, in the allocation of analysis and design tools and in development itself. It has also been expressed in individual terms through coverage on machine-centred systems design. Common to all these processes and settings are certain characteristics which make up a technical paradigm. Linstone defines seven attributes of this paradigm which may be used to define its strengths and limitations as an approach towards developing expert systems:-

i) The problem-solution view: This view presupposes that a solution always exists. It defines the problem as being bounded and structured and therefore defines the solution in terms of structured techniques. Conventional programming techniques are congruent with this perspective because the design problem is more structured, although the development of a design in an organisational context makes the outcome less predictable. Linstone (1978) talks of current analysis techniques being applicable to these design systems. However, the distinction between ES and conventional programming techniques is that the former is primarily concerned with human skills and expertise, whilst the latter is concerned with procedures. For this reason, the development lifecycle of ES is much less well defined.

ii) 'Best Solutions'

Human expertise is essentially qualitative rather than quantitative. The human expert does not optimise but seeks a satisfactory solution and may infact be unable to solve problems in which case, the expert will apply methods to reduce uncertainty. In a social or political context moreover, it may be impossible or ill-advised to provide a best solution. The expert's 'model of enquiry' therefore is not exclusively scientific nor reliant upon formal methods but heuristic and holistic.

The notion of a 'one-best way' suggests a single process of enquiry. However the process of arriving at a solution may differ among experts, as experiences in building a troubleshooting expert system in *Chapter 7* show. Furthermore, the format of the solution will vary according to the different users and contexts in which the knowledge is required. Expert systems with multiple dialogues and separate levels of analysis have been produced for this reason.

iii) Reductionism

A characteristic of the machine-centred approach is the division of knowledge (Goranzon:1988). However not all problems can be simplified or made more manageable by dividing them into sub-systems. Cooley notes that human expertise is mainly judgemental and intuitive and in order to exercise these skills a holistic approach to problem solving is necessary. A scientific approach reduces the 'problem' to the technical components and thereby overlooks the contribution of non-technical factors in providing a solution.

iv) Reliance on Data and Models

Linstone identifies the usefulness systems design and analysis techniques in providing insights and guidance for decision-making, but distinguish them from the actual process of decision making which involves organisations and individuals with different perspectives from those of the 'rational analyst' or technology assessor. A pre-occupation therefore upon the refinement and increased complexity of these 'thinking aids' may overlook the core scientific values upon which they are based. The implications fo ES development are that firstly the technology represents a simulation model of some facet of human knowledge but in no way can it duplicate human behaviour or expertise. To paraphrase Linstone, " The reality created by the computer model in the mind of the programmer or user can never be a duplication of a human or societal reality" (p13).

v) Quantification

In this, there is the implicit view that by quantifying a process, it is legitimised and worthwhile and that which cannot be quantified is of no importance. The effect of this is to disregard intangible costs and potential benefits which are generated through the system. In the case of expert systems, quantification is difficult because many of the benefits expressed by users are 'added-value' , in other words they add qualitatively to a process rather than reduce costs, improve productivity and other quantifiable effects. This makes the use of traditional cost-benefit type analyses difficult to apply (Harbridge:1989).

vi) Objectivity

A technical model presupposes that users are 'observer invariant' (Linstone:1981); in other words that they are objective, rational and unbiased in decision-making. Furthermore, Clegg (1988) describes how, within this paradigm, the human input is judged as being error-prone and unpredictable. However by definition, expert systems lack objectivity because their role is to capture the a personalised and highly subjective process of solving problems. Therefore, the scientific notion that the properties of the observer must not enter into the description of that person's observations (as the expert) is clearly unattainable.

vii) Ignoring or Avoiding the Individual Viewpoint

An essential process of development is in understanding the needs of the individual and reflecting these needs through the design of the system. Moreover, it is necessary to understand as a political process the influence of various individuals (project manager, developer, expert and user for instance) over the design process, and conversely, the impact that the technology has upon the individual. In each case, the influences and effects of the individual are detached from the 'technical' processes of assessment and design. A technical model marginalises human opinion from development because it is considered irrational and secondary to more formal and analytical processes.

The implication for ES design from points v), vi) and vii), is that automating expertise as an objective will improve the efficiency and certainty of the operation. There are however many practical reasons why expert systems should not be conceived with a 'replacement focus'. Dreyfus (1972) and Weizenbaum(1976) express this in ethical and societal terms. Rosenbrock(1988) and Goranzon (1988) by contrast define the limitations of technology and the 'distinct and unique' qualities of human beings.

The predominance of a technical perspective does not reflect popularity, but rather a lack of awareness of other enquiry systems. Alone, it is insufficient in representing the implicit, soft and informal elements of development. This is not to admonish the role of scientific knowledge, tools, models and techniques, but to place their use in a context. Linstone calls these tools '*intraparadigmatic*' in that they operate within framework of a single technical perspective to the exclusion of individual and organisational perspectives. As Linstone notes, this requires that technology assessment, but also the initial problem definition and the subsequent processes of development be, "..taken out of the singular perspective and become all encompassing of all perspectives" (1981:pg16).

3.3. The Contribution of an Organisational Perspective

If a T perspective alone is unsuitable in ES assessment and development, it needs to be seen what contributions are offered from other perspectives. An Organisational(O) perspective is not diametrically opposed to a technical perspective and sees value in its selected use as a sub-set of a wider systems-based analysis. As with the technical perspective, organisational perspectives hold an initial conception of the problem from which a design philosophy is defined. It differs however in that the relationship between the two is not fixed but iterative and evolves around a context-specific definition of the problem. The essential contribution of an O-perspective is thus twofold: in understanding the problem better; and in emphasising the processes of development as well as initial requirements.

3.3.1 Understanding the Problem Context and its Settings

An O-perspective distinguishes between hard to soft problems. 'Hard' problems tend to be structured and well defined and inherit the main strengths and weaknesses of a technical model. By contrast, soft problems are unstructured and will be conceived of differently according to the organisational actors involved. Most problems in an organisation contain both hard and soft elements. To adopt too hard an approach will result in a 'technique-orientated' design approach where the danger arises that the problem situation will be distorted to fit the technique. A tendency towards Soft Systems Design will promote a problem-orientated approach where the organisational settings shape the way the analysis is carried out.

Checkland (1981), Wilson(1984), Wood-Harper *et al.*(1985) and others working in the applied systems field, approach problem conceptualisation in semantic terms. Attention is thus paid to forming what is called a "root definition" of the problem. This is a statement embodying the essential features of the situation and on this is based not only a conception of the problem but also a description of the organisational actors and their culture. This is a "conceptual map" of the organisation viewed as a system showing the interactions between system components. This map shows the relationships between the perceived activities which are considered necessary for the system to function as it does. The map differs from a conventional systems model in that it does not attempt to show what is happening in the "real world" but what has to occur to make the system described by the root definition work. These conceptual maps, again unlike classical models, may be formed at a number of different levels of differing degrees of resolution and may reflect the interactions between these levels.

The difference between organisational and technical models is that in the former, emphasis is placed on the role and intentions of the organisational actors and of their perceptions of the problem. It is assumed that no single true or optimal solution is achievable; rather that the aim is towards creating or obtaining an adequate understanding of the situation, through a consensus between the conceptual frameworks held by the actors. The theoretical underpinning of this approach lies in the recognition that all the actors in the situation, including the analyst, bring to it their own conceptual frameworks. This dimension is added to the process of analysis rather than, as in the technical paradigm, minimised or excluded. However it is brought together in a way in which the analyst retains controls the process of problem formulation. By investigation of the numerous conceptual frameworks, the analyst generates problem themes from which an explicit definition of the required system is made. This reductionist philosophy has its motives in making human behaviour explicit for effective technical design.

3.3.2. An Emphasis upon the 'Processes' of Development and Change

Organisational perspectives depart from traditional system building approaches which assume that organisational changes are caused by technology rather than the interaction of technology with its context. Invariably the technology will have an impact upon the organisation; the philosophy behind design approaches is that by managing the change process, these impacts can be planned, as much as possible, to have positive effects. The organisational perspective also departs from the traditional assumption that it is the legitimate role of computer analysts to manage and control the contributions of users to systems building. Mumford for instance, suggests that it is the role of the analyst to act as a facilitator of change rather than impress upon the user his or her own professional opinions. Moreover, Markus argues that the analyst's role as project leader is unfounded because of this person's bias towards tasks rather than the process of change in the organisation. Markus states that the functional manager rather than the analyst should therefore head the project and adopt an '...explicit and active role in problem definition, solution generation and implementation planning'.

An O-perspective assumes that organisational change and learning about systems takes time; this infers that systems development is evolutionary and iterative . By contrast, an underlying premise of traditional system analysis and design methods is that there is a static model of the problem and they prescribe a single strategy of implementation for every context and system. The certainty of the success of the project is contingent upon the technical feasibility of the system's design. An organisational perspective adds to this that the technical design features should match the problem context and therefore uncertainty may be reduced by managing the process of change.

The mechanisms by which to attain an organisational perspective are varied. However the most frequently mentioned is that of *user participation* in the design and development process (Rees:1988). *Chapter 4* defines a particular approach called ETHICS which was influential in formulating an approach towards expert systems design. User participation is based on the assumption that technology is more flexible than assumed in a technical model and therefore the outcome of its use can be controlled by involving all those who will be affected by it. Technical as well as social objectives are generated from which a set of feasible socio-technical solutions is defined. The motives for participation from an organisational perspective may appear instrumental from a human perspective however. For example, Rauner *et al.* (1990) speak of a form of participation as a means of accessing job-specific knowledge about the production process which should be included in the technical design rather than necessarily customising the design of the system to suit the particular needs and skills of the individual. Similarly, Markus views participation as an organisational device rather than an end objective in itself.

3.3.3. Contribution towards a Conceptual Approach

In terms of contributing towards a conceptual approach, an organisational perspective views system development as a socio-technical process. Two essential concepts are used based on *consensus* and *participation*. It also adds importance to the processes of development as well as design in determining the outcome of the system. Therefore, it is primarily reactive in that impacts are analysed and subsequent changes in the course of development are made. A system evolves therefore and departs from the scientific notion of predefined paths.

In terms of actors, Markus recognised the unsuitability of the systems analyst in defining the organisational context because of the attitudes and bias towards scientific models and technology and also because they advocate a design principle which is independent of the organisational structures in which they are used. By contrast, the 'organisational manager' is more sensitive to the political and social setting and therefore is more suited to the defining system boundaries and constraints.

A further contribution of the organisational perspective is that it adopts a more eclectic design philosophy in its attitudes towards the use of tools and techniques. It makes use of hard and soft approaches concurrently to evaluate feasibility and define design requirements. The decision over whether to use hard or soft processes is based upon the organisational context of the problem. Once a schema has been decided, the organisation of development and users' roles are specified. However, the essence of an organisational perspective is that the infrastructure is flexible and dynamic enough to respond to changes in emphasis or socio-technical setting.

As with a T-perspective though, there are problems in a single dimension approach. For instance, an organisational perspective adopts an aggregated view of the individual as a social entity rather than as a human entity. The explanation of impacts is therefore reconciled in terms of technology, structure, culture and politics. The individual per se is thus seen as a component of these- 'systems user', 'systems designer' and so on- and individual impacts are defined in terms of social interactions, such as job content and job opportunities and political influence. A 'good design' is one which matches organisational features well rather than necessarily meeting personal needs. Attitudes towards the individual therefore are instrumental as a means of achieving effective socio-technical designs rather than inherently humanistic. Furthermore, although this perspective recognises that there are different individual perspectives of an organisational problem, it adopts a consensus design model to reconcile potential conflicts because it argues that in a practical socio-political environment, it is not possible to optimise design around individual needs. This viewed is reflected in Markus's notion of an 'interactive perspective'(1984) in describing the individual as an organisational and political actor.

In short, an O-perspective defines a situation context in organisational terms of structure, politics, culture and impacts which although useful do not completely animate the problem nor do they describe all the relevant features of the setting in which the system will be applied. It is particularly flawed in its analysis and understanding of the individual which can only be rectified by considering individual needs from an individual perspective.

3.4 Contributions from an Individual Perspective

This perspective is closer in many ways to a technical model than an organisational one because it seeks to optimise design, but using human criteria and specifies human goals before the design process, rather than accept that design is a compromise of socio-technical factors. Technology is viewed as a human facilitator with the implicit belief that human-driven systems are feasible through the innate flexibility of technology to be customised to individual human goals. Organisational concepts such as participation are adopted therefore, but orientated around individual rather than organisational objectives. Like the technical and the organisational, the individual perspective is composed of a conception of the problem and a subsequent approach towards development.

3.4.1. Understanding the Problem Context

An individual perspective towards development views the problem in human terms. Central to this perspective are debates on the interdependency between explicit and implicit forms of skills and knowledge. Goranzon (1988) defines how under a T-Perspective, prominence is given to explicit or 'propositional knowledge' which is acquired through the theoretical study of an activity (such as formulae and models), whilst tacit knowledge (such as intuition, practical skills and knowledge by experience) is discouraged. By contrast, 'human-centred' concepts put forward by Cooley (1987) and Rosenbrock (1981) for instance, provide a conceptual approach to problem-solving in which people and computer collaborate in processing and communicating knowledge. The computer deals with the explicit part of knowledge and co-operates with human skill to make it more productive rather than attempt to eliminate that skill. The problem is therefore conceived in terms of human worth and the potential for technology to increase this in the tasks performed.

3.4.2. An Individual Philosophy of Assessment and Development

Attempts to incorporate the individual perspective in development have adopted two distinct concepts. Both acknowledge the tacit dimension of knowledge, but address the problem at different conceptual levels:-

i) Human 'Engineering' Concept.

This model as a particular expression of the individual perspective has become popular through the use of ergonomics, user interface design and human-computer interaction in expert systems development. For example, Diaper(1988) describes a 'people-orientated' approach for requirements analysis; Hart(1986) mentions the usefulness of human factors engineering in knowledge elicitation; and Kidd (1985) defines a 'user-driven' knowledge representation technique; whilst Hammond *et al* (1983) apply human factors concepts to the design of system interfaces. Taylor (1988) notes two essential characteristics of these techniques: firstly the process of modelling the technically required human inputs; and secondly reviewing modifications to the technical features of the system in the light of the above. In these approaches, human factors are constrained to a setting characterised by the sequential design of human and technical subsystems. Furthermore, as a sub-set of a technical process, human requirements are not involved in the technology assessment process. Bright *et al* (1989) have criticised this narrow definition of perspective and cite the failure of expert system projects which have used attempted to use these approaches.

ii) 'Human-Centred' Concepts

The concept of human-centred design by contrast is that these "hard" interpretations of the individual are useful, but as a subset of a human *driven* process. Whilst hard techniques normalise individual behaviour through the prescriptive use of design tools and an emphasis upon explicit knowledge, soft techniques promote a more implicit(tacit) and pro-active approach to all levels of development. Human-centred concepts represents a rather 'evangelical' response to the domination of organisational and technical models in technology assessment and development. These concepts are distinguished by their universal application and intended to address the 'core' values of technical assessment and policy making. However attempts to apply human-centred concepts (Rasmussen & Tottrup:1990; Brodner:1988) have focused at a the same level as human engineering concepts they were intended to supplement. So far then, human-centred concepts are limited in scope to discussions on the allocation of functions between humans and machines. This highlights the empirical social and political difficulties of undertaking intrinsic studies in an socio-technical setting. Taylor(1988) describes individual paradigms as 'intangible phenomena' and therefore difficult for organisations and management to formulate and apply. Shelton(1989) moreover describes a naivety in assuming that the individual perspective can be made the prominent perspective in technology development operating within the reality of a commercial setting. Indeed, Cooley(1990) argues that global changes in industrial culture are necessary for the widespread adoption of human-centred concepts whilst Kochen *et al* (1986) state that cultural change may be achieved within the organisation through the explicit formalisation of human-centred goals at the strategic organisational level.

The contention and lack of cohesion lies in the mechanisms for human-centred change rather than the integrity and intentionality of the concepts. Human-centred concepts have heightened the importance of the individual perspective and therefore contributed significantly towards a multiple perspective concept for the development of expert systems.

3.4.3 The Contribution of I-Perspectives

The individual perspective focuses upon issues of skill, self-actualisation, human worth and the human use of technology. More specifically, the contribution of an I-perspective may be expressed in two ways:-

> a) The user is the subject rather than the object of design and therefore should be given the opportunity to adopt a participatory rather than passive role in the process of dialogue in shaping technology. Furthermore, there is scope for human involvement beyond the design of technology-human interfaces.
>
> b) Organisational and technical concepts have defined normative models of the individual ('organisational actor' and 'rational actor' respectively), which allow extrinsic assessments at the social or technical level. By contrast, the individual perspective requires that intrinsic assessments are undertaken specific to each individual.

3.5 Defining Methods by which to Combine Perspectives

It is clear from the above discussions that the three main perspectives, technical, organisational and individual or human each have important and necessary contributions towards a conceptual framework for development; but singularly, they are incomplete and have a number of weaknesses. A concentration upon a T-perspective has been described in particular detail because it remains the dominant focus in expert systems assessment and development.

If each perspective adopts distinct models of assessment, the question remains how might they be combined to provide a complete picture of any given situation ? One problem is that every development situation is unique and the weighting of perspectives will therefore vary according to the stage of development and the level in the organisation at which the study is targeted. A second problem is with respect to what should the balance of perspectives be determined (the technology or the organisation for example)? In both cases, the work of Markus and Robey (1988) is of help. These authors provide a model of causal agency which distinguishes between two 'forces of change' (*sic*). In the first, the technology is viewed as the cause of organisational change and constrains the behaviour of individuals. This they call a 'technology imperative'. Thus, for example, human-centred and machine-centred concepts and the belief of a 'technology-push' from the market to user organisations, all of which have been described in this chapter, all adopt an implicit technological imperative because arguments for change centre around the design and impacts of the technology whilst overlooking the potential for organisational and political mechanisms to define an alternative option. Therefore in this imperative, perspectives are combined with respect to the technology. By contrast, Markus and Robey define a second 'organisational imperative' in which the role of technology and of organisational 'actors' is to satisfy and fulfil organisational needs. Perspectives are therefore combined with respect to the organisation and in this approach, the functioning of the technology and people is of secondary concern as long as organisational goals are achieved.

A problem in both these approaches though, is that the perspectives are not really combined because each perspective tends to concentrate upon different levels of analysis, what Markus and Robey have called macro and micro theories. This is of limited use for the assessment of expert systems because it results in 'discipline-based' rather than 'cross-boundary' analysis. Thus, for example, introducing ES technology into the organisation (macro-level issue) may affect the skills and competence of end-users (micro-level issue). Therefore by combining levels of

analysis, macro and micro, attention may be paid to the individual perspective whilst discussing organisational, societal and technical issues.
From an analysis of theories which adopt these imperatives, Markus and Robey conclude the need for an alternative 'emergent perspective' which offers a seemingly straightforward solution to the dilemma of how perspectives are combined. These authors argue that change emerges from an unpredictable interaction between technology and its human and organisational users. The subsequent emergent perspective identifies that the relationships between cause and effect is not predetermined, but possible among a number of alternative outcomes and only under certain conditions. Since behaviours and outcomes cannot be predicted therefore, the objectivised intention of structured development is replaced by a more informal and interactive process of change.

By not acknowledging a 'dominant' cause of change, and accepting that technology has no inherent characteristics which determines change but that it is defined by the context of its use, itself determined by the change process, emergent models differ qualitatively from the causal arguments of the technological and organisational imperatives. An emergent model also suggests that the ability to manage the 'change process' requires a dynamic understanding of organisational and human processes together with the features of expert systems technology at both a macro and micro level.

3.5.1.. Towards a Conceptual Framework for Assessment and Development

The contribution to thinking of emergent theories has been in understanding the possible relationships between ES technology and organisational change. It departs from single perspectives and imperatives to define the need for mixed levels of analysis from different perspectives. However, since emergent theories are orientated around defining theoretic structures of the change process, it has defined solutions to deterministic concepts in terms of models and structures, although acknowledging the complexity of this approach (Markus & Robey:p.589). The presumption of incorporating all perspectives and relevant settings makes the notion of an 'all-encompassing', normative model unattainable. Rather, that these theories, as with other concepts and approaches described in this chapter, contribute towards an understanding of a development context for expert systems. The implication therefore is for the use of an 'meta-enquiring system' or conceptual framework rather than a model or process which embodies a particular theory.

The availability of conceptual frameworks for ES development is at best limited. McLoughlin and Clark (1988) describe a 'social action perspective' which is essentially a synthesis of emergent and contingency theories which emphasizes the processual nature of change and in particular the processes of strategic choice and negotiation. As with emergent theories however, it is orientated around the process of technological change, whereas the requirement is for a framework which defines the original problem to which the technology is applied. Wainwright and Bowker (1989) have defined a 'holistic framework' whose value is in providing a context for the selection of particular methodologies and tools according to the nature of the manufacturing problem and implementation context. The orientation of this approach however is strongly technical and is intended primarily for the design of Advanced Manufacturing Technology (AMT).

Green *et al* (1987) offer a 'Transaction Approach' specifically for the development of expert systems. It distinguishes between the technical activity involved in developing the ES, and the activities needed to develop 'systems' more generally so that they can be used in the organisation. The authors recognise the necessity for 'societal, organisational and individual contexts' but they relate this to a transactional model of the knowledge engineering process rather than attribute these concepts to a wider framework which addresses the whole development process. Finally, Wood-Harper *et al* (1987) provide a multi-view approach to the development of information systems. However, the notion of a multi-view is made on the basis that through the prescriptive use of selected socio-technical tools, a multidimensional approach to development will ensue. In all these approaches a number of misunderstanding were evident:

> a) An approach which is multi-disciplinary is not necessarily multiple-perspective.
>
> b) There is confusion over the distinction between conceptual models and conceptual frameworks. As Ackermann(1989) notes, the former provides a representation for development and the latter provides a **context** for development. It is the latter which is of concern in this chapter.
>
> c) Most approaches address technological development per se, rather than technology assessment and problem conceptualisation.

In response to these difficulties, three overriding properties were determined for the emerging conceptual approach. First, that the concept should be used as a framework rather than a tool and should endeavour to promote understanding rather than define priorities or techniques. Secondly, the concept should be pertinent to all levels and dimensions of the ES lifecycle from problem conceptualisation to systems evaluation. Thirdly, the concept should promote the integration of human, behavioural and organisational contexts with more established technical assessment approaches. Finally, the concept should not be prescriptive nor relate to a particular aspect of development.

On the basis of these criteria, Linstone's Multiple Perspective Concept was chosen and adapted, and its proposed use is described in the following section.

3.6. Multiple-Perspective Concepts: Theoretical and Proposed Roles

The Multiple Perspective Concept (Linstone:1978), has also been described as a 'holistic thinking process' used to communicate and evaluate different perspectives (Linstone:1984); as a means of reducing the dominance of a Technology(T) perspective in Technology Assessment; and most preferable, as a 'meta-inquiring system' (Linstone:1981) which includes all other inquiring systems, formal, informal, hard and soft, explicit and tacit.

In a slightly modified form of use, Multiple Perspective Concepts (MPC) is intended to serve as a basis for a more balanced interpretation of problems and their settings and therefore a more balanced allocation of hard and soft tools in the assessment and development of expert systems. No claims are made that the approach is 'theory producing', but rather that it produces a better sense of understanding. The approach may be considered effective if it allows the recipient to define organisational problems through a process of enquiry which highlights different perspectives and evaluates the potential contribution of ES in solving these problems. Its use is as, what Linstone calls (p.37) "a conscious, quasi-routine procedure" in that it leads to continual re-assessment of context, cause and effect at all stages of investigation.

3.6.1. The 'Mechanics of MPC'

The phraseology is important with the use of MPC in which perspectives and frameworks are defined rather than models, to emphasize inquiry and understanding of diverse viewpoints rather than their formalisation through models (pp26). Furthermore, Linstone stresses the simultaneous rather than sequential investigation of perspectives: all perspectives are placed on one subject of study and therefore fulfils Markus and Robey's notion of simultaneous macro- and micro-levels and cross-disciplinary analysis.

There are three primary perspectives in the MPC framework. The Technological (T) perspective is used to study the technical element; the Organisational (O) perspective addresses the organisational element; and the Personal (P) element. However, the essence of MPC is that any perspective may illuminate any element. Although for instance, a T-perspective is essential in understanding the technical component of expert systems and defining the potential contribution of the technology to the organisation, the O- and P-perspectives add important insights which amplify the developers understanding of the problem.

There are numerous dimensions to MPC according to the interface between each of the perspectives; these 'settings' are highlighted in **Figure 3.2.**, and are discussed next:

The **Technical Setting** (*Setting(1),* **Figure 3.2.**), represents in this case discussions on the choice of expert systems technology based upon an assessment of tools and problem characteristics, and also the selection of formal, structured design and analysis tools for development purposes. It is also concerned with the physical setting, that is, the physical conditions under which the system will operate. For example, an ES may require special protective casing to operate in a factory environment, or may require a particular locational setting in the production process. An understanding of technical requirements and capabilities is an essential phase in technology assessment.

The **Socio-Technical Setting,** *Setting(2)*, includes situations where the technical and organisation elements interact. It recognises that by changing the technology, there may be intended or unintended organisational and business restructuring. Cost-benefit analysis and Critical Success Factors for example are the technical means by which to evaluate some of these interactions.

The Techno-personal Setting, *Setting(3)*, is based on the premise that technology affects and is affected by the individual. This setting highlights the individual contribution in the design process and is expressed as a factor in constraining design. It also describes the individual's use and effect upon the technology.

The Organisational Setting, *Setting(4)*, acknowledges that the organisation has diverse characteristics in structure (functional, hierarchical) and communications (organic, mechanistic) for instance, and also has its own organisational problems. At a intra-organisational level, there will also be formal sub-groups such as departments and project teams and informal groupings such as cliques which define a particular orientation of people, culture and resources and which interact with other sub-groups in ill-defined and inexplicit ways.

The Individual Setting, *Setting(5)*, recognises that assessment and development requires different individual actors, of different persuasions, interests and professional backgrounds. In addition to have specific needs and personal goals, they will also behave and react differently when alone than when co-operating as members of an organisation.

The Political Action Setting, *Setting(6)*, refers to the interplay between the organisational culture of the firm, groupings within the firm and individual attitudes and needs. The political setting and the political organisation of change has been identified as a major constraint in the implementation of technology (Markus:1984). Hence, the innovation process may be used to achieve group or individual objectives or safeguard group interests: it is precisely such settings that add uncertainty to the change process.

The Decision Focus, signifies the culmination and indirect consequences of all settings and perspectives and focuses upon the boundaries of where decision-making and assessment should ideally take place (this is represented by the shaded area in *Figure 3.2.*). However, it accepts that the decision process will also interact and possibly affect other settings, such as the choice of expert system for instance, which will have recurrent effects upon other settings. This suggests an iterative, dynamic process rather than a one-time specification of requirements.

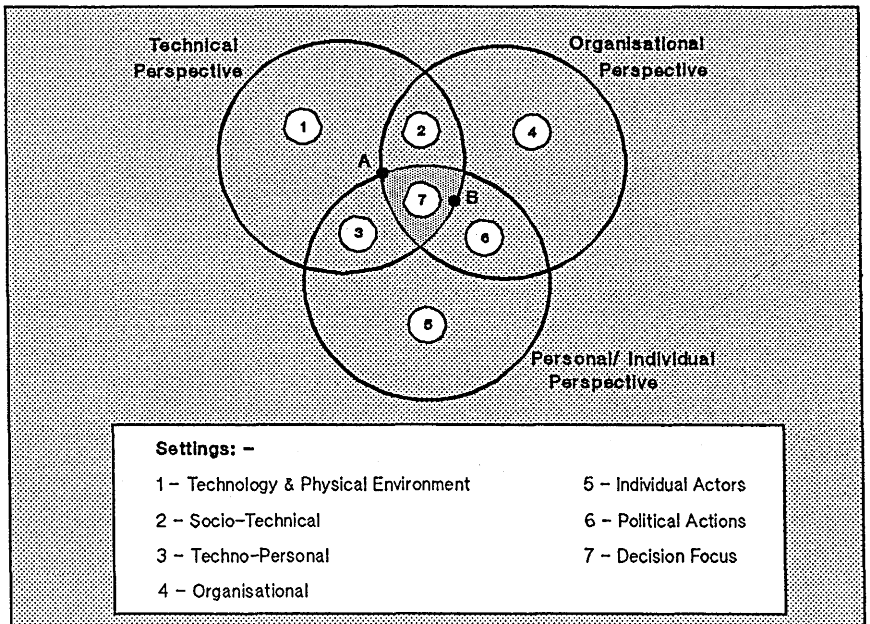


Figure 3.2: A Diagrammatic Representation of the Multiple Perspective Concept.

As an example of its use, it is instructive to consider how the focus of settings might vary according to the type of problem. For structured problems for instance, the dominant perspective is technological and the decision focus will shift to point 'A' in **Figure 3.2**. For less structured, or 'fuzzy', problems where the decision style for example, may be judicial and intuitive, O- and P-perspectives play a more important role in understanding the problem and defining needs, and so the decision focus will shift to somewhere near point 'B' in **Figure 3.2**.. Concern over the balance of perspectives though, and how they should interrelate, is secondary to the concept of enlarging the capacity of users of this framework to see any situation from alternate points of view.

3.6.2. Applying Multiple Perspective Concepts: An Overview of Its Intended Role

In using MPC, it does not necessarily seek for aspects of feasibility of design, although this is an important issue; and neither is it concerned specifically with traditional approaches to technology assessment. As a initial input to the process of development, technology assessment is an important precursor and in its widest sense may be used as a means of avoiding negative social and organisational impacts upon the individual and the enterprise. However, the value of MPC lies in the formulation of problems and therefore is more akin to an organisational than a technology assessment. It therefore provides a setting for the use of technology by animating the reasons behind problems and not only the symptoms. By accommodating the individual, organisational and technical perspectives within its enquiry system, it facilitates the appropriate combination of development measures and helps to identify the appropriate balance of perspectives at any one time in the development lifecycle.

It is not the intention in this section to describe how MPC was used; this is discussed in greater depth in other chapters. However an outline of it's use may be useful in underlining the centrality of its role in the rest of the study:-

a) As a means of highlighting the shortfalls of current methodological approaches and providing a context for the formulation of new processes of development *(Chapter 4)*.

b) To enhance the use of 'IDEFo' (functional methodology) in undertaking an organisational assessment of the firm. Also, used as a 'communicating device' in discussing different points of view from different settings *(Chapter 5)*.

c) As a complement to the use of formal problem specification and application selection techniques. Also influential in the design and presentation of identification and assessment check-lists *(Chapter 6; Appendices VIII & X.)*.

d) As an aid in understanding and defining the application context of candidate expert systems projects in the lead up to the selection of the Office Systems Help-desk, and as a precursor to more detailed requirements analysis *(Chapter 7. and Appendix IX.)*

e) In understanding the different settings and perspectives of technology transfer and as an enhancement to models of knowledge transfer (*Chapter 8.*).

3.7 Conclusions

This chapter has set out to establish a suitable conceptual framework for the assessment and development of ES and it is the belief that a derived use of Linstone's Multiple Perspective Concepts provides a valuable basis for this. Central to use of such a framework is the presumption that the use of tools and techniques is shaped by an original conception of the problem which in turn motivates a particular attitude towards development expressed by a 'design philosophy'.

The evolution of development models has seen the rise of dominant, single perspective concepts which have been predominantly technical in nature. A technical perspective is inherent in the process of problem conceptualisation, technology assessment, design and development. It is also used to reconcile social and organisational effects from a similar technical viewpoint. This influence is identifiable by the attributes of formalisation, externalisation, rationalisation and objectivity.

Reactions to a technological perspective, though providing necessary insights and tools, fail in a similar way by viewing the problem from the same single dimension, although clearly the effects are different for each perspective. Organisational models for instance emphasize the influence of the processes of change in shaping impacts, but hold an aggregated, organisational view of the individual in contributing to this change. Organisational concepts also refer to 'social impacts' in which there is a retrospective analysis of the social effects of technical change. This approach is limiting because it reacts to change rather than attempting to define personal and group needs for example in a more up-front or 'pro-active' manner. Organisational models are therefore sometimes called 'extrinsic' because they operate outside rather than within the change process. By contrast, individual models are essentially 'intrinsic' because the motivation for change and the subsequent approaches to development are defined on the basis of individual needs. This too generates problems though because assessment and investigation tends to concentrate at the human-technology interface with a disregard for the causal influence of organisational features.

Clearly a balance is required between what are extremes in a spectrum of possible viewpoints. At one extreme, technical and human models focus upon a 'one-time' specification of needs from technical and human perspectives respectively. The reality of expert systems glimpsed at in the last chapter shows that, because of the specific virtues of the technology and the subsequent choice of methods adopted, all development needs cannot be specified at the beginning of development. Similarly, the reality of manufacturing industry dictates that more value is placed upon business worth than human worth. At the other extreme come organisational models which promote an evolutionary picture of change and state that requirements cannot be predicted in advance, but rather that they emerge from specific social and technical settings. This contrasts significantly therefore from technical and human models and this distinction is drawn again in the next chapter but with respective to tools and techniques rather than perceptions and concepts. These differences do not mean though that concepts deriving from an organisational perspective such as consensus, participation, and attention to the processes of change, cannot be used and applied with other concepts derived from other perspectives.

The rest of this chapter was then devoted to identifying an approach which would facilitate the integration of perspectives in an appropriate way. Contingency and Emergent theories are useful in that they expose the weaknesses of single paradigm development approaches, but fail to offer a viable alternative because of their emphasis upon models and techniques rather than the value of insight and appreciation of context which are attainable through conceptual frameworks. Numerous other approaches were also considered but most failed because of their undue emphasis upon design rather than higher levels of analysis, notably problem conceptualisation and organisational and technology assessments.

Linstone's Multiple Perspective Concept was chosen and adapted because it was interparadigmatic, in other words, it places equal emphasis upon technical, organisational and individual perspectives and their respective settings. Only through a process of enquiry and understanding of all perspectives is a focus then chosen. The conceptual approaches to design and problem understanding outlined in this chapter are all likely to be useful at some stage in the development lifecycle: the intentional use of MPC is in defining emphasis and contribution of each at any one time and context. The potential contribution of using MPC in this way is that it combines not just different perspectives of the problem and the subsequent role of technology, but also that it combines different processes and levels of analysis: extrinsic and intrinsic, hard and soft, formal and informal for example.

As a conceptual framework though, MPC provides no conception of how difficult it is to actually combine perspectives in practice. For example, in the current industrial climate, some of the goals of a human perspective such as human-centred technology development, however valuable, are unlikely to be achieved unless there are cultural and societal changes. This does not diminish the objectives of MPC, nor does it downgrade the value of MPC in its goal of promoting multidimensional analysis. However is does convey a 'realism' of what is achievable and suggests that it is therefore necessary to place the use of MPC within a context. MPC is simply a 'device' for communicating that every situation may be conceived from different viewpoints, each of varying importance according to the settings and contexts of the situation itself. One should not try to 'intellectualise' its use; indeed it is very difficult to do so because the process of applying MPC is almost sub-conscious in that it provides a 'frame-of-mind' for undertaking assessment and development rather than embodying a particular model or method. To use MPC effectively therefore also implies that the individual is required to be receptive to the framework in the first place.

Chapter 4.

An Assessment and Development Framework for Expert Systems

4.1 Introduction

A mapping of Chapter 4 is outlined in Figure 4.1. The structure of this chapter is very similar to the last in that it explores the influence of a technical perspective in expert systems design and the potential contributions of organisational and personal perspectives. However, it does so at a methodological rather than conceptual level in terms of the choice and use of assessment and development tools and techniques. Moreover, where Multiple Perspective Concepts (MPC) provided a means of combining perspectives, this chapter describes a development framework which achieves the same purpose but again at a tool level.

The chapter thus begins by looking at the 'traditional' approaches towards expert systems (ES) development. In structural terms, two distinct 'schools of thought' emerge from the literature. The first and dominant opinion is that expert systems are unique and should be considered distinct from other computing approaches with the essential difference being the preeminence attached to prototyping and evolutionary development. A second contrasting view is that expert systems are 'just another' computing tool and therefore many of the tools of software engineering may be applied. Between these two extremes, this chapter adopts a central view which states that both are equally valid and necessary within a hybrid approach. In each of the three models of development, associated methodologies are described. Since a large body of literature is referred to, much of this detail has been annexed to Appendix III.

However valuable a hybrid approach is in terms of improving development, it stills suffers from the same limitations by operating within the bounds of a technical perspective which were described in the last chapter. As in the last chapter also, a great numbers of enhancements may be made to this basic technical model by considering methods and processes which have their conceptual basis in organisational and individual perspectives. This chapter describes why it is therefore necessary to look beyond the field of expert systems and consider other previously unrelated fields and disciplines. By adopting organisational concepts, greater importance is attached to understanding and defining the original problem context in organisational, business and strategic terms, and of performing technology development in a way more sympathetic towards the needs of the organisation. Furthermore, these concepts stress the importance of the processes of change with a bias towards how implementation is undertaken.

Possible enhancements from an individual or human perspective are described. Two forms of enrichment are discussed. The first is labelled 'human-centred' in which intrinsic human needs constrain the processes of technology assessment and the role of technology in organisations: many of these arguments however are theoretical in nature and necessitate societal and cultural changes to fulfil. A second and more realistically achievable form is what have been termed 'human engineering' factors, where human concepts are applied to constrain the design and development process.

The blend of concepts and associated approaches derived from each of these three perspectives add new insights, focus upon different aspects of assessment and development and also adopt different approaches (for example formal, informal, structured, unstructured and so on). This makes their combination into a single methodology difficult and indeed approaches which claim to be multidimensional still function within a dominant single dimension with the consequent limitations which were described in the last chapter. Rather than speak in terms of methodologies therefore, this chapter adopts a 'framework':this provides a means by which to combine tools and approaches which have their conceptual roots at each of the three main perspectives in a way which is sensitive to the application context. In developing this framework a number of key concepts are introduced:-

a) *Activity framework*: rather than use predefined tools which adopt a single dimension and prescriptive structure of development, an activity framework provides a functional description of necessary activities whilst being independent of tools or methods. Thus this provides a specification of necessary tasks and assessments without defining at this stage the means to invoke them.

b) *Development Approach*: prior to selecting tools, it is necessary to outline the nature of enquiry required of them. Thus this chapter speaks of formal and informal, structured and unstructured, macro and micro and hard and soft analyses for instance. These 'process mechanisms' together with the development context and activity framework provide a complete environment for the selection and combined use of tools, methods and approaches.

c) *Development Context:* the task mechanisms which define which tools can feasibly be used to accomplish tasks defined in the activity framework and the process mechanisms which specify how they should be used are themselves constrained by the development context in which they are to be applied. The development context is analysed and understood using MPC described in the last chapter.

d) *Company Attributes:* essential to using the assessment and development framework effectively is a detailed knowledge of the client organisation to which the framework is applied. This chapter thus speaks of the need to understand the 'emergent properties' of the company and the next chapter addresses this issue in more depth by assessing organisational models which help one to realise this.

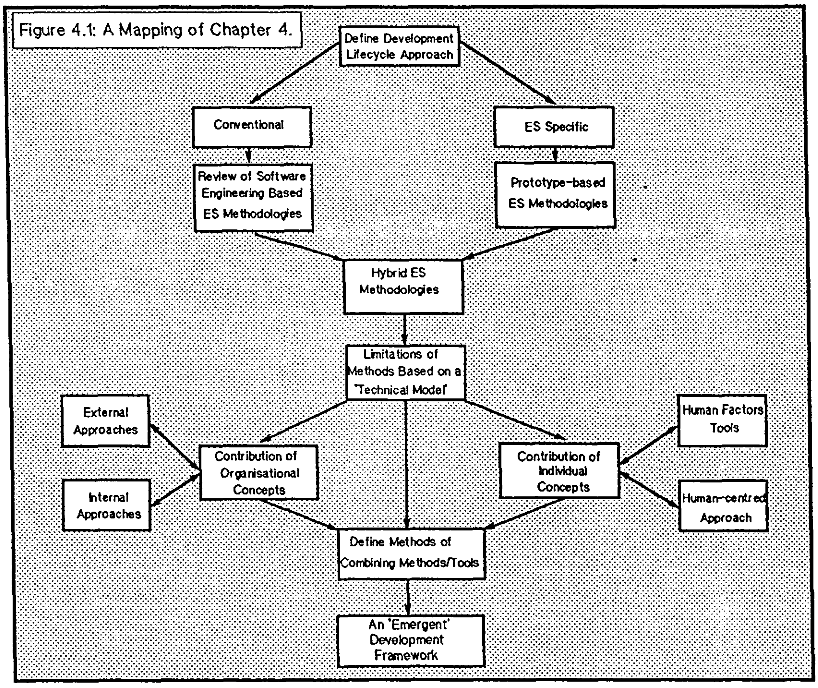


Figure 4.1: A Mapping of Chapter 4.

4.2. A Review of Expert System Development Life-cycles

Table 4.1. shows where research effort and developments in expert systems (ES) have been concentrated. The most widely quoted, though now dated, life-cycle is

Table 4.1. Life-cycle Models of Expert Systems and Traditional Software Development.

Expert System Life-cycles					Waterfall Life-cycle
Hayes-Roth (1983)	**Harmon & King (1985)**	**Lock Lee (1986)**	**Beerel (1987)**	**Turban (1988)**	**Rowen (1990)**
Identification	Selection of an Appropriate Problem	Technical Feasibility	Outline Specification	Identify Problem	Define Scope of System
			Knowledge Acquisition	Select Expert	
Conceptualisation	Development of Prototype	Commercial Feasibility	Knowledge Representation	Select Software & Hardware	Requirements Specification
Formalisation	Development of Complete System	Definition of Problem	Prototype Development	Knowledge Acquisition	Solution Design
				Build Prototype	
Implementation	Evaluation of System	Conceptualisation	Main Knowledge Acquisition	Evaluate for Performance	Implementation Design
Testing	Integration of System	Formalisation	Detailed Specification	Use in Parallel with Expert	Validation & Certification
			Systems Development		
			Testing & Validation	Develop Plans for Maintenance	
	Maintenance	Implementation			Acceptance
			Implementation	Document System	Maintenance
		Testing	Maintenance	Systems Release	Obsolescence
Unstructured					.....Structured

from Hayes-Roth *et al* (1983) who define five basic steps of identification, conceptualisation, formalisation, implementation and testing. Each step is considered interdependent and serial in nature, although an iterative process makes it possible to return to any earlier step. As with nearly all others though, this life-cycle defines *what* steps should be taken without necessarily defining *how* these steps may be undertaken.

The life-cycle of Hayes Roth *et al.* may be contrasted in Table 4.1. with that used in traditional software developments, known as the 'waterfall' life-cycle (discussed later). Sviokla defines two major differences between these (1986). The first is that the ES life-cycle assumes that an initial requirements specification is unattainable and such needs can only be addressed during the conceptualisation and formalisation stages using prototyping as the main mechanism. Secondly, that the ES life-cycle fails to recognise maintenance as a development issue. As Sviokla notes, despite the fact that 60-80% of all programing effort (over the lifetime of the system) is expended on maintenance, this responsibility is left to the end-user.

From the basic model of Hayes-Roth et al., there have been a number of additions and enhancements. Harmon and King(1985) for instance, distinguish in their life-cycle between prototype development as a precursor to full systems development. These authors also define a specific set of tasks to be undertaken during each stage and introduce such requirements as interface design, systems integration, resource planning and maintenance. Similarly, Waterman (1986) proposes different stages in prototype development, distinguishing between demonstration, research, field, production and commercial prototypes. Waterman also provided limited guide-lines to help managers to identify technically feasible problems for ES. This work was extended by Lock Lee (1986) who considers the commercial as well as technical feasibility of a proposed system, although again little detail is given on how this is achieved.

More recent life-cycles, such as those of Beerel (1987) and Turban (1988) shown in Table 4.1. and others such as Rolston (1988), have added greater detail still to the life-cycle in areas of problem identification, resource planning, design specification and development control. A significant trend from a comparison of these life-cycles is that the rigid differences between the early ES life-cycles and waterfall life-cycle become more equivocal; with later versions not only beginning to use the same terminology, but also adopt and incorporate some of the same stages as the waterfall life-cycle. For example, Table 4.1 shows how Turban speaks of 'maintenance planning' and formal documentation requirements and Beerel defines the merits of pre-development specification and phased documentation.

Despite specific differences and contrasting terminology, there are also a number of common attributes to the ES :-

Stage 1: problem description - This stage varies in scope according to the criteria adopted and the importance attached to this phase. However Wilson *et al* (1989) identified three common elements to problem description:-

a) the identification of problems where the use of ES within an organisation will provide the leverage to overcome them cost effectively,

b) initial problem analysis. A description of the problem will be assessed as to whether it is amenable to a ES solution,

c) description of problem requirements. The intended role of the system and the requirements it should meet are identified and a user model is established to guide future development.

As, Wilson *et al.* note, in practice, prototypes are developed at this first phase to present capabilities of ES to potential users and consequently a sequence of successive prototypes is often developed. By contrast, Waterman argues the case for a formal feasibility study during this stage. According to this approach. for a problem to be feasible for an expert system, it must past the test of three separate analyses based on justification, necessary requirements and 'appropriateness'.

Stage 2: Knowledge Acquisition (KA): this process has been described as the 'bottle-neck' in the ES life cycle (Welbank:1983) and is the process of gathering the desired knowledge from the experts and formatting it into a knowledge-base. KA takes place mainly during design, but is also necessary during implementation and maintenance. Having located domain experts with justifiable expertise, KA may involve either single or multiple experts. There are many techniques available for KA, including interviews, case descriptions, performance analysis, repertory grid and protocol analysis amongst others. The diversity of techniques and the arguments for using them in different circumstances is reviewed in detail by Neale(1988) and will be summarised in *Chapter 7.*

Stage 3: Prototype Design & Development: in identifying appropriate forms of knowledge representation, inferencing strategy, design of user interface and definition of user requirements, a prototype system may be developed. The argument for adopting this approach is that in an expert system, many of the problems are not revealed until actual implementation since, unlike traditional software projects, the exact specification of what can be done, and how, is not precisely known. There are diverse arguments for and against prototyping but, whatever the view, the developer will be limited to using knowledge representations available in the prototyping tool, rather than the natural representation of the domain knowledge or expressed logic of the expert.

Stage 4: Implementation & Refinement: when the knowledge acquired and the specification of design have reached an acceptable level of validity and completeness, the full system is developed. This may be a progressive development of a prototype or a completely new design and implementation.

Stage 5: Testing: when the system is developed, it is transferred from a development environment to a trial delivery environment in order to assess how it meets the requirements. The delivery environment may be simulated by the expert, or the ES may be used in parallel with the present function of expertise in order to compare the results and verify the systems use.

Stage 6: Installation & Systems Integration: the technical system is installed when it reaches an acceptable level of stability and quality. Turban for instances, suggests that rule-based systems should be installed when the knowledge-base can handle 75% of cases. The system requires integration in an organisational as well as technical environment. This will involve training, possible restructuring and documentation support.

Stage 7: Maintenance: ES require continual maintenance because in many domains of expertise, knowledge and working practices may change with time. A major problem in maintenance is defining and identifying where modifications and updates should be applied. It is insufficient to simply add rules as this may lead to unforeseen conflicts and may also decrease the competence and performance of the knowledge-base. Maintenance is the least well understood aspect of the life cycle and there are two recurrent themes which will be discussed in future sections: first, whether it is possible to anticipate maintenance problems at the problem selection stage ; and secondly, whether the costs, time and effort of maintenance can be estimated in advance.

The ES life cycle represents a narrow view of a methodology, with its contribution most well understood in terms of development tools and techniques aimed at achieving technical feasibility and ease of design. Furthermore, each stage is presented only at a high level of abstraction and provides no guidance on the use of these tools.

4.2.1 An Evaluation of ES Life-cycles

A problem with development life-cycles is exactly this; they are only concerned with the development of the technology and overlook the importance of a technology assessment process so strongly emphasised in Chapter 3. Thus, as Jamieson and Szeto point out, 'The methodologies reviewed from the literature seem to lack consideration of the activity of project selection....The methodologies assume that an application exists and the justification will be carried out at the feasibility stage' (1989). Common to all of the development life-cycles is that they operate within a predominantly technical perspective by drawing attention to the design and construction of the system. what Hickman defines as issues relating to the 'internal' operation of the system, to the exclusion of other concepts and processes (1987). Although there is pressure to develop alternative approaches concerned more with the introduction of ES into organisational rather than technical settings (Guida & Tasso: 1989), the bulk of methodologies are structurally still reliant upon the concept of the ES life-cycle.

As well as these limitations, what is also clear from the above analysis is that, even at this level of abstraction, there is no definitive life cycle model of development that has established itself as a standard Born (1989). Moreover, Wilson *et al* point out that most expert system developments in practice utilise a 'colloquial life cycle model'. Therefore, different developers identify different stages and emphasise different decision points during the development process. Meyer for instance, focuses upon the integration of the ES with conventional systems (1987); and Hart (1986) and Welbank(1983) concentrate upon the knowledge conceptualisation and elicitation processes. At whatever level the heart of debate is concentrated, two central themes emerge as being critical: the legitimacy and extent to which ES development methods may adopt conventional software engineering principles and the preeminence afforded to prototyping during the design process. In both cases, it is necessary to consider the conceptual and structural differences between expert systems and conventional software developments.

4.3. Two Models of Expert Systems Development

Two basic models of ES development may be defined from the above. The conceptual foundations of the waterfall model are based on the notion of what Partridge and Wilks (1987) call a 'SPIV' (specify-prove-implement-verify) model. Implicit in this view is a belief that in order to be commercially viable, expert systems should adopt traditional software engineering concepts. By contrast, expert system life-cycles, particularly the earlier ones of Hayes-Roth *et al* and Harmon & King shown in Table 4.1, stress the 'uniqueness' of expert systems and are based upon a 'RUDE' (Run-Understand-Debug-Edit) model of development. The contribution and arguments behind each of these generic models is outlined below from which their derived methodologies are described.

4.3.1. The 'SPIV' model of ES Development

In conventional systems, development follows a sequential process of discrete and pre-defined stages known as the 'waterfall life cycle'. The significance of SPIV approaches to ES development is that the waterfall life cycle may be used as the structural basis for development. Thus, by defining the components and operations of the waterfall cycle, the rationale for its use in ES development may be understood. As with the technical 'model' of development described in Chapter 3, a number of key assumptions are made in this model:-

a) The problem is well defined and predictable.
b) The problem is capable of being proved correct.
c) A complete understanding of the systems requirements can be derived and described at the start of the project.
c) It is possible to define and prescribe discrete and intermediate steps each of which can be independently realized and verified before moving on to the next.
d) It is assumed that the output of each stage can be exactly traced back to the original requirements.

Like the ES life-cycle, there are many derivatives of the basic waterfall life-cycle; however Rowen's model, shown in Table 4.1., seems to include most of the basic activities (1990 p11.). Each stage in the waterfall model development is intended to produce a deliverable which can be verified against previous stages and validated against the client's requirements. As the systems progresses through the life cycle, it becomes more complete, but also more complex so that changes at later stages are more costly than those at early stages. The benefits of adopting this sequential structured approach is that it ensures quality and control over the design process. Furthermore, it facilitates the use of formal development methods and tools to support the development process, and thereby assists in estimating costs, time and effort requirements through a project life cycle (Boehm:1981).

Rowen notes that the waterfall life cycle is a useful model for the systems analyst to adopt because specific documents and deliverables are associated with each step of the development process. It is in effect a reductionist or 'divide and conquer' means of dealing with the complexity of large projects. For the development manager too, the model provides known objectives, and the means by which to measure these, together with formal control mechanisms.

4.3.1.1. Methodologies based on the 'SPIV' Model

This class of technical development accounts for the short-comings of ES development in terms of poor specification, planning and control, and utilizes software engineering principles of predictability, repeatability and reduction in order to resolve what it sees as being the main causes of the problem. Thus, Wilson et al (89) argue that ES have been developed using 'informal and random' views of the systems life cycle. Tools have been developed to support some of the more technical stages of the life cycle in an 'undisciplined' manner. The emphasis by such authors, as with Hayward (1986) and Hickman(1987), is that the life cycle needs to be more closely planned and controlled, particularly at the stage of prototyping. SPIV models of the life cycle are adopted with the implicit view that development process can be fully specified before implementation. Furthermore, structured engineering techniques, such as requirements analysis, systems analysis and structured design are used.

Two methodologies are reviewed in Appendix III as being representative of this class. The first is a life cycle approach called KADS resulting from the ESPRIT project 1098 (Hayward et al:1987) and is based directly on the waterfall model. The second methodology by Keller(1987) proposes an approach based on Yourdon's structured systems analysis methodology. It provides guide-lines on how ES should be integrated into the organisation at the information level and is orientated more towards design analysis than the full development cycle

In both approaches, development is viewed as a modelling activity, with different levels of abstraction defined, from data models and design models to conceptual models. Furthermore, the differences between SE and ES development is expressed in terms of the latter requiring more complex design models rather than the two being conceptually or methodologically different.

The strengths of these methods is that they allow the developer to closely control and monitor the transformation of elements. For each phase of the model, precise goals, outputs, tasks and operational activities are specified. This accountability, as Hickman points out, makes verification, debugging and enhancements, and maintenance more routine and predictable procedures (1989). The emphasis upon specification and control at each stage of the development project offers a number of benefits, particularly in that they mirror the limitations of ES development reliant upon prototyping. The possible contribution of applying these methods to ES development include the following:-

i) improved capabilities in planning the project
ii) improved capabilities in estimating development requirements
iii) improved capabilities in sizing the likely software product
iv) improved capabilities in estimating hardware requirements
v) the availability of documents for monitoring and control
vi) the ability to apply quality management techniques to development

As well as the above, Hickman also argues that ES will only be used effectively when they applied and implemented by the present software engineering function in the organisation and so the development tools used must be compatible and integrable with existing tools. However, in that both methodologies adopt the waterfall model to varying degrees, they also inherit many of the same limitations as conventional software techniques:-

i) The methodology is design and analysis orientated and does not incorporate SE techniques for verification and validation across all stages of the life cycle.

ii) The methodology does not incorporate prototyping and follows a strictly incremental path (Wilson *et al*).

iii) There is no statement on the applicability of the methodology to different problems and circumstances because requirements analysis governs the movements within the methodology only: it does not assist in the selection of the original problem and tools (Guida & Tasso).

iv) KADS particularly, is a substantial life cycle based methodology, with complex and highly prescriptive technical and organisational procedures. In this sense the methodology may impose disciplines and structure which may be unacceptable or unattainable to the organisation.

V) The KADS methodology outlines project management guide-lines which are useful, but these are orientated more around the use of the methodology as an 'innovation' itself, rather the implementation of the project.

Experiences in using KADS have shown greatest success in applying planning, quality assurance, and documentation procedures for project monitoring and control, although this covers the analysis and design phases only and the methodology does not support prototyping. Wilson *et al* claim that neither SE techniques or management theories offer a solution to the problem of estimating the effort, time scale and manpower for ES development and that methods of metrication can only be based on empirical data which is not yet available. However in the sense that SE techniques add accountability and impose a design discipline, the development process itself may become more predictable making the process of estimating easier.

Methodologies based wholly on the waterfall life cycle are highly prescriptive and very complex: the complexity results from the difficulties in modelling ill-defined structures and in this sense suffers from the omission of prototyping. Little attention is devoted to pre-implementation issues, such as problem identification and business planning, but the philosophy upon which these methodologies are based assumes that these issues may be addressed, as with conventional software development, within a technical perspective.

4.3.2 The 'RUDE' Model of ES Development

Partridge and Wilks note an inherent pressure for expert systems development to adopt a particular model for the reason that all software development should proceed along a certain path; and that applications ought to be developed in areas where it is possible to specify behaviour completely and in advance of the programming process. These authors quote Hoare (1981) as being the epitome of the 'SPIV' sensibility:-

> "A lack of clarity in specification is one of the surest signs of a deficiency in the program it describes, and the two faults must be removed simultaneously before the project is embarked upon. "

However, a number of authors stress that there are certain differences between ES and conventional software: these are expressed in different ways and with varied emphasis. For instance, de Jong (1988) expresses these differences in terms of the components of the end software product: Partridge (1988) makes a distinction between the two on the basis of the problems they address; and Bader *et al* (1988) focuses upon the differences between the waterfall model and the ES life cycle described earlier. In each of these studies, there are common themes and these will be summarised: -

a) **Expertise:** An essential differences between ES and conventional programming development are that with ES make use of human knowledge , experience and judgement rather than with procedures (Hart: 1986). As a consequence, stages in the development life cycle are less well defined than in ordinary systems development and the idea of a feasibility study becomes more difficult to implement. Moreover, stages overlap more and is becomes difficult to estimate time, effort and costs. Problems are ill-structured, i.e. solution procedures differ on every problem, but may be solved routinely by human specialists or experts. It is not possible to specify design requirements prior top implementation and so prototype designs are built in order to fully understand design requirements and implications.

b) **Feasibility & Requirements Definition:** Although information technology is a 'moving target', many of the data processing (DP) applications used in industry are well established (Bader *et al.*). Issues on feasibility are therefore not solely concerned with whether it is technically possible to build the proposed system, but instead how long it will take to build, its costs and relative benefits. With ES there is the added uncertainty of whether the design proposal is technically possible to achieve. Worden (1988) notes that the risks at the start of an ES project are much greater than those in conventional systems projects for three reasons:-

> i) The complexity of capturing and representing knowledge in a way that gives acceptable performance.

ii) It is not known in advance what level of performance is adequate for a particular role, or how much knowledge is required in order to achieve a particular level of performance.

iii) There is frequent misunderstanding over the functional role of the ES

As Worden notes, 'these sources of uncertainty follow not from the immature state of ES technology and tools, but intrinsically from the empirical and heuristic nature of the knowledge embodied in an ES.' This makes problem analysis and problem selection activities an essential precursor to the conventional programming concept of requirements analysis, although Worden adds that such uncertainty cannot be reduced through tools and techniques but only through experience. The implication from this is that ES development is essentially an experiental, trial-and-error process of iterative redevelopment.

d) Performance Measures: A conventional system can have a precisely specified set of output requirements. By contrast it is considered too difficult to circumscribe the performance of an ES because of the uncertainty of eliciting all the required knowledge from the expert and representing it in an optimal format.

e) Experimental Prototyping: Even at the feasibility stage, it is recommended that an exploratory prototype is built (Waterman). The intention is to assess the likely costs and requirements of a fully operational system. If the prototype proves acceptable, then it may serve as an active specification for the operational system. In this way, the exploratory use of prototypes is used instead of the waterfall model stage of requirements analysis. It is possible moreover, that prototypes developed at the feasibility stage are discarded once concepts and ideas have been formulated and a new operational prototype is constructed. In a data-processing environment, this would arise only in exceptional cases (Bader *et al*).

f) Analysis: A necessary first stage of conventional systems development is analysis of the existing manual system in order to produce a functional model of the activities encompassed within the scope of the problem, from which data flow diagrams are constructed. The latter defines data movements and storage and retrieval transactions in order to establish desired data outputs from activities. Jackson Structured Programming, Z, Yourdon, and Demarco are some of the more common *formal methods* used to perform these tasks (see Avison & Fitzgerald: 1988 for more information). Where conventional analysis is concerned with data, the equivalent ES analysis is concerned with understanding and eliciting the problem-solving activities of the human expert using knowledge acquisition tools and techniques. Although these tasks are broadly analogous, Sharman and Kendall (1988) view the role of knowledge acquisition as displacing conventional analysis and design rather supplementing it: moreover, the application of formal methods in knowledge acquisition is considered inappropriate by these authors.

g) Design: The complexity involved in the design of conventional systems, as Bader et al describe, is expressed in terms of size and project management. By contrast, although these problems may exist in larger projects, complexity in ES design arises due to the need to analyse and model human expertise in a structured way in order to devise an executable knowledge representation schema (whether it be rules, frames or semantic networks and executable in the sense that this schema may be encoded).

h) Implementation: In the waterfall model of conventional software development, stages follow one another sequentially so that design decisions made in the early phases of development are not reviewed until the code has been evaluated much later on in the life cycle. It is assumed that analysis and design phases are able to generate a set of program specifications which are complete in the sense that they satisfy the conditions specified in the requirements analysis stage. In ES design, it is assumed implicitly, that iteration is an inevitable and necessary part of analysis, design and coding in order to identify, specify and represent the range of problems and cases defined by the expert. Iteration is also considered essential in the validation of designs by experts and potential users.

j) Testing: In conventional systems development, testing is used to verify that performance objectives and user requirements specified during the requirements analysis phase, have been met. In ES development, testing is a more complex, lengthy and ambiguous process of verifying with the expert that the knowledge encoded in the knowledge-base is complete and correct and that it performs the tasks required of it satisfactorily in the eyes of the expert.

k) Maintenance: In conventional software, maintenance is unnecessary (but for the removal of bugs), so long as the original specification is met. In practice, demands for improved functionality and performance result in frequent redefinition of the specification and therefore upgrading. In expert systems, maintenance is a continuous process of enhancement rather than redesign, although as with DEC's XCON expert system, the knowledge-base may become so large and unstructured that redesign becomes necessary.

For these reasons, Partridge (1987) argues that the SPIV model is not viable for ES development and offers an alternative development model, 'RUDE' (run understand-debug-edit) which emphasises the central role of prototyping or evolutionary design in development, not in the organisational sense used by Markus in Chapter 3, but in terms of achieving technical feasibility in development. In place of the 'correctness' notion of SPIV, Partridge adopts the concept of 'adequacy' whereby acceptable systems evolve through continual iterations of design.

The essential stage which distinguishes the ES life cycle from the waterfall model is that ES emulates the problem solving behaviour of a human expert from beginning to end. It is thus not only necessary to specify what the human expert does when solving a problem (specification), but also how the expert does it (design and implementation). This implies that knowledge acquisition is an activity that should occur at every phase of the ES development life cycle. The following development methodologies are based on this premise and therefore are juxtaposed to the SPIV model.

4.3.2.1. Methodologies based on the 'RUDE' Model

Authors such as Partridge (1988) view ES as distinct and separate from conventional programming theory on the basis of the importance attached to prototyping. 'RUDE' models of the life cycle are adopted which emphasise the evolutionary nature of development and methods are used which are orientated around prototyping.

A central issue in prototype based methodologies is the extent to which prototyping may be viewed as a development approach in itself, in which case the prototype becomes the end-product, or whether it should only be used as a knowledge engineering 'tool' to augment the design process. The orientation of the methodology in the first case is towards resolving how the prototype should be constructed and when it should be accepted as the final product. In the second case, the methodology is orientated towards providing guide-lines on when to build a fully operational system from the prototype. Methodologies of this genre therefore require sensitivity to these differences. Ince (1989) provides a classification of prototyping techniques, each type placing different demands upon structure, resource requirements and criteria for evaluation:-

a) Throw-away prototyping: this type is used for the purposes of requirements identification and clarification. It is a substitute for the waterfall phase of requirements analysis and is therefore referred to as specification prototyping. At this stage, emphasis is placed on rapid development rather than quality factors such as efficiency and maintainability. Ince notes the necessity to orientate the construction of the prototype about pre-defined evaluation objectives and only address those aspects formulated in the prototyping objectives. The value of this type of prototyping, as Ince notes, is in the process itself rather than the product and therefore most of the effort goes towards evaluating the prototype rather than the design.

b) Evolutionary prototyping: As indicated earlier, proponents of this approach argue that ES once installed evolve steadily, invalidating their original requirements. methodologies based solely on this approach therefore require sufficient flexibility to cope with change during and after development. The process is highly iterative and dynamic; and during each iteration, re-specification, re-design and re-implementation takes place with re-evaluation proceeding each phase. The value of evolutionary design is that there is less risk involved compared to the waterfall model since the impact of early errors is less serious. An additional benefit, is that the presence of a crude but operational system early on in the project life cycle allows the user, according to Ince, to influence the direction of design in a way not possible in waterfall-based methodologies. This 'process-orientated' approach to design depends heavily on the ability of the designer and the functionality of the methodology to accommodate what Ince calls 'modifiability'. It also requires a managed balance between creative design and project planning to be commercially viable.

c) Incremental prototyping: Ince defines a final classification of prototyping where the system is built incrementally, one section or stage at a time. Incremental prototyping is distinct from evolutionary prototyping in that the former is based on one overall design whilst in the latter, the design evolves continuously. It is also used more at the implementation phase than during design and analysis. Thus, although the system is gradually enlarged, it does so in a more well-defined and structured way ,as Ince notes. "Incremental prototyping provides less scope for adaptation than evolutionary prototyping but has the advantage of being easier to control and manage."

Ince's classification has a number of methodological implications. Firstly, it suggests that there are different types of prototyping which are relevant at different stages in the life-cycle. Second, it suggests that each class of prototyping has different functional objectives and performance evaluation characteristics. Third, some forms of prototyping are more structured than others and therefore may be more accepted in 'SPIV'-orientated models of development. Finally, it suggests that each class of prototyping may require markedly different tools and techniques. On this point Ince observed that most prototype tools used are only appropriate for the development of throw-away prototypes.

In practice, a methodology may adopt a number of prototyping techniques each appropriate for a specific stage in the life cycle. Differences between methodologies arise in the importance attached to the contribution of prototyping to the total development process. As an example, two methodologies are reviewed briefly in Appendix III. Both are commercially available and make substantial use of prototyping, although in fundamentally different ways. The Greenwell Methodology (Greenwell:1988) uses a throw-away prototype to help define requirements: in this capacity, the prototype is used to augment more conventional feasibility and analysis studies. Once a satisfactory specification has been formally documented, an operational system is developed using a incremental prototyping techniques. By contrast, the Abacus Method (in Citrenbaum et al:1986) adopts evolutionary prototyping in a highly unstructured and exploratory fashion with the original prototype turning out to be the end product.

The choice and suitability of each prototyping method is contingent upon the perceived technical risk and also the organisational setting in which they are developed amongst other factors. However there are a number of inherent limitations in undertaking prototyping of any description (Diaper:1988) and these may be summarised:

i) Neglect of user issues: despite the importance attached to the 'process' of development in prototyping, Diaper found that end-users are seldom consulted. Rather that prototyping is used to first define decisions on the functionality of the system after which user interface issues are discussed as a separate issue. Diaper refers to this as a 'damage limitation' approach to human-computer interaction (HCI), "..where the HCI professional is required to turn a badly designed system, from the users' point of view, into one that is at least just about usable." A concern argued in *Chapter 3*, is precisely that human issues are appended to the design setting, called 'techno-personal' using Linstone's nomenclature, rather than being expressed explicitly to shape the course of development. This point is important because a number of commentators have assumed that prototyping is implicitly more 'human-centred' than software engineering (see for example Rauner:1990). Infact, there are a number of formal SE techniques which begin with human assessment prior to technical design. Carver (1988) for instance describes the use of Jackson Structured Design (which is a formal design technique) to establish the end-user interface using goal and task analysis. Although clearly, such techniques are limited in scope, it does begin to place priority to human requirements before technical functionality.

ii) Rapid Growth: A second problem, with evolutionary prototyping particularly, is that it may lead to uncontrolled growth of the knowledge base and inference engine. Diaper notes a number of possible consequences of this:-

> a) Even if well defined, the original model of the domain may be obscured by the iterative changes that are made, with the effect that it becomes expensive and difficult to maintain and up-date.
>
> b) The knowledge-base and inference engine are developed in an un-coordinated and piecemeal fashion such that further iteration becomes increasingly difficult and knowledge becomes highly distributed and possibly hidden from the developer.
>
> c) In large systems particularly, relatively simple modifications to the knowledge-base may have unpredictable and adverse effects that are undetectable by non-domain expert users.
>
> d) The system may become what Diaper calls 'unauditable' in that it becomes difficult to identify what the final ES version will *fail* to do because the completeness of the ES is unknown.

iii) Organisational Plan: A third weakness of prototyping, identified by Diaper, is that there is often no implementation plan for the organisation, only for the technology. Curnow(1987) notes that the introduction of an ES will require changes to operating practices of both individuals and the organisation, especially where local expertise of a domain expert is widely distributed within the organisation.

iv) Project Management and Control: Prototyping is valuable in communicating capabilities and potential: and it is essential in understanding user requirements and defining technical aspects of systems design. However, there are difficulties in managing and controlling the process, with uncertainty being the greatest with evolutionary prototyping due to the highly unstructured and iterative process of development. Methodologies based on prototyping as the primary development approach pay poor attention to planning, design documentation and control as Wilson *et al.* highlight:-

> " In many current ES projects, stages in development are not defined with the result that there is little management control, the work of individuals cannot be integrated and software is hard to validate and maintain."

Although there are characteristics of ES which are distinct and separate from conventional programming, there are common objectives such as project planning and monitoring, quality assurance, production standards and documentation. However, by adopting a 'RUDE' model of development based solely on prototyping as the main approach, the potential value of these techniques has not been exploited.

4.4. Combining SPIV & RUDE Models: A Hybrid Approach to ES Development

To reiterate, a major difference between SPIV and RUDE models of ES development is that the former assumes that an exact specification of the problem is possible before development whilst the latter argues that specification is a product of the development process. Development methodologies based purely on the waterfall model are inappropriate because they overlook basic differences between the functioning of ES and SE. Furthermore, the wholescale dismissal of iterative design makes them highly complex and difficult to use. Similarly, methodologies based purely on the RUDE model also have evident weaknesses. It is the belief that a convergence of both models is required in order to produce systems according to contractual time scales and industry standards (Kelly:1986). Bader *et al* suggest that this may be achieved through a 'POLITE- Produce Objectives-Logical/Physical Design- Implement-Test -Edit- model of development based on a synthesis of SPIV and RUDE models described earlier. Thus in order to 'engineer' an ES, the iterative features of the RUDE model need to be incorporated within the structured and controllable SPIV model.

The complexity of such hybrid methodologies in practice is in combining the attributes of both models within the framework of a coherent methodology. There are practical reasons for a marriage of SE and ES techniques. Foremost is that because commercial ES are being integrated with conventional systems (Worden:1988), they will contain both heuristic and conventional components. Secondly, Bader *et al* note that if ES are to gain acceptance in industry then their development should be based on the 'engineering methods' already in use and understood, although this is less of a justification than an indictment of the pervading culture of development. There are also problems involved in combining these models. Wilson *et al.*, for example, mention that a number of ES development methodologies that incorporate software engineering principles have done so at the expense of ES ones. It is necessary for a hybrid approach therefore to recognise the differences in design philosophy underlining the use of these tools.

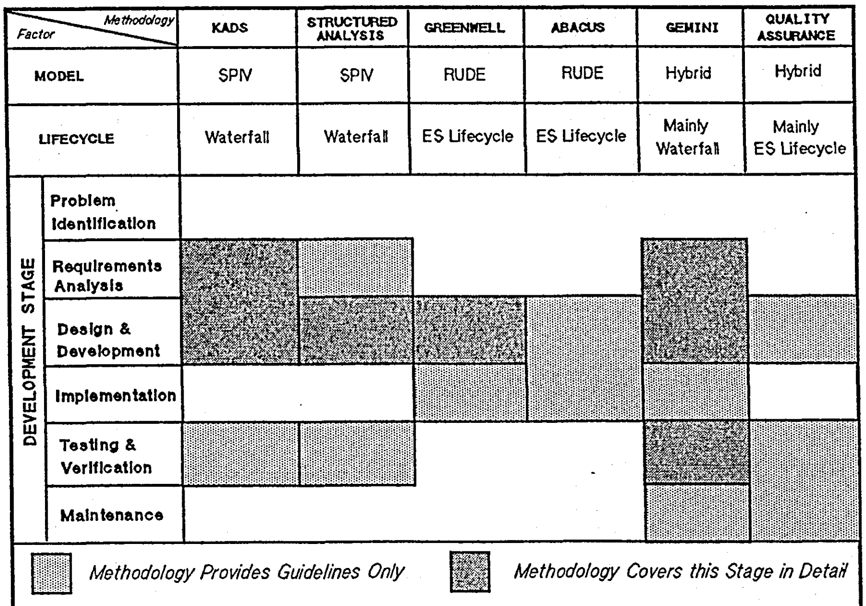
There have been few attempts to develop hybrid models. Kelly (1986) adopts a waterfall model but utilizes prototyping methods to provide greater degrees of iteration and feedback between each stage. Worden defines a similar approach, but proposes an approach in which the SE and ES components are partitioned from each other so that the first version of the system is developed independently of knowledge-based functionality. A third approach has been on the use of SE principles and techniques to manage, control and thereby assure the quality of prototypes though an enhanced life cycle. Two particular methodologies are discussed in Appendix III: the first, 'GEMINI', is based on the hybrid models of

Kelly and Worden in that the basic structure most closely approximates a waterfall life cycle and prototyping is incorporated in a controlled, structured development environment as a distinct and separate dimension to the life cycle. In the second 'Quality Assurance' approach by Born (1989), the focus shifts to a predominant ES life cycle approach with the use of SE techniques of documentation and change control to assure the quality of the process and the final prototype.

4.5 Evaluation of Development Methodologies: The Contribution of a Technical Perspective

Figure 4.2. summaries the main strengths and weaknesses of the methodologies covered so far. It was mentioned earlier that two principal issues have emerged from these discussions: the role of prototyping; and the contribution of software development techniques in ES development. The trend towards hybrid methodologies suggests that neither RUDE or SPIV models of development are wholly appropriate and therefore aspects of both are necessary or useful within a life-cycle.

Methodologies based on the waterfall model contribute accountability, management control and resource planning techniques amongst other benefits. However, this model does nor provide a complete fit, and it should be recognised that ES also have distinct requirements; these include knowledge elicitation, knowledge representation and knowledge-base testing for instance. In addition, ES has special Human-Computer Interaction (HCI) requirements in areas such as explanation and dialogue; and new problems are generated such as the balance of initiative between the system and the user. In these areas, the role of prototyping is indispensable; not as a development approach *per se*, but as a development tool used to augment more conventional processes of analysis, design and implementation. The complexity of such hybrid approaches is in accommodating necessary iterations whilst providing a structure and retaining control.



Methodology / *Factor*		KADS	STRUCTURED ANALYSIS	GREENWELL	ABACUS	GEMINI	QUALITY ASSURANCE
MODEL		SPIV	SPIV	RUDE	RUDE	Hybrid	Hybrid
LIFECYCLE		Waterfall	Waterfall	ES Lifecycle	ES Lifecycle	Mainly Waterfall	Mainly ES Lifecycle
DEVELOPMENT STAGE	Problem Identification						
	Requirements Analysis	▩	▨			▩	
	Design & Development	▩	▩	▩	▨	▩	▨
	Implementation			▨	▨	▨	
	Testing & Verification	▨	▨			▩	▨
	Maintenance					▨	▨

▨ *Methodology Provides Guidelines Only* ▩ *Methodology Covers this Stage in Detail*

Figure 4.2: A Comparison of Current Expert System Development Methodologies

Discussion so far has focused upon the technical merit of current development methodologies. Each positively contributes to an understanding of technical requirements and their relative strengths and weaknesses may be understood in the same terms. However they all share common limitations in so far as they operate within the bounds of a technical perspective. For instance, Figure 4.2 shows a complete neglect of initial technology assessment and other pre-project issues such as problem identification and application selection. These limitations are more properly understood and therefore moderated by adopting differing perspectives on the problem. *Chapter 3* has described the conceptual value of organisational and human perspectives; these arguments will be extended further to define potential organisational and human processes and mechanisms specifically for ES development.

4.6. The Contribution of Methods Derived from an Organisational Perspective

It was suggested from Chapter 3 that processes of assessment and development which derive from an organisational perspective promote a problem orientated approach where organisational settings shape the way development is undertaken. The implication is that an understanding of the problem and the problem context should define the selection of technology, its features and development approach. Three major concepts were also defined from this perspective: a consensus based approach to solving socio-technical problems; an emphasis upon the human and organisational processes of change in shaping the outcome of technical development; and the importance of evolutionary (organisational) design and participation as a means of reducing project uncertainty. A characteristic of all three is that they necessarily function at a level which is internal to the organisation of the development project itself. It is possible also to identify a second (macro-scopic) level which relates the project, and its characteristics (such as structure, politics, culture, etc.), with that of the rest of the organisation as a functional unit or service, as a business entity, and as a user or consumer of other products and services in the marketplace. This external organisational setting thus covers issues of market positioning, business value, vendor links, technology transfer and other 'environmental' factors which shape and influence the discourse of internal organisational factors. Both are discussed with respect to the limitations of current methodologies and possible enhancements are suggested.

4.6.1. External Organisational Settings

A first consequence of adopting this setting is that analysis and investigation should begin not by evaluating ES technology but by defining organisational needs. These needs may be expressed in business terms and with respect to the strategic links with customers and suppliers of hardware and software for instance. The contribution of ES may therefore be expressed in terms of improving business value, strengthening market position through competitive advantage, or improving organisational effectiveness. This shift in perspective from technology to business and organisation has a number of implications:-

a) That ES development should be managed as a business rather than a technical matter and therefore business planning and analysis should become part of the life-cycle.

b) The way in which ES technology is introduced into the organisation, the management process, people involved and transfer mechanisms, will shape the project outcome.

c) That organisational assessments should necessarily precede technology and business assessments.

The potential in making use of methods and techniques at this setting in order to enhance current ES approaches, is restrictive if they cannot be incorporated into the development process. This requires firstly, that business and organisational value is identifiable from ES technology potential; and secondly, that a mode of technology transfer is achievable which allows for the effective introduction and diffusion of the technology from the marketplace into the organisation. Common to both these factors, again, is that they are processes which precede technical design. Their role in the conventional life-cycle would be in providing a business and organisational context for problem identification. Figure 4.2. shows however, that of the current ES methodologies discussed previously, none undertake pre-development assessment. Furthermore, the notion of problem identification, as exemplified by Harmon and King(1985) in the ES life-cycle for example, is one of identifying task characteristics which indicate that an ES solution is feasible rather than desirable for the organisation.

4.6.1.1. Defining Business Needs and ES Technology Potential

A number of authors have expressed the need to adopt a 'business-driven' approach towards the development of expert systems (Bader & Weaver:1988), and describe how ES may be used strategically (Berkin:1986). However, there is scant coverage in ES literature on how to achieve these requirements. Stow *et al* (1986) propose a business analysis approach which focuses upon the decision-making processes in the organisation. The assumption is made that unstructured decisions are likely to be more valuable to the organisation than routine and structured ones and therefore the business potential for ES will be greater in these areas. Probert (1989) refers to the concept of the 'knowledge lens' by which information relevant to the operations of an organisation is focused by ES technology to provide critical decision-making processes. The assessment process therefore is one of identifying business opportunities where it is important to distil information. Similarly, Harbridge(1989) adopts an information resource mapping methodology which analyses the information inputs to a business activity and the 'added-value' contributed through the addition of knowledge and experience.

Two particular business tools are frequently used to describe the business value of information technology from the conventional Information Systems (IS) planning domain and these are applied to expert systems in the next chapter The first is Porter & Miller's internal 'value-chain'(1985). This is an approach which assesses the potential of IS to provide a competitive advantage by creating barriers to competitive entry, switching costs for customers, offering new products and services and changing the balance of power with suppliers for instance. A second method is to define the 'critical success factors'(Rockart & Creszenzi:1984) of the organisation. Analysis of the factors which make up these critical aspects of the business may reveal ill-structured tasks where ES may be used (Buchant & McDermott:1984).

A criticism of these techniques is that the business value of IS is described in terms of competitive advantage through new and innovative uses of the technology. However, as Galliers (1986) has pointed out, business value can be expressed in terms of organisational efficiency and effectiveness as well as competitiveness. The former two operate at the organisational level whilst the latter at the market level. Furthermore, as with technical methodologies, the important issue is the context of their use rather then focusing discussions at the tool level only. In business terms, this requires that attention is drawn towards business strategies and how they may be included in the ES life-cycle. Earl(1989) provides a useful comparison of three main business strategies for Information Systems: these have been adapted to expert systems specifically and are summarised in Figure 4.3.

i) Top-Down Clarification: this strategy has also been called a business planning approach (Parker *et al.*1988) and begins by the formal definition of a business strategy based on business goals and plans. From this, the factors (functions, activities etc) critical to success in achieving these goals are identified. Finally, the information systems(technology based and manual) which support these factors are determined either through the application of technology in products and processes, or the development of information systems for co-ordination and control of activities and for management decision-making. In both cases, the potential opportunities for ES may be assessed. The benefits of this approach are that business planning drives business organisation, which should drive the ES planning that is intended to support it. As Earl notes, the process is one of clarification: first business strategies and needs have to clarified as much for senior and operational management as for those directly involved in ES. Only after this process can the potential contribution of ES can be clarified, suggesting priorities, direction and outline needs.

ii) Bottom-Up Evaluation: In this business strategy, the planning process begins first by understanding and evaluating the present situation, in terms of IS investment. The logic behind this approach is that it is necessary to understand the strength and weaknesses of current IS application before further investment in systems like ES can be made. Furthermore, examination of current systems may indicate under-utilization of systems, that they may be better exploited for competitive advantage, or be enhanced upon, using ES, to become more effective. Earl notes that this approach is evaluative and that by identifying problems and shortfalls in the current IS schema, a list of development priorities may be defined which add upon the value and may be integrated with existing systems. The potential for ES therefore is in 'adding value' to existing systems by improving performance and problem-solving capabilities.

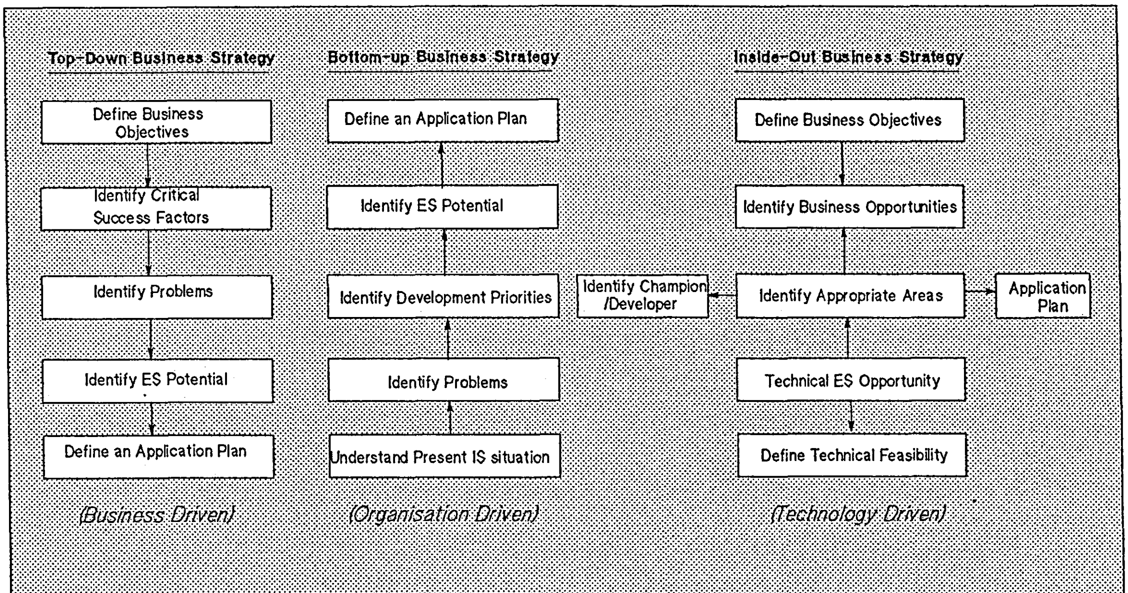


Figure 4.3: Possible ES Business Strategy (Adapted from Earl:1989 & Parker:88)

iii) Inside-Out Innovation: this strategy has also been called technology impact planning (Parker *et al.*) in that opportunities for the future use of technology are defined on the basis of their ability to change business strategy and plans. Earl describes this approach as creative in that an organisational and technological environment is designed which promotes ideas and innovations by individuals. Innovations are then judged on their potential to add-value to existing systems or create new strategic applications. This strategy therefore gains impetus from inside the organisation by the use of one-off techniques, investment in organisational processes which foster innovation and construction of an enabling technological environment.

Each of the three strategies has different requirements and impacts upon the organisation and upon the technology transfer process. Earl notes that a business strategy will be made up of aspects of all three approaches. Moreover, the predominance of each is contingent upon the emergent properties of the organisation and the emphasis laid on each will vary over time.

According to the business strategy, or mixture of strategies adopted, there are numerous business planning and impact tools, frameworks and models which may be used. Earl again provides a useful classification of generic frameworks according to awareness, opportunity and positioning of the technology in the organisation. These are discussed in greater depth in Appendix III and Opportunity and Positioning Models particularly are applied in proceeding Chapters.

a) **Awareness Frameworks:** These could be used to demonstrate how ES may be used for strategic advantage, and help senior management in an organisation assess the potential impact on their business, internally and externally. Awareness frameworks are more conceptual than prescriptive devices.

b) Opportunity Frameworks: Unlike awareness frameworks, opportunity frameworks are analytical tools which are used to systematically define an organisation's business strategy. They could also be used to clarify business strategies in order to demonstrate options for using ES.

c) Positioning Frameworks: These frameworks are orientated towards implementation rather than formulating business strategy. They may be used to help clarify the current IT situation of the organisation against which new ES developments may be evaluated. The aim therefore is to improve understanding of how ES should be managed according to the specific structure and layout of the organisation.

The business frameworks and models suggested vary in scope and application; from being informal and educational, to highly formal and prescriptive. Their use as with other tools and techniques, must be shaped by the requirements and culture of the organisation, and the strength and weaknesses of the technology itself. However, a number of generalisations may be made: Figure 3A in Appendix III shows the positioning of frameworks and respective models according to class of strategy defined earlier. It can be seen that top-down strategies tend to be highly structured and analytical and are driven by business factors rather than technology. Bottom-up strategies by contrast are organisationally driven and therefore evaluative in that planning begins by examination of the present organisational situation. Inside-out strategies are highly informal and creative and driven by the technical potential of ES, for instance, to change business performance. A combination of all three may be necessary at different stages of organisational development and technological maturity.

4.6.1.2. *Approaches towards Technology Transfer in ES development*

A second benefit of viewing development from an external development perspective is the realisation and necessity of extending the ES life-cycle beyond the specific development issues to include the processes of embedding ES in the organisation. Guida and Tasso note a failure by researchers and practitioners to consider technology transfer as being a significant determinant in the success of ES in an organisation. Smith (1984) argues that the peculiarities of ES as a new and unproved technology, together with the potential impacts on management, technical staff and end-users, make ES technology transfer different from traditional technology transfer. Furthermore, the different disciplines involved and structurally different development processes, such as rapid prototyping and knowledge acquisition, suggest that different people and different parts of the organisation will be involved in development not previously required in conventional systems development.

Hazeltine (1987) describes some of the difficulties and problems in transferring ES from technology research to developers and from developers to end users. However the accent in this and other studies is upon substantial ES applications within large corporate organisations. Research requires broadening in this area to include an understanding of market and vendor relationships and influence over the transfer process: of how organisations acquire knowledge and expertise of ES technology ; and the different transfer options and routes available. These issues are discussed in some detail in **Chapter 8.**

4.6.1.3. Commencing ES Development with Organisational Assessment

There are a number of reasons why organisational assessments should be undertaken foremost:-

i) the belief that the organisational needs and characteristics should constrain technical development,

ii) to ensure that the organisational problem is understood,

iii) to understand and predict the possible impacts of embedding ES upon functions, people and business. For example, Diaper cites examples where ES improved the internal efficiency of a task function, but disrupted those who were serviced by the function. In this case organisational mapping could help gain an understanding of the causality of problems across levels and between functions in the organisation.

The requirements of an organisational modelling process is addressed directly in the next chapter. However pointers are provided in this chapter by describing ways in which tools and models may be used in formal, structured and informal (or tacit) ways to achieve certain objectives.

4.6.1.4. Changing Development Structure: From Life-cycle to Activity Framework

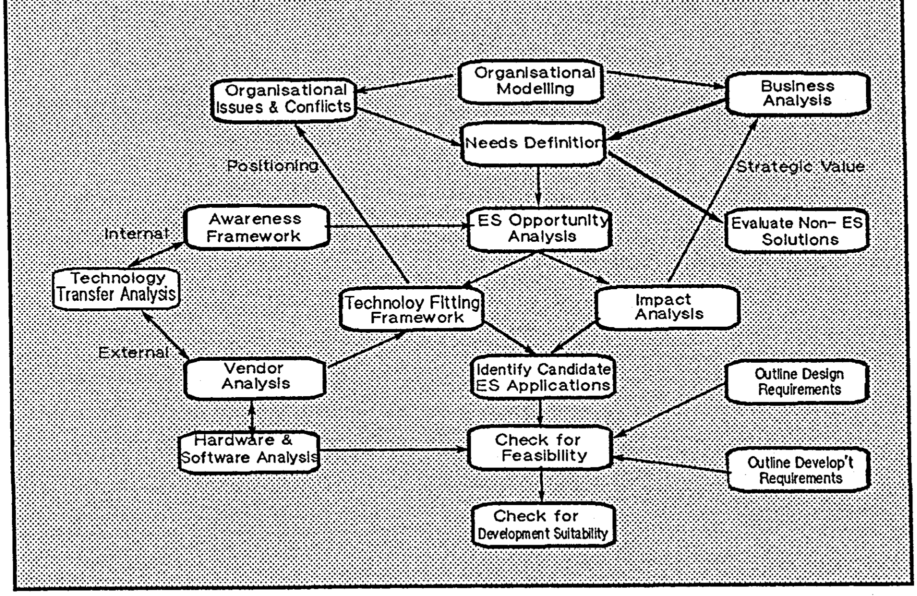
Discussions from within a technical perspective have focussed upon the structural and methodological differences between conventional software development and expert systems life-cycles. The role of the life-cycle is to provide a basic framework around which the activities prescribed by a methodology can be organised. This has led to arguments for a hybrid structure which combines the merits and attributes of waterfall and ES life-cycle based methodologies. Closer attention to external organisational factors moreover, suggests the need to incorporate business and strategic factors into the life-cycle and address technology transfer issues, both of which indicate a greater focus upon pre-project feasibility and a preference for organisationally driven analyses.

This progression in thinking requires a departure from the life-cycle concept towards a different structural approach. As a means of differentiating between the two, the term '*activity framework*' will be used. The activity framework, is intended to provide a logical description of the necessary tasks required in order to accomplish not just ES development, but assessment, evaluation, testing and so on. An example for the assessment process making use of some of the concepts described in this section, is given in **Figure 4.4.** It is invariant of task and activity definition, techniques and methodology, and implementation, because these processes require an implicit understanding of the task and human and organisational contexts in which the methodology will be used. It differs therefore from the sequential, one best way and highly prescriptive notion of the SPIV model in that the framework is intended to provide a basis by which organisations may define their own technical and non-technical needs rather than adapt organisational settings to the constraints laid down by a 'standard methodology'. Furthermore, in that the framework widens the scope of activities to include various pre-project assessments through to post-implementation evaluation, it overcomes the construction and design bias of RUDE development models. By being invariate of tools and methods, it also has two further benefits:-

i) Current methodologies, like KADS, are designed for large ES development projects and their use requires substantial organisational commitments in time and resources. They fail to address the majority of ES projects which are small-scale and shell-based. The framework is more applicable to this class of development, but also relevant to all classes.

ii) Current methodologies require specialist technical skills in order that they may be understood and applied. The framework by contrast, is activity rather than methods based and is therefore likely to be more comprehensible to a wider spectrum of people and backgrounds, all of whom may be directly involved or affected by the project.

Figure 4.4 : A Sample Activity Framework for Expert Systems Assessment



By drawing attention to the emergent properties of the organisation the framework is shaped into a *development approach* which defines the appropriate processes and tools required in order to accomplish those tasks outlined in the activity framework. This understanding can only be attained by regarding development from different perspectives, beyond the technical and business. In particular, the application of the activity framework requires an understanding and integration of internal organisational and human concepts. Their contribution is in shifting the emphasis from simply defining the range of tasks necessary for development (i.e. 'ends') to understanding also how these tasks may be successfully undertaken in a specific organisation (i.e. 'means'). The following two sections look firstly at how current technical ES methodologies fail to incorporate human and organisational perspectives; and secondly assess if it is possible to incorporate methods derived from these perspectives into a development approach.

4.6.2. Internal Organisational Processes: Adding the 'Means' to Development

Internal organisational concepts view the organisation as a socio-technical system and therefore contribute in two respects: in terms of understanding the problem from a number of viewpoints; and of embracing a philosophy of design based on human participation.

4.6.2.1. Problem Conceptualisation

A distinction was made in Chapter 3., between hard problems which are well structured and predictable and soft problems which are unstructured and conditional on personal perceptions, and therefore differ according to the organisational actors involved. Checkland has noted that by viewing problems in hard terms alone results in a subsequent 'technique orientated' development approach (1981). Despite the unstructured characteristics of knowledge and the organisational difficulties of implementing ES (Mumford:1989), current ES methodologies adopt a hard conception of the problem on the basis, as Montgomery(1989) identifies, that they are reliant upon specific techniques. For instance, GEMINI described earlier is interworked with the development methodology SSADM and Kelly's structured analysis based methodology is dependent upon Yourdon. This reliance upon techniques moreover has extended to the orientation of techniques around a specific software application as in the case of IBM's development methodology for their ES product 'ESE' (ITSC:89).

A limitation of hard techniques is that they seldom provide constraints within which they operate and therefore they assume validity in all problem situations (Veryard:1987). Furthermore, the soft processes which constrain the use of techniques are not included in the techniques themselves. A soft approach recognises different views of the problem and that there is no single best approach towards development. There are two related benefits therefore in the addition of soft techniques to current ES development methodologies:-

i) **Enhancing Problem identification:** Figure 4.2 has shown a neglect of methodologies to consider problem identification as part of the life-cycle. Where it is included moreover, it focuses upon the technical feasibility of applying ES based upon the technical characteristics of the problem. Soft techniques may shift the focus towards understanding the problem in organisational and human terms as a precursor to technical analysis. In this way, solutions may be defined in non-technical as well as technical terms.

ii) **Non-prescriptive:** a characteristic of waterfall based ES methodologies particularly, is that they define a single procedural approach based upon the technical efficacy of techniques to control and manage development optimally. This prescriptive approach is based on the assumption that there are single optimal development goals. A soft approach highlights that there are multiple conflicting and complementary goals determined by social, political and business factors as well as technical. The process of defining requirements shifts from being prescriptive to consensus based in that different viewpoints should be accommodated.

The extent to which soft methods may achieve these potential benefits is contingent upon the effectiveness of the method chosen, if indeed it is possible to define a 'soft' methodology whilst still retaining the attributes of soft concepts. Checkland(1981) has developed a Soft Systems Methodology (SSM) for this purpose. The basis of SSM is described in Appendix III. It is intended more as a 'learning and enquiry system' or 'problem-solving framework' rather than a specific technique, which uses system based models to understand what Checkland calls 'real-world' problems (Checkland:1989). It is not intended to outline the methodology here, since it's influence in future chapters is in the nature of the investigation rather than the specific mechanisms used.

4.6.2.2. 'Process-Orientated' Design and Development:

Mumford(1987) criticises ES development approaches for concentrating upon technical design aspects of the system and failing to consider the factors such as desirability and acceptability. She argues moreover that the social processes of development are equally as important as the technical ones. Markus also speaks of the importance of social factors in systems design and lays particular emphasis upon political relationships in shaping outcomes. From this organisational perspective, a number of enhancements to current ES methodologies are required:-

a) Consider implementation: a focus by methodologies upon the tasks to be completed overlooks the necessity to define how they may be implemented. Figure 4.2. shows that methodologies based on the SPIV model omit implementation issues outright, and although methodologies like KADS contribute by providing techniques for managing and controlling aspects of implementation, this is related to metrication and technical issues, rather than the social organisation of people and work. Those based on the ES life-cycle cover implementation in the sense that prototyping is used. However, Mayhew and Dearnley (1987) note that this form of prototyping is used to specify technical requirements, such as knowledge representation language. These authors identify the need for an additional form of prototyping which they call 'organisational prototyping' which considers the interaction between the user and the system to assess the complete systems suitability from an organisational context. This role for prototyping is more congruous with Markus's notion of evolutionary design discussed in the last chapter (1984).

b) Addresses the context of development: a standardisation of tasks and procedures brings benefits in highly structured and technical aspects of development. However current ES methodologies may be considered too structured and standardised in that they are insensitive to the context of development. An organisational perspective highlights the fact that emergent organisational and individual settings will constrain and be constrained by the technical setting. Thus the methodology should recognise informal as well as formally defined needs.

c) Methodology as an Innovation: the size scope and complexity of methodologies such as KADS, GEMINI and SSADM are such that they may impose new disciplines, procedures, work organisation and may even change management structure and generate possible resistance. Alternatively, the organisation may become too dependent upon the methodology with a loss of confidence in personal judgement. An organisational perspective recognises , as Veryard (1987) observes, that a methodology carries its own culture and structure which may clash with that of the organisations unless it is flexible enough to be adapted.

A means of adapting the methodology as well as the technology according to socio-technical settings is *participation* in which users drive the social dimension of development. A particular approach based on participation is ETHICS (Mumford:1979;1983). As with Checkland's Soft Systems methodology, the contribution of ETHICS is in the nature of the socio-technical approach rather than the specific mechanisms of the methodology and therefore it will be described in these terms. Details of the approach are given in Appendix III. The main purpose of ETHICS is the identification of compatible pairs of alternative technical and social designs after establishing technical and social objectives. This provides a set of 'feasible' socio-technical solutions, which are evaluated against the technical and social objectives for the system and ranked. Markus points out that ETHICS and similar approaches based on end-user participation are limiting in their focus on job satisfaction whilst overlooking management structure and other political and organisational settings (p117). Despite this accepted reservation, such approaches are potentially useful for ES development for the following reasons:-

a) It accepts the likely technical orientation of the development team and therefore incorporates explicit social goals in the assessment process. Furthermore, the process of defining social objectives is user-driven.

b) That social objectives have been defined means that there is a social dimension to the evaluation process. Current methodological approaches for ES which include evaluation in the life-cycle define it in terms of whether designs have been achieved to cost and specification rather than whether it meets organisational needs.

c) It assesses the resources available for use by the technical and social aspects; the latter may include training needs, recruitment ,communication requirements and office re-organisation for instance.

d) It recognises the value of assessing the current situation in terms of working relationships and other organisational factors from which social needs may be defined. It also discerns that the strengths in the existing work organisation should be incorporated into the design of the ES.

e) It recognises that to each problem there may be numerous technical and non-technical alternatives each judged on the merits of separate social and technical criteria. For each technical alternative, the analyst is required to explicitly state technical needs, identify technical constraints, resources available for technical development and specify technical objectives for the system. Social alternatives are defined in a similar rigourous way through the use of questionnaires and user workshops.

4.7. The Contribution of Approaches Derived from an Individual Perspective

The conceptual basis of an individual perspective was described in Chapter 3. The development problem is conceived in terms of maximising human potential with the role of technology being considered as a 'facilitator' in achieving this objective. The methodological expression of an individual perspective may adopt one of two approaches according to the classification defined in Chapter 3. An 'engineering paradigm' in which human factors are limited in scope to defining and constraining technical design features; and a 'human-centred paradigm' in which wider social and cultural changes lead to a re-definition of the role of technology and the development process according to the predominance ascribed to human needs. Both represent extremes in a spectrum of possible forms of human contribution. The extent to which either may be used is contingent upon the social, political and cultural characteristics of the organisation and business sector.

The wide variety of users and other people involved in assessment and development, and the evident complexity of addressing their needs, makes the integration of human concepts with current methodological approaches difficult to achieve as an objective. Although there are a number of different ways in which users can be brought into the assessment and development process, as Damodaran & Eason (1981) have identified, such as user representation, user evaluation and user-driven prototyping and a less formal interpretation of participative design, Candy and Lunn imply that these processes can only be effective when the user is given the opportunity to influence design implicitly (1988). Furthermore, they argue that the only way to design an 'acceptable' system is to centre the process on the personal needs, tasks, scope and operational conditions of the individual. In practice, Eason *et al* (1987) note that a compromise is required between what is *desirable* to the individual in terms of personal fulfilment, what is *realistic* and *feasible* within technical, economic, commercial and human constraints, and what is *acceptable* to all affected individuals and groups involved. This successfully combines assessments based upon technical and organisational and human concepts, but the balance struck between the emphasis and contribution of each must ultimately lie with the person or organisational function placed in charge of the project or proposal. This leads to the issue of how ought different assessments based upon these different concepts be combined most effectively ?

4.8. The Need to Combine Assessment Concepts

Table 4.2., summarises the possible contribution of technical, organisational and human concepts and their respective methods across the activity framework. Each concept contributes in different ways towards understanding the context of development, defining needs and providing tools and techniques by which to perform assessment and development. However, it is apparent from *Table 4.2.* that there is no single perspective which covers all aspects of development and also that each perspective is more appropriate at particular phases in the activity framework than others. If no single perspective covers all aspects of development, then by implication, there can be no single methodology which covers all phases of the activity framework. Furthermore, Benyon and Skidmore (1987) argue that attempts to find a single 'one-best' approach to deal with the wide variety of applications and contexts leads to 'bureaucratic and elaborate' methodologies.

	Perspective *Activity*	**Technical** *'SPIV' Model*	**Technical** *'RUDE' Model*	**Organisational** *External*	**Organisational** *Internal*	**Individual** *Human Engineering*	**Individual** *Human-centred*
OUTLINE ACTIVITY FRAMEWORK	Needs Definition			*Organisational Modelling & Technology Transfer*	*Problem Definition*	*Human Requirements Definition*	
	ES Opportunity Analysis	*Hardware & Software Analysis Technical Problems*		*Technology Fitting Framework*		*Human problems characteristics*	
	Selection	*Technical Feasibility Guidelines*		*Business Assessment*	*Selection By Consensus*		*'Human Return' Criteria*
	Requirements Analysis	*Specification & Planning, Structural Analysis*	*Throw-away Prototyping*	*Activity Framework*	*Soft Systems Analysis*	*Human Factors Analysis*	
	Design & Development	*Formal Design Methods*	*Evolutionary Prototyping*	*Technology Transfer (Internal)*	*Organisational Prototyping*	*Interface Design*	
	Implementation	*Documentation & Control, Quality Assurance*	*Incremental Prototyping*		*Consensus Participation*	*Intrinsic Participation*	•
	Testing & Verification	*Validation & Verification*		*Business Evaluation*	*Organisational Evaluation*		*Human-centred Evaluation*
	Maintenance	*Maintenance Documentation*					*Iterative Human-centred Re-Design*

Table 4.2: The Contribution of Different Paradigms In Expert Systems Development

Methodologies based upon a single dominant perspective may fail in a number of ways :-

a) fail to represent the interests of all people involved,

b) fail to recognise some activities as being important to development and therefore fail to cover them in detail,

c) fail by being inappropriate to the tasks, social and political organisation or individuals involved.

The value of combining perspectives in assessment is that the inherent weaknesses of each perspective are compensated by the strengths of others. This value may also be expressed in terms of reducing the total uncertainties of development made up of technical, organisational and individual components. Technical uncertainty arises from conditional factors in design and risks of technology failure. These are reduced by formal design methods, change control procedures and quality control for example. To the analyst or developer, technical risks may be conceived as being the only important risks and therefore efforts are made to ensure predictability through the use of formal methods and standardisation of procedures. The analyst though may be less sensitive or unaware of a second form of uncertainty, organisational uncertainty, which derives from a neglect of organisational and social processes in defining needs and during implementation for example. Mechanisms which have been developed to reduce this uncertainty, such as participation and evolutionary design may seem necessary to the manager but superfluous to the analyst (Markus:1984). A third uncertainty, subjective or individual uncertainty, is based upon the likelihood of acceptance or rejection of the technology by the individual as a user, manager of expert for example. This uncertainty may be reduced by incorporating approaches which promote the implicit involvement of the individual during all phases of development in order to achieve human as well as technical and organisational goals.

Each form of uncertainty may only be apparent to individuals who hold the same viewpoint, and therefore the total uncertainty of the project may be minimised only when all interest groups affected by the proposal are actively involved in the development process. A potential value therefore in combining viewpoints is that despite the inherently ill-defined and ill-structured characteristics of ES development (Wilson *et al*), the process may become more predictable such that estimates of effort timescale and manpower requirements, for instance, may be estimated more accurately.

4.8.1. How Can Different Assessment & Development Concepts be Combined ?

Chapter 3 indicated that Multiple Perspective Concepts (MPC) were useful in communicating the need to combine different perspectives and settings in the assessment and development process. This chapter then. outlined a number of approaches and techniques which are derived from each of these perspectives, all of which are potentially useful. The remainder of this chapter seeks to define how, in practical terms, these tools might be combined in a way which is sensitive to the development context and needs. First guide-lines are available from the literature. Benyon and Skidmore(1987) for instance, advocate a flexible framework or 'tool-kit' within which a variety of appropriate tools and methods may be applied. Similarly, Edmonds(1987) replaces the notion of a methodology with a process of 'synthetic production' based on the belief that the diversity of needs generated by different viewpoints cannot be mastered by a single developer or designer. Rather that the developer's role is one of 'co-ordinator' of different groups and interests so that the development process is integrated into a wider social and organisational context.

The contribution of both approaches is that they opt for an eclectic framework which rather than follow a single methodology in effect produce a unique method for every project. However, they are of restricted use because although they outline structural and methodological options available to the developer, they fail to provide the *means* by which the most appropriate may be identified, according to task and organisational characteristics. For instance, Benyon and Skidmore identify a hierarchy of five modelling techniques available to the systems analyst operating from the organisational level down to the data and information level but provide no indication when each should be used, other than stating that the analyst should be able to judge intuitively where and when each is suitable. Since the analyst is likely to have little other than a technical understanding of development requirements, this approach is of little value if mixed assessment is the aim. A further problem is that in delineating methodologies, tool use may be motivated by local difficulties within the construction of the knowledge-base and fail to support other stages of development. As Wilson et al note, "...it is difficult to select appropriate tools for different phases in the development process because of compatibility problems between the output provided by one tool and the input required by the next " (p194). Thus, although ES tools may be considered commercial and 'pragmatic' because they address a specific aspect of development successfully, such as knowledge elicitation, their performance overall may be suboptimal because of their inability to integrate with other formal and informal mechanisms.

A more satisfactory approach , consistent with the themes of this chapter, is to retain the eclectic approach of previous authors, but rather than focus at the tools level, to provide a taxonomy of approaches according to their underlying concepts (Wood-Harper et al:1988). Moreover, Avison et al (1988) add that in order for eclectic approaches to succeed, they require an 'explorative framework' in which the analyst is aware of the underlying philosophies and assumptions of each of the tools and methods used. Although this has formed the basis for categorising current and potential ES development approaches in this chapter, a shortcoming of both the approaches of Wood-Harper and Avison et al. , as Benyon and Skidmore have argued, is that although these authors take account of the underlying concepts of the methodology, they do not define the *development context* which should influence the selection process in the first instance.

Consolidating on the above arguments, it has been identified that no single methodology is appropriate, nor is development from a single perspective satisfactory. The diversity in levels of assessment and underlying design philosophy requires different representations, formal and informal for example, of the same problem in order to help determine basic features and identify where priorities for development lie. An eclectic approach was defined in which tools were selected according to the development situation: however the selection process is complex because it focuses at the tool level. An alternative approach was described which selected approaches according their elementary concepts. This requires a conceptual framework which allows approaches to be combined to form a 'unique methodology'. The full benefits of this approach however are only attainable when approaches are combined not solely according to underlying philosophies in relation to each other, but in relation to the development context.

4.9. A Framework for Assessment and Development

The intention of a 'development framework' then is to provide a basis by which the benefits of different assessments, tools, methods and processes, each motivated by different perspectives, may be combined structurally and conceptually in a way which is congruous with the 'emergent' properties of the organisation. This section therefore brings together the theory described in this and the last chapter to provide the framework shown in Figure 4.5. Each component of the framework, its functioning and relationships to other components will now be described:-

4.9.1 'Conceptual Framework'

Chapter 3, described a conceptual framework, MPC which highlighted the importance of mixed levels and mixed conceptual assessments, and demonstrated that by looking at the problem from different settings, it was possible to define an appropriate context for development. The ability of MPC to both communicate the necessity of combining perspectives in analysis, and define a development focus suggests that it may fulfil the requirements of both a development context and act as an 'exploratory framework'. This allows for an 'eclectic' development approach for ES development which is founded on two principles: firstly, an understanding of the underlying concepts behind the choice of tool or approach adopted; and secondly an appreciation, through the use of MPC, of the development context. The development context not only inhibits the choice of development approach during selection, but it also shapes its use during delivery. The choice of mechanisms (mechanism denoting anything from a 'hard' tool to an informal process) is made on the basis of the balance of perspectives identified by MPC for each task defined by the activity framework. What is less clear about this proposal and necessary in order to render it a coherent approach, is to define how the different characteristics of methods from each approach (formal/informal, explicit/implicit for example) may be reconciled within a common framework and what status should each have in relation to the other ? To understand this requires that each of the components of the framework, and their functioning and interaction with other components, is considered in greater detail.

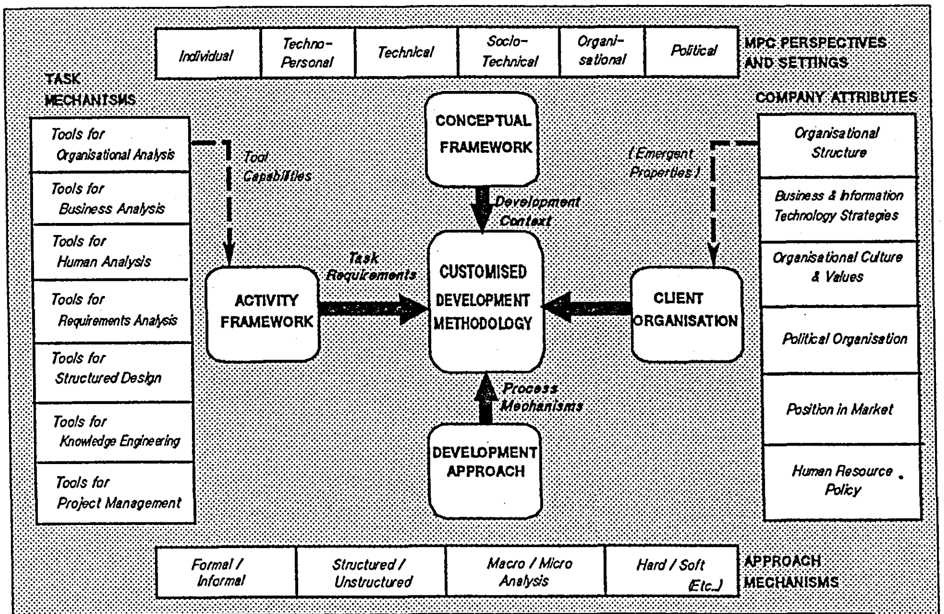


Figure 4.5: An Organisationally Specific Assessment and Development Framework

4.9.2. *Activity Framework*

In was indicated earlier that a main feature of the activity framework is that it is invariant of context and application, and instead defines the required steps for all development approaches. It is analogous to concept of minimal critical specialisation, as defined by Markus (1984;p106) in which the minimum development requirements are specified necessary and essential. This prevents overspecialisation prior to application of the framework and makes it flexible enough to be adapted to the emergent properties of the organisation whilst providing a scheme of assessment and analysis tasks. It is assumed that for each of the tasks defined by the activity framework there are numerous business, organisational and technical tools and techniques which may be utilised, some of which have been described in this chapter. The application of the framework addresses the issue of not simply *what* is necessary for development, but *how* development may be undertaken. Internal organisational and human concepts contribute methods and processes which achieve this aim.

4.9.3 *'Development Approach'*

The balance of perspectives and the subsequent allocation of formal and informal methods and processes is contingent upon the balance of organisational settings defined using MPC. Thus the term 'eclectic' should be used not just in the sense of using many different tools, but that they may be used in different ways according to the settings of the applied context. The necessity of relating the conceptual basis of mechanism with that of the development context is that the former is shaped by the characteristic settings of the latter. For example, Avison et al(88) note that in some organisational settings, mechanisms like participation may require an explicit statement and formal structuring in a methodology like ETHICS; in other development contexts however, it may be acceptable to use participation implicitly and informally in a wider development approach. A second example, which relates to experiences in Chapter 5, shows the opposite effect in which soft methods in a hierarchical and political environment prove difficult to use and it was therefore necessary to provide a hard 'front-end' to these soft processes for purposes of legitimacy and acceptability.

The implication from both examples is that mechanisms however precise may be used in a formal and informal way. In order to ensure a predictability in the mechanism's use it is necessary to match their underlying design philosophy with that of the development context of the task in hand. It helps therefore to understand the characteristic attributes of different approaches so that they may be combined effectively. Hurst(1984) provides a useful distinction between hard processes or 'boxes' and soft processes or 'bubbles'. His theories relate to managerial decision-making, but equally they may be used to define appropriate mechanisms for a particular task and development context. In most situations, Hurst notes that bubbles are required to complement box processes: each address different concepts and focus upon different values as with the examples shown in **Table 4.3** below. Current ES development methodologies and tools represent hard box solutions: bubble processes are absent from these approaches because they are not considered as being valuable.

Hurst suggests an approach which defines five basic sets of hard and soft relationships. These are tasks and roles, structure and groups (e.g. organic or mechanistic), communications (e.g. formal or informal), people (e.g. rational or social) and strategy (e.g. objectives or values). By recognising the dichotomy between the two sets, Hurst suggests that it is possible to map an appropriate balance. For the purposes of providing a framework for ES, a development approach based on this mapping may be defined, indicating the interdependency of 'boxes and bubbles' between different basic forms of assessment necessary in ES development. In discriminating between development mechanisms in this way, the approach reveals a number of meanings:-

a) From each assessment there are formal tasks and deliverable as well as informal processes which interact informally with other mechanisms.

b) Each formal assessment may is used in a formal and informal way; in the second case, the assessment is shaped by other assessments and the development context.

c) Hard activities tend to provide one-off deliverables, formally specified in advance. By contrast, soft process are on-going and iterative because mechanisms are constantly being re-adapted according to changes in requirements and the development context.

d) Combined hard and soft processes tend to be informal and soft because no structured framework or formal method is attainable or organisationally appropriate,

e) Informal processes may evolve from formal activities or vice versa, or can be made formal or produce formal results,

The development approach is thus sensitive to the level of analysis in that informal processes are likely to be used at the strategic and executive levels of the organisational but may be communicated downwards through more formal processes. Mixed level analysis is necessary because changes required at one level may only be attainable by providing mechanisms at another organisational level. For example, in order to achieve personal human-centred goals at the individual level, it is necessary to change attitudes and values at the organisational and business level by providing informal awareness schemes or formally implementing a human resource strategy for instance.

The potential value of this approach therefore, is in understanding the range of hard and soft mechanisms available through mixing concepts in assessment and through an understanding of the context in which these mechanisms are to be applied. It therefore defines appropriate *approach mechanisms* which may be combined with earlier theories to provide an emergent framework for ES development.

Activity Framework Tasks	Technical Approaches **Hard Boxes**	Organisational Approaches	Individual Approaches **Soft Bubbles**
Organisational Modelling	Closed		Open
	Structure		Framework
	Formal		Informal
Needs Definition	Objectives		Values
	Necessary		Sufficient
	Rational		Intuitive
Justification	Objective		Subjective
	Direct Benefits		Indirect Benefit
ES Design	Precise		Evolving
	Scientific		Experiental
	Defined		Undefined
Implementation	Control		Influence
	Task		Process
	Sequential		Lateral

Table 4.3: *Hard Tasks (Boxes) & Soft Processes (Bubbles): Characteristic Attributes of Methods*

4.9.4. Client Organisation.

Central to the effectiveness of the framework is the necessity of learning how the organisation operates and of gaining an appreciation of each of its settings in order to define its emergent properties. This is especially pertinent in the case of the 'outside consultant' who is brought into an organisation to fashion policy/strategy, perform technology assessments and development work. The need to understand the properties of the organisation begins at a macroscopic level through an understanding of company operations, structure, physical layout and history. In the author's case, information about the sponsoring or **client company** is provided in Appendix II for this reason.

There is also a second level of understanding though, often overlooked by the consultant, which explains the organisation's culture, values, politics problems and requirements. Such 'insider' information determines how strategy is *actually* implemented and why policies are made and the relative success of both. It also conveys much about why a company has problems and is equally relevant in understanding how such problems might be resolved. Despite this, very little research has been carried out on effective means of gaining such an understanding, other than to spend considerable amounts of time within the organisation. The question of how this might be achieved is discussed in the next Chapter.

4.9.5. Using the Framework

The uniqueness of the framework is the interdependency between understanding the *tasks requirements* necessary for ES assessment and development, recognising the development context in which these are to be applied and defining appropriate *process mechanisms* which allow them to be applied in an suitable and predictable manner. Thus, the choice of process mechanisms and assessment and development tools is contingent upon the development context. The emergent properties of the organisation define what is acceptable and unacceptable and help to shape the activity framework towards a unique approach which is customised towards the specific ' needs of the organisation.

In the ideal use of the framework, it can be assumed that there is complete knowledge of available software and development tools and the organisational capability to apply them. It can also be assumed that there is an organisational member who is sympathetic and able to apply the framework in practice. Using the model requires also that this person (or function) has access to company information at many levels and has the organisational support to apply it in what may appear to be unrelated areas. Clearly, unless the initiative is taken by senior management or the framework is used in a scaled-down or informal way, then the reality of its future use in uncertain.

4.10. Conclusions to Chapter 4.

A first observation which can be drawn from this chapter, is that the bulk of current research on expert systems methods and approaches is very much focused at a 'tool' level, with little attention paid to the processes of development. Furthermore, they presume that an application is justified as long as it is technically feasible, with a proclivity towards particular issues of ES design and development, whilst overlooking initial selection, assessment requirements, business and organisational value and other 'pre-development' issues.

This chapter has shown that processes and concepts derived from organisational and personal perspectives have the potential to contribute significantly towards assessment and development and compensate for the limitations of current methods which are based upon concepts derived from a technical perspective. As with any single dimensional analysis, whether technical or not, there is an inherent inflexibility in the approaches taken and therefore a subsequent uncertainty that the change process will go as planned. This uncertainty can only be reduced by combining different types of assessment in some way. There is a danger in doing this however that the mix of assessments itself becomes large, bureaucratic and unmanageable: worst still, that it begins to *prescribe* specific approaches. This is possible in well bounded, technical areas, but not so where people, groups and organisations interact.

The notion of an 'eclectic' model was valuable because it suggested a independency of the evaluation the tasks to be undertaken in assessment and development from the process of identifying tool alternatives. Thus a 'basket' of tools and disciplines were available which could be drawn upon according to their ability to perform the task, but also their compatibility with other tools and processes. Unfortunately, 'compatibility' is measured in these cases exclusively in terms of the congruence between tools and disciplines and not with respect to the development context into which they could be applied. Chapter 3 showed that MPC (multiple-perspective concepts) was an effective conceptual framework which allowed the context and settings of a particular situation to be considered. Thus this forms the backbone of a 'development framework'.

The principle behind this framework is that the organisational settings and development contexts should drive the selection of tools and approach rather than be constrained by them. It was noted, for example, that the large lifecycle methodologies were considered innovations in themselves and prescribed a method of organisation which was often discordant with the internal workings of the company. This is compounded, as O'Neill and Morris (1989) note, by the fact that these methodologies are developed in a 'research' environment and therefore are lacking an organisational perspective in their design and use. This might also explain the findings in Chapter 8 which show that very few companies actually adopt these methodologies and prefer to develop their own lifecycle. As well as determining the selection of methods and tools, the framework is also used to determine how these tools might be used (formally, informally, hard, soft etc). To achieve this requires a functional representation of the necessary analyses to be carried out (called an 'activity framework'); a knowledge of available tools and an appreciation of the settings in which they are to be used. Thus, each of the assessments are dependent upon others in order to be complete.

It is evident from the framework shown in *Figure 4.5.* that there are essentially four parallel and interdependent assessments: an evaluation of task mechanisms and tool capabilities; an evaluation of development context using MPC; an analysis of the client company's organisational attributes (culture, structure, operations, values and policy) in order to derive 'emergent' characteristics about the company; and on the basis of these three assessments to undertake a fourth which crafts approach or 'process mechanisms' which decide how assessment and development activities should be undertaken (formally, informally, hard, soft etc). In any organisational group, individuals may be adept at each of the assessments and indeed undertake such approaches intuitively; whilst others may require devices such as MPC to understand the need to consider divergent viewpoints and settings. This again requires careful management in allocating roles and identifying 'responsive' group members.

In practice, the framework is likely to operate within a number of constraints, time and resource, cost or political factors for example, which make some settings, task and process mechanisms more relevant, but also prevent the possible inclusion of others. It may be difficult to operate the framework within these constraints whilst retaining the integrity of the framework concept. It is also important to recognise the limitations of the assessment and development framework. Its use at a company level requires careful management and planning and requires a large body of information to apply it. It may also take some time to evolve a methodological approach from the framework. Although the framework itself does not define roles or a scale of operations, it is likely that a single person is unlikely to have a deep knowledge of task mechanisms (i.e. hardware, software, development tools and techniques) together with a integral knowledge of the organisation. For this reason, the framework brings together technical and organisational (and personal) roles and thus functions at a group level. In terms of managing the framework, it is likely that, for reasons of political and organisational support, a strategic function or senior person in the company should act as a facilitator.

In terms of using the framework in the client organisation, three principal roles are intended:-

> i) to communicate and validate why a particular approach towards assessment and development, which proceeding chapters describe, was adopted for expert systems.

ii) to allow a programme of assessment and development to 'emerge' and be shaped by the properties and contexts of the organisation rather than apply limited an highly prescriptive expert system methods, tools and techniques.

iii) to rationalise the problems, conflicts, failures and successes experienced during development and evaluation.

A wider role for the framework is envisaged in developing a programme of technology assessment and development rather than using it to derive a customised methodological approach. The difference being that in the former greater attention is paid towards the total lifecycle of innovation and technology transfer between organisations (vendor to user-organisation for example) as well as within a specific company. This makes the 'task mechanisms' more organisational and strategic rather than focusing upon specific tools and methods of development as Figure 4.5. shows. However the same attention to the context and processes of change are still necessary. This adaptation may also be extended to defining strategy. From Figure 4.5., this could be achieved by defining the desired 'end-state' of business or information technology strategy for example and working *backwards* to formulate a development context and process mechanisms which will enable a particular set of technical and organisational activities to be carried out which accomplish the goals of the strategy.

Chapter 5

Organisational Modelling and the Role of IDEFo

5.1 Introduction

Central to the effectiveness of the development framework outlined in the last chapter was the necessity of learning and understanding about an organisation in order to determine what were called its 'emergent properties'. A distinction was made between external(formal) information such as company details, business strategy and technology layout, and internal(informal) information such as culture, policy, politics and values. Both sets of information were required to understand the company's problems and requirements. This chapter looks in detail at how such information might be acquired through modelling the organisation in some way, as a necessary first stage of technology assessment and development. An outline of Chapter 5 is shown in Figure 5.1.

Discussions on modelling the organisation raise two fundamental questions which are related to the expectations of the modelling process: first, from which perspective should modelling take place; and second, at what level in the organisation should modelling be undertaken? As with the design and use of development methodologies described in Chapter 4, there is a difference between what a model is designed to do and how it is actually used in practice. Variances may arise between the two through differences in the way in which the model is implemented; by the attitudes and motives of the people that apply it; and the organisational setting in which the modelling technique is used.

The design of organisational models may adopt one of five possible modelling viewpoints. Although there are countless modelling techniques available, each is subsumed by this division. Each viewpoint reflects different organisational needs and interests and defines a modelling approach in order to meet these needs, whether they be business, organisational or technically motivated. As at a conceptual level in Chapter 3 and a methodological level in chapter 4, in reviewing current modelling approaches, a common criticism is that each tends to adopt a single viewpoint with the result that a model gives a one dimensional representation of the organisation when a multidimensional investigation is required. Thus, a model may succeed in identifying problems, issues and conflicts which arise from a particular setting or perspective, but fail to recognise the importance of understanding and modelling the organisation, if this is possible, from other perspectives.

From the general short-comings of modelling approaches, this chapter focuses upon the specific requirements of modelling for expert systems assessment and development. In considering modelling needs , it is concluded that conceptually, each viewpoint is necessary at different stages of the development lifecycle. For instance during problem selection, organisational and business viewpoints are the most important, whilst during knowledge elicitation information processing and human viewpoints are valuable. The potential contribution of each viewpoint is described in this way, from which arguments progress to consider ways in which these viewpoints may be combined structurally within the same modelling technique.

The selection of IDEFo, an activity based modelling methodology, was not based on any outstanding attributes for any one of the viewpoints, or because it was specifically designed for ES development, but rather that its functional representation allowed it to be used formally and informally to accommodate each of the five viewpoints, satisfactorily, when required. Furthermore, it provided the opportunity to investigate the organisation in some detail and thereby gain an insight into business, organisational and technical priorities and problems prior to technology assessment. Experiences in using the model and undertaking the IDEFo project in the company are discussed from two fronts. The first is as a formal exercise, with the objective of mapping business activities and identifying organisational problems as a precursor to general business and technical developments. Previous attempts at modelling the company are described and help to provide a context on the expectations of IDEFo and the motivation for its use. A second role for IDEFo, and one which was less well recognised by the company, was as a starting point and vehicle for a programme of assessment and evaluation of ES technology.

From this juncture, the chapter looks in closer detail at the application of IDEFo; since it is used in an 'applied' form, the theory of IDEFo is discussed at a practical level only. The first few diagrams of the model are discussed to provide an understanding of how IDEFo diagrams may be read and interpreted. Examples IDEFo use are given in Appendix IV at two levels in the organisation: the first looks at the business level and considers the Computer Department as a servicing function to the rest of the company; whilst the second looks in more detail at the physical layout, in terms of equipment and operations, of this function. The management and implementation of the IDEFo project is also discussed in some detail, since it was during the process of undertaking the study that other opportunities for secondary assessment became possible. These are outlined in this chapter and discussed in depth in the next two chapters.

The value of the IDEFo project to the company is described in explicit terms, such as providing a mapping of company activities; but also in intangible terms, as in the changes in organisational culture it cultivated, which are likely to be of greater significance in the long term. The value of the study is thus expressed in terms of the modelling *process* itself rather than simply the deliverable, an annotated model of organisation, which it is shown has proved to be of little use. An evaluation of IDEFo must therefore take account of its formal and informal uses. However it is difficult to completely separate the formal from the informal since in many cases, the latter was not possible without the former for political and organisational reasons. Possible enhancements to IDEFo in its different roles takes cognisance of this factor.

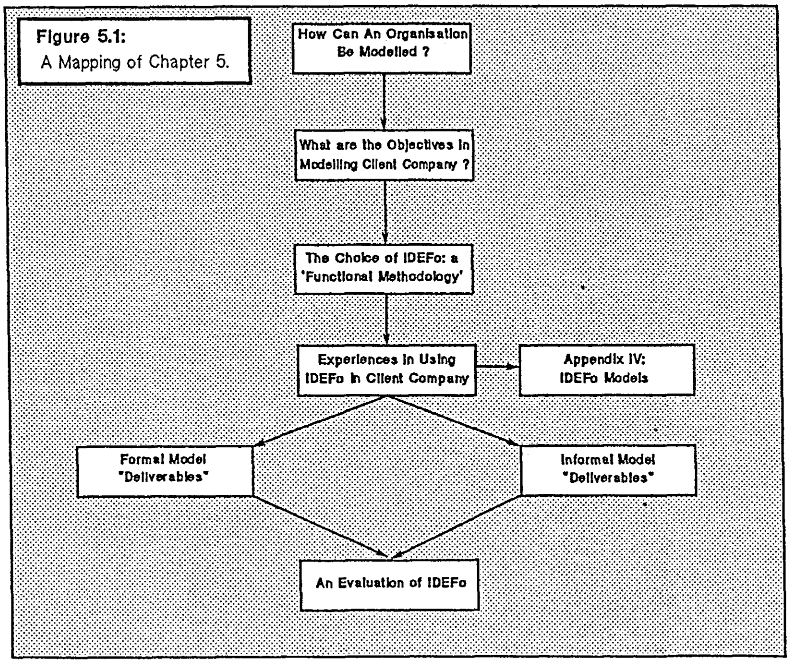


Figure 5.1:
A Mapping of Chapter 5.

5.2. Defining Modelling Needs for ES Development

Figure 5.2 identifies five basic viewpoints of the organisation which shape the modelling technique adopted. Furthermore, to each concept of the organisation, the actual modelling process converges to one of three underlying perspectives, technical, organisational and human. *Figure 5.2* also shows that perspectives and viewpoints may be linked and examples are given from the theories discussed in previous sections. The diversity of modelling approaches highlights the difficulties in defining an approach towards organisational assessment. The contention lies not in the need for an organisational assessment first, but in identifying what it is used for and how. Each modelling viewpoint is likely to contribute towards ES development in some way: business human and socio-technical viewpoints are of greatest value at the pre-project feasibility stage, for example; whilst more structured viewpoints, such as information modelling are useful during systems analysis and design. The necessity of modelling from each viewpoint, as discussions next will show, is based upon its ability to perform specific tasks as well be applied to other levels in the organisation: -

5.2.1. Modelling the Organisation as a Business entity

At a strategic level, the reason for modelling the organisation is to define the way in which business assessments are carried out. Emberton and Mann (1988) for instance, note that prior to any Information Systems (IS) planning exercise, a mandatory first task is to undertake an investigation of the organisation in order to provide a context and understanding of company problems. Characteristics of modelling at this strategic level are, as Avison and Fitzgerald (1988) note, irregular, ad-hoc and variable, and based on knowledge external to the organisation. In terms of modelling therefore, these requirements require a conceptually different approach. Avison & Fitzgerald add that, "There is no point in analysing a strategic activity and constructing detailed data flow diagrams and logic representations if it is a rare and somewhat *ad-hoc* activity unlikely ever to be repeated in exactly the same way" (pg464).

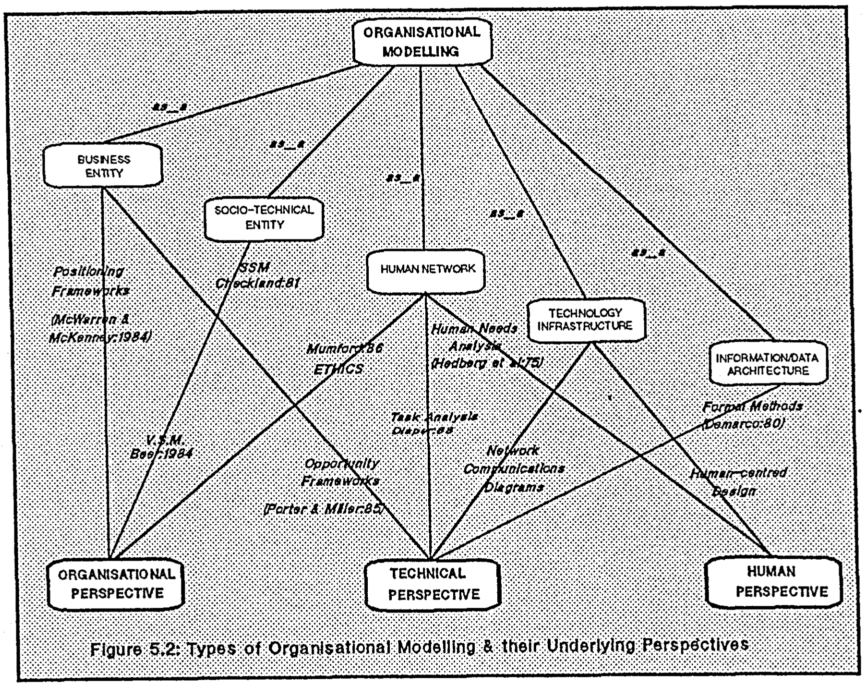


Figure 5.2: Types of Organisational Modelling & their Underlying Perspectives

Markus(1984) proposes a modelling approach by defining the business functions of an organisation and relating these to lower level activities . These activities define how business functions are implemented and in turn manage and maintain other lower level activities. This approach presupposes that the organisation is understood well enough for business functions to be related to specific tasks, and requires mechanisms by which business objectives may be transcribed into functional activities. Alternatively, business tools may be applied to models of the organisation based on different viewpoints. Strassman (1986) for instance outlines an approach by which business needs may be defined from information models of the organisation. Similarly, business frameworks, described in Appendix III, help to define business needs and priorities and provide a framework by which to assess the business potential or business impact of new technology.

5.2.2. Modelling the Organisation as a Socio-technical System

The potential benefits of an organisational model from this viewpoint are twofold: firstly, it defines problems and conflicts in social and technical terms and therefore defines the interrelationship between organisational needs and technology impacts; second, it could help to provide a specification of the role of the intended ES in the client organisation and thus lead to a useful initial specification of its functionality. However, the possible modelling tools available are limited in this sense since they start modelling at a 'conflict level' without defining a wider context from which the conflict may be viewed: moreover, the model is required to identify opportunities as well as problems. Markus(1984) by contrast emphasises modelling as a statement of the whole organisational process rather than at the perceived level of the problem; indeed she observes that the problem itself may only be a symptom of the fault, rather than a cause, which may exist elsewhere in the organisation. Moreover, the problem will have reverberations throughout the whole organisation.

Checkland's Soft Systems approach (SSM) is an example of a conflict resolution approach and is described in Appendix III (Checkland 1981). Despite this shortcoming, it provides a means by which to elicit and relate different individual roles to different viewpoints of the organisation. For instance, executive management may view the organisation as a business entity , whilst a systems analyst may look at the organisation in terms of computer hardware and information flows. By combining viewpoints therefore, different facets of the problem are identified. As a conceptual device, SSM provides an approach towards attaining a consensus on organisational problems and offers a useful first step before technology assessment. However because of its focus upon soft issues at a high level of abstraction, it is less appropriate at lower level, physical and data viewpoints of the organisation.

There are other approaches which have used system ideas for organisational modelling. Beer's Viable Systems Model (1985) for example, provides a tool to study the organisation holistically , analysing the structure of organisations from different viewpoints and levels. Similarly, Markus's organisational prototype concept may be useful to establish the socio-political structure of the organisation.

5.2.3. Modelling the Organisation as a Human Network

Some form of organisational modelling is also necessary at the individual level since a large number of people, directly and indirectly, will be affected by the introduction of ES. As with strategic modelling, these requirements, and the objectives for doing so, will be distinct and separate from other modelling roles. Diaper (1988) proposes an input-output model which defines individual roles associated with job functions. From the linkages between roles, a 'first-pass' organisational model is produced from which more detailed investigations are carried out to elicit particular aspects of information and knowledge. This model is useful in highlighting political and responsibility relationships between individuals, but is orientated towards knowledge elicitation as a modified form of task analysis.

The value of organisational modelling from this viewpoint is that it helps to identify individuals likely to be affected by the proposed system and therefore to be involved in the development process. Diaper noted that where there were time and resource restrictions or the organisation was very large, a means of limiting the scope of the organisational model was to identify relevant personnel, in this case direct users, indirect users and domain experts.

This viewpoint acknowledges that as there are hard and soft modelling approaches of the organisation and that these are perpetuated by different orientations of the individuals. Therefore in the same way that the modelling approach should provide a basis for hard and soft assessments, so it should be available and understandable to numerous background of people at different levels in the organisation. Diaper explains that using organisational modelling in this way allows for a more 'people orientated' assessment in that rather than addressing what tasks the proposed ES should undertake, the focus is shifted to what the individual requires from an ES by a formal understanding of the tasks listed by the organisational model and an informal recognition of social and political relationships with other functions. It may also help to anticipate possible impacts before the system is implemented, as Diaper adds:

> " Predicting early in the project, the consequences of introducing the proposed expert system to the organisation is likely to lead to a more co-operative perspective from those who will be subsequently involved in its development . It should also reduce subsequent creep in the project as the function, purpose and users of the expert system are clearly defined. " (pp13: 1988)

5.2.4. Modelling the Organisation as a Physical / Technology Structure:

At the functional level, organisational modelling may be used to define organisational activities in terms of machines, plant, resources and tasks and flows (information, materials, money, commands and so on) between these. At its most basic level, these models may define the architecture or technology layout of a particular computing system or function. Modelling from this viewpoint allows an investigation of what is required and may help in assessing the implications of implementing a particular technology in the organisation. However the main purpose of modelling is to impart an understanding of the situation in physical and technology terms rather than provide a rigourous analysis or design technique. Consequently, an organisation may develop its own informal notation for particular applications. Alternatively, there are functional models, such as IDEFo and BIS, which provide a standard notation and add a modelling discipline.

Despite their potential to be used in numerous phases of ES development, such models have been restricted in use as a communications device between members of a development team, and as an aid to knowledge elicitation.

5.2.5. Modelling the Organisation as an Information & Data Structure:

Modelling from this viewpoint takes an information processing perspective of the organisation and therefore defines a 'system' as being made up of data sources, information flows between transactions and data sinks. It is inconceivable that modelling at such a level of detail is practical at the company level, and therefore such an approach is used only within the boundaries of a chosen information systems design or problem domain. In ES development these techniques have been used extensively for mapping decision-making structures during knowledge elicitation; and are also used during the later stages for design specification and maintenance. Their emphasis upon formal methods and explicit data structures however, does not allow them to show essential human inputs to information transactions, such as the element of judgemental reasoning in decision-making. Furthermore, they are not designed to show the criticality of information or data and therefore for both reasons require higher level viewpoints, such as business and socio-technical, to provide a context for their use. The issue of combining viewpoints in a single modelling approach is discussed next.

5.2.6. Combined Modelling Requirements

The need to undertake organisational modelling of some form was heightened through discussions from an external organisational perspective in the last chapter. This perspective points out that the investigation and analysis of ES potential should begin not by evaluating the technology, but by defining organisational needs in the first instance. Three main reasons were given: first, to ensure that organisational needs and characteristics constrain technical design, development and implementation; second, to ensure that the organisational problem is understood and agreed; and finally, to predict possible impacts of embedding technology upon the organisation and its people. *Figure 5.2.* indicates that there is a possible hierarchy of modelling capabilities, corresponding to different tasks and needs at various levels in the organisation. Analysis of each of the five viewpoints reveals that each is necessary for development at different stages of the activity framework. However, of the modelling approaches reviewed, none adequately represent other viewpoints because conceptually they conformed to a single paradigm, or structurally, they were unable to be applied in an acceptable way (e.g. too 'soft' in a highly formal environment etc.). Similarly, approaches tended to be either top down or bottom up whereas in order to link business and human factors with technical design criteria for instance, it was necessary that both top-down and bottom-up viewpoints were embraced within the same framework. In that MPC highlights the necessity of combining perspectives in assessment in order to gain a true representation of the problem, it is apparent from *Figure 5.2.*,that in order to achieve this structurally, it is necessary to combine different viewpoints of the organisation within the same modelling and assessment process. This places a number of demands upon an organisational model:-

i) **Cross-paradigms:** that the model is sufficiently flexible to incorporate different viewpoints and processes and that accordingly, it may be applied to different aspects of the organisational problem. Structurally, in order to facilitate a cross-paradigm analysis, it is imperative that the modelling approach does not presuppose a particular viewpoint initially and therefore is not intrinsically biased towards a particular structural approach before the inquiry process. For this reason, it is also important that the model is functional and relevant company wide. Modelling the organisation according to the boundaries of the organisation or organisational unit should not be used to define the context of a system because, as Markus points out, it prevents analysts from identifying and recommending required changes and it may encourage managers and developers to try and use systems to reconcile organisational problems.

ii) **Cross-level:** that the model is comprehensible to different people in the organisation. In order to achieve cross-paradigm modelling, it is necessary that in the process of modelling, the same technique is used at different levels of the company and therefore different groups of people with specific organisational roles and personal needs. The modelling process should therefore be 'logical' so that it may be adapted to different viewpoints and also adopt a standard notation in order that it may be understood across disciplines and at different levels in the organisation. Markus notes that organisational analysis involves the support and development of managers and functional personnel with the political and hierarchical weight to pursue such investigations with other parts of the organisation. By contrast, technical analysis Markus argues, is restricted to analysts and developers who have the technical skills to undertake the technical analysis, but lack the management skills to place the study within an organisational context. Furthermore, the tools used by analysts are appropriate only for assessing the technical feasibility of the proposed solution rather than questioning the organisational problem. In the approach by Lundeberg et al.(1981), this problem was resolved by formulating a design team made up of a mixture of managers and analysts and representatives of all other groups likely to be affected by change.

iii) Cross-disciplinary: that the model may be adapted for use with tools and techniques and modes of assessment specific to each organisational viewpoint. Lunderberg *et al.* for instance describe an approach whereby they use a graphical technique to define existing organisational activities and then use the same graphical conventions to later document the design features of the proposed computer system in a particular area. Thus the technique is used in the first instance to understand problems and needs of the people and groups consulted, from which the design team generate *change alternatives* for each identified problem area or issue and evaluate these against human social, technical and economic criteria. A modelling approach which is able to span different viewpoints in this way has advantages in that it incorporates assessment and evaluative processes of different conceptual backgrounds together within a common structural framework.

So far, the characteristics of modelling techniques have been defined in terms of their ability to perform certain tasks and their underlying viewpoint. A third consideration is the application context as this may influence or re-define the way in which the model is actually used when applied in an organisational setting. This interrelationship is summarised in *Figure 5.3.* This figure shows that in assessing modelling alternatives, possible constraints may be structural, in that the modelling technique fails to perform the task requirements, whether it is at the information level or at the data level; or conceptual in that the underlying perspective of the proposed modelling technique is inconsistent with the particular view of the organisation, or is inappropriate for the application context or settings.

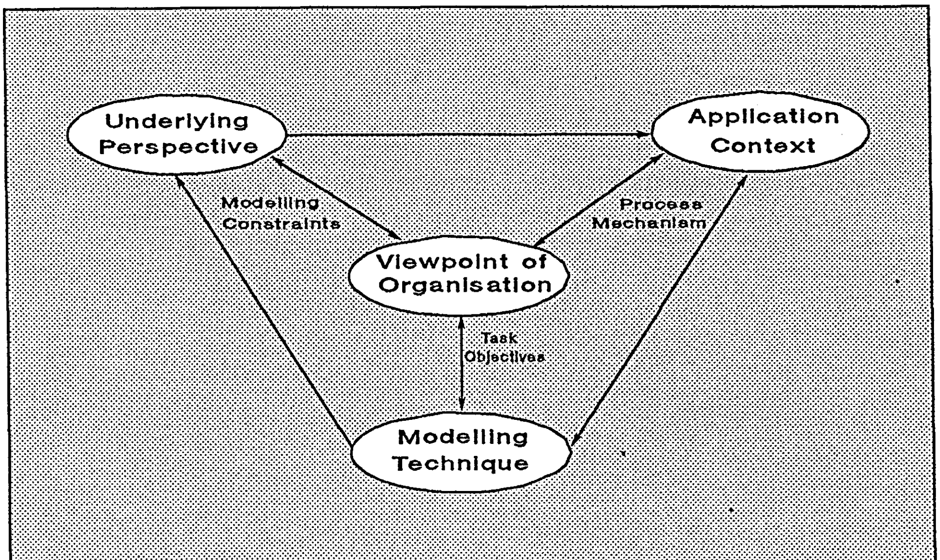


Figure 5.3:The Interrelationship of Viewpoints, Application Context & Technique

The application context varies according to the level in the organisation at which the modelling takes place, the people involved and social and political factors which will determine how the modelling process is carried out and used as a deliverable. Thus, the value of the chosen model must lie in its ability to be applied to a number of tasks, contexts and settings. As with the development context described in Chapter 4, the application context may be understood by using Multiple Perspective Concepts (MPC). As in *Chapter 4* also, it is useful to distinguish between the formal use of tools in yielding 'hard' results, and the informal use of the same tools in order to accomplish different goals. However, although the organisational modelling process may provide opportunities to undertake such alternative forms of assessment, it is important that the correct initial choice of modelling approach is made so that the design of tool fits the modelling objectives as closely as possible. This requires further insight into the client organisation and their expectations and motives for modelling the organisation.

5.3. Company Objectives in Modelling the Organisation

The reasons for modelling the firm are analogous to use of formal methods in design, the difference being that in the former the 'system' constitutes computer hardware or software whilst in the latter it represents an organisation. Thus when there are numerous people involved in a system development, formal methods enable developers to interchange ideas and communicate on a common basis. They also ensure that separate components fit together accurately, and when systems are modified, clear diagrams aid maintenance and make it possible for new team members to understand how the program works for instance, and allows developers to understand the possible effect of any changes in design. When debugging the system, diagrammatic notation may also help in understanding how the system ought to work and for tracking down what might be wrong. Unlike the use of formal methods in modelling computer systems though, modelling the organisation places more demands upon not just the choice of modelling technique but *how* it is used in the organisation.

The motivation for an organisational assessment in the Client Organisation was borne from an independent study which assessed the company's Information Technology strategy and in particular it's commitment towards Computer Integrated Manufacturing (or 'CIM'). A recommendation of this study was that it was first necessary to understand the organisation, its problems, structure operations well enough for practices and procedures to be simplified and rationalised prior to computerised integration. From this it was possible to define and express a primary aim for undertaking an organisational modelling exercise of this fashion:

> "to compile a comprehensive, definitive and coherent model of the operation of the company's business. This should describe the main functions of the business, their interrelationship and information flows. The model should be structured from the top-down and should be defined to a sufficient level of detail to enable problem identification and rectification to be achieved."

This was to be used as the 'motherhood' statement for the project; what was less certain was how this could be achieved and from what viewpoint ? During the assessment of possible modelling techniques, it was clear that the reasons for modelling the organisation defined by senior personnel corresponded closely to those of the five viewpoints of the organisation. This is demonstrated in *Figure 5.4.* These different viewpoints had influenced previous attempts in modelling parts of the organisation for specific purposes and had expressed disparate objectives for the most recent company wide modelling attempt.

a) Modelling the Organisation as a Business entity: As Figure 5.4 shows, specific objectives formulated from this viewpoint included, "to clarify and make more visible the company's overall business strategy" and "to define our business related objectives for CIM, and hence re-define our CIM strategy. The organisation had a clear statement of business goals and objectives, but less of an understanding of how these may be applied in terms of a business strategy and technology strategy. It was acknowledged that strategy development was critical in order to regulate technology development in accordance with long-term business needs and provide measures by which to monitor and control organisational performance. The attractiveness of a global organisational modelling process was that it provided the opportunity to systematically define business activities and identify problems , issues and conflicts. It was also requested that relationships and information flows between suppliers, customers and the corporate head office be included in the mapping process. These needs necessitated a 'top-down' mapping process in which business strategy re-define the functioning of computer systems in the organisation.

b) Modelling the organisation as a Human Network: No modelling exercise had been undertaken from this viewpoint other than by defining job functions and control hierarchies within an organisational chart. This reflects the difficulty of defining implicit personal needs, political relationships and individual exchanges (power-based or task based) at the organisational level other than in an explicit aggregated form.

c) Modelling the Organisation as a Socio-Technical System: Objectives formulated from this viewpoint were difficult to define, least well understood and difficult to justify company wide. To "prepare for change" was the final label chosen to describe this viewpoint and represents a move to understand the organisation processes in terms of social relationships, structure and requirements so that organisational layout and practices may adapt to new technologies and environmental constraints. A worthwhile attempt at modelling the organisation from this viewpoint made use of Checkland's Soft Systems Methodology as described in the last chapter and Appendix III. This exercise will be described in some depth because lessons learnt from its use provided useful ground-rules for the viability of future modelling approaches in the Company.

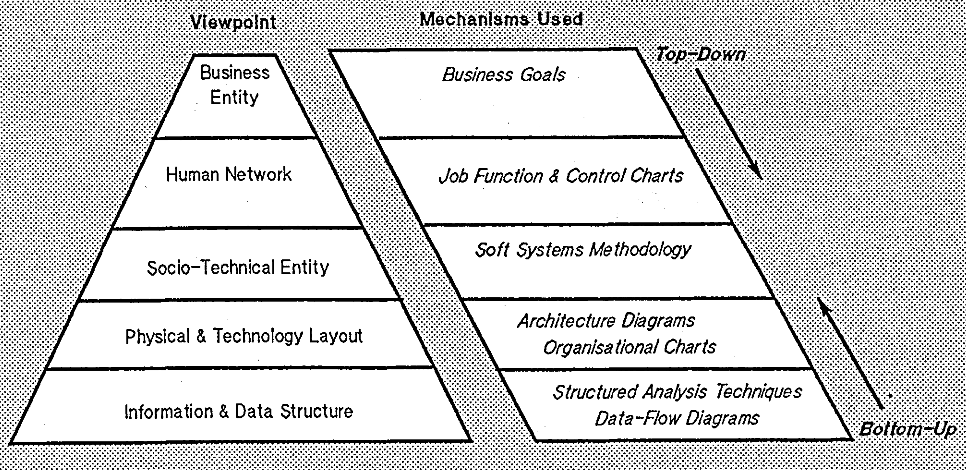


Figure 5.4: Company Expectations Of Organisational Modelling

The modelling project focused upon a particular organisational problem based upon the functioning and interrelationship between engineering and planning functions. A number of conflicts and breakdown in procedures arose based upon inherent differences in purpose and interests: engineering were design orientated and concerned in meeting the customers' specification; whilst planning were concerned with the design's "manufacturability" and the ease by which it could be produced. The problem was less concerned with the internal functioning of each group than with their interfacing: how to make formal and informal communications and feedback between each function most effective. This required a consensus between each group of the inadequacies of the present situation, change requirements and mechanisms. The value of modelling using SSM (in all thirty five models were produced) was as a vehicle for discussion and further co-operation. The models themselves however, did little to resolve the problem for a number of reasons:-

i) Too soft: although SSM appeared a valid choice of modelling technique, it turned out to be inappropriate in the formal and structured environment in which it was applied. Thus a pre-requisite in the selection of modelling technique is that the model itself reflects the organisational context.

ii) Abstract: most of the individuals consulted in the study were engineers with an implicit technical bias: many found the approach too "abstract" and difficult to relate to.

iii) Champion: the instigator of the project was a senior manager within the planning department. A number of members of engineering therefore perceived the modelling exercise as a political gesture and were less likely to participate freely and openly with the result that the modelling viewpoint was partisan.

iv) Company wide: SSM is essentially a diagnostic tool with modelling orientated about an initial conception of the problem. What it failed to do therefore is define causal relationships with other functions in the organisation in terms of impacts and the ability of other functions to re-define the initial problem. To resolve this 'local problems merit local solutions' approach, it was recognised the modelling approach should cover the whole organisation rather than commence at the problem level.

d) System Architecture & Technology Layout: Extensive modelling of parts the organisation took place in terms of defining the physical layout of a department or company wide systems architecture. For example, detailed network diagrams of communications between computing systems were produced in planning for computer integration and automated manufacturing. Similarly, diagrams of plant and machinery were produced by planning and engineering functions showing the sequencing of resources on the shop-floor. Activity models were also produced by project planning functions describing the sequencing of tasks to be undertaken, together with the major functional inputs and outputs to each task. In all these cases, modelling techniques were used to perform specific design tasks rather than model the organisation as such, although some of the functional tools used could be applied to this purpose.

e) Information and Data Structure: Specialist IT and engineering functions of the company favoured information processing and data modelling approaches of the organisation which would allow them to identify information and decision-making anomalies from which to specify physical computing enhancements to existing systems. Previous studies had made use of structured analysis models of specific activities earmarked for computerisation and even more detailed data models for specifying computer programming requirements. As Figure 5.4. shows, In this sense these model were 'bottom-up" in that they depict information and data relationships and therefore are practicable only when limited in scope.

5.4. The choice of IDEFo for Modelling the Organisation

These different expectations from the same modelling techniques placed a number of constraints upon the options available:-

i) the technique could be applied universally in the organisation,

ii) the modelling technique should be comprehensible to managers and commercially orientated staff in the company as well as systems developers and engineers,

iii) the model was not the 'end-state' but could be enhanced to meet different tooling needs from more than one viewpoint.

iv) that the technique provided standard notation and formal documentation and was equipped with procedures for implementation and validation,

v) that the technique could be learnt and applied company wide in less than six months,

vi) that the modelling process was necessarily top-down rather than bottom-up, and therefore motivated by business and organisational factors rather than by technology.

It can be seen that many of the constraints defined by the organisation correspond to the requirements of a modelling approach for expert systems assessment described earlier in *Section 5.2.* (and indeed as a precursor to any new technology assessments) and therefore the use of the model would be of additional value to the company. On the basis of the above criteria, six possible modelling techniques were reviewed, the attributes of which are summarised in *Figure 5.5.* These will be discussed according to their principal viewpoints:

a) A Soft Systems Approach Using SSM: *Figure 5.5.* shows that this methodology takes a soft approach in the diagnosis of organisational problems and make use of simple bubbles and arrows notation to denote hard and soft processes and primary flows. Its value lies in the soft modelling technique by which perceptions and viewpoints may be defined. This approach was not adopted however, for the reason that it had failed in the organisation on a previous attempt as mentioned earlier: although this was in part a vindication of the management of the project that the methodology itself. However, company management envisaged difficulties in understanding the conceptual models produced by Checkland's approach, particularly by systems analysts and engineers. Moreover, it was felt that SSM was orientated towards systems diagnosis of a particular organisational problem and less relevant in understanding and modelling the whole organisation.

b) A Hard Systems Approach Using Structured Analysis Techniques: The attraction of using these techniques to model the organisation were that they provided a 'discipline of modelling' by providing standardised notations and control and documentation procedures. Moreover, the advent of automated development tools allowed computerised modelling to take place which would reduce the time taken to model. It would also improve the quality, consistency and final presentation of diagrams. Despite these potential benefits, the detailed rigours of techniques such as Demarco (1980) and Gane and Sarson (1979), as *Figure 5.5.* shows, with their formal requirements orientated about data modelling and software systems, were not considered appropriate for the development of a general model of the organisation. Rather that they were useful enhancements to the development model. Structured tools had broadly the same notations with a process symbol indicating an information transformation of some form and information flows linking data-sources and terminating at data-sinks. These techniques were considered too specialised, complex and difficult to learn and apply given the time restrictions, and likely to be incomprehensible to non-computer staff.

The potential of modelling the organisation using formal methods and particularly Petri-nets (see Peterson:1981) was suggested on the basis that it could be used to identify parallel relationships and synchronous activities and information flows. Furthermore, the notation was simple to understand based as it was upon information linkages between circles denoting information (or physical) processes. Wyatt(1988) points out that modelling systems using such formal methods are precise and avoid the problem of ambiguity. However, despite the simplicity of the graphical notation, there are complex mathematical relationships defining each data entity; and although petri-nets can be used at a high-level to model physical systems, it was considered too complex and of limited value as a technique for modelling the organisation.

Model / Factor	S.S.M.	IDEFo	B.I.S.	Demarco	Ganes & Sarson	Petri-Nets
Principal Viewpoint	*Socio-technical*	*Physical*	*Physical*	*Information*	*Information*	*Information*
Decomposition	*Perceptory*	*Functional*	*Functional*	*Data-driven*	*Data-driven*	*Data-driven*
Tool or Methodology ?	*Soft Methodology*	*Functional Methodology*	*Structured Tool*	*Structured Tool*	*Structured Tool*	*Formal Method*
Notation	⬭ *Process*	▭ *Process*	▭ *Activity*	○ *Process*	▢ *Operation within overall procedure*	○ *Process*
	→ *Flow (hard & soft)*	→ *Flow (hard) Input or Output*	→ *Flow (hard)*	→ *Information & Data Flow*	→ *Information & Data Flow*	↔ *Data flow*
		↓ *Control*	⬭ *Source or destination of data*	= *Datafile*	▭ *Source of Information*	
		↑ *Mechanism*	⊏ *Store or Sink*	▭ *Data Source or Sin*	⊏ *Data store*	

Figure 5.5: A Review of Candidate Modelling Techniques

c) A 'Mixed' Approach Using Activity-Based Modelling: For a company wide modelling exercise it was decided that an activity-based technique would be most pragmatic and would provide scope for various secondary (hard and soft) assessments corresponding to different viewpoints in the organisation. Two techniques in particular were reviewed, IDEFo (*see later*) and BIS (1989). These techniques both modelled activities in a hierarchical and top-down approach; and described resource (materials, goods, money, people, etc) as well as information flows. Although BIS defined more symbols (like Demarco and Gane & Sarson it too had data sources, sinks and processes), IDEFo was more descriptive because it distinguished between constraints and mechanisms as well as inputs and outputs. A particular benefit of this is that computer systems (and potential technologies such as expert systems) could be modelled as mechanisms and the *use* of these systems as activities: consequently, the emphasis of the model would be on what is done rather than what is used. A further weakness of BIS compared to IDEFo was that as a 'tool' it provided few guide-lines on how modelling should be undertaken. Finally, *as Figure 5.5.* indicates, BIS is used extensively for structured analysis and is more suited for software development than organisational modelling. By contrast, although IDEFo was originally designed to model manufacturing systems (ICAM:1981) it has been used extensively in the UK to model whole organisations (CAPM: 1988). There are further documented benefits in using IDEFo which led to the decision to adopt it as the company's modelling tool. These include:-

i) Mixed Modelling: IDEFo combines the benefits of structured analysis techniques with interviews and 'walkthroughs' between the modeller and the person being interviewed. This allows models to be produced which are technically valid and are modelled from a common viewpoint (Koriba:1988).

ii) Flexibility: The model can help to identify activities, information flows and sequences between activities, together with the means to accomplish these and the conditions under which they take place (ICAM:1981).

iii) Implementation: The IDEFo methodology includes procedures for developing models by a large group of people, as well as integrating support and project control procedures into the methodology. It also provides guidelines for defining and distributing interview, collation, modelling and verification work (Maji:1988)

iv) Top-down: It is possible to progress from abstract representations of the organisation down to detailed information processing levels within the same hierarchical model. For example, at a top-level company activities are represented in strategic and planning terms and at lower levels, specific functions and information flows are defined in operational terms. (ICAM).

As with all tools though, their effectiveness rests in the way in which they are used. The next section describes the rudiments of IDEFo after which its introduction and use in the company is discussed in some detail.

5.5. The Basic Concepts of IDEFo

In the early 1970's, Ross (1976) proposed the Structured Analysis and Design technique (SADT) in which through the successive decomposition of activities into lesser tasks, and limiting the amount of information portrayed on any one page to six or fewer elements, organisational and system problems could be simplified to produce a more comprehensive and manageable analysis. This idea was refined by the US Airforce's programme for Integrated Computer Aided Manufacturing (ICAM) which proposed the use of structured methods for applying computer technology to manufacturing. IDEF (ICAM **DEF**inition) was developed as a suite of modelling techniques designed to capture graphically the characteristics of a manufacturing environment. IDEF addresses three aspects of the manufacturing system: first, what functions are being performed? Second, what information and data is needed to support these functions ? And finally, what changes to functions and information occur over a period of time ?

There are thus three corresponding divisions of IDEF methods, known as IDEFo, IDEF1 and IDEF2 which are designed to address the above questions. Each is necessarily based on the other, beginning with IDEFo which produces a functional model of the manufacturing system. From this IDEF1 is used to produce information models and a derivative of this, IDEF1x provides a data modelling methodology. These structures provide a basis for IDEF2 , a dynamic modelling technique that describes graphically the time-variant behaviour of the functions and information. IDEF1 and IDEF2 models adopt a viewpoint of the organisation as a series of information and data structures. This is useful for the design of computer systems in that it allows actual information requirements of the organisation or system to be identified and through formalised graphical representations provides a precise understanding of the structure of information. The validity of both models depends upon the accuracy of representation of activities at the system's level defined by the functions in the IDEFo model. IDEFo provides higher level description of the system in terms of activities and resource flows . It is therefore more suited as a general purpose modelling process in that unlike either IDEF1 or IDEF2 it does not have a pre-defined role, but may applied at various levels of abstraction according to different viewpoints of the organisation. Furthermore, IDEFo is used more for specification, whereas IDEF1 and IDEF2 are designed as analysis methods.

5.5.1. IDEFo: A Functional Model of the System

IDEFo provides a tool for modelling the relationships between activities and flows between functions in a system. Systems may be hardware software or organisations, and flows may be information, data, materials or any thing that is processed or handled by the activity. As a descriptive model, it has been used to understand how the current organisation (known as the 'AS-IS' model) operates and thereby highlight possible deficiencies (Maji). It also provides a common interpretation of the detailed working of the organisation (ICAM). For new systems, IDEFo may be used first to specify the requirements and functions and then to design an implementation that meets the requirements and performs the functions. A complete IDEFo model is intended to allow developers and management to understand an existing system or organisation, propose system enhancements and evaluate their effects prior to any physical alteration. The 'TO-BE' model is how the system should function given these system enhancements.

An IDEFo model consists of diagrams, texts and glossary cross-referenced to each other. The text provides a verbal description of the IDEFo diagrams, while the glossary defines all terms mentioned on the diagrams. Diagrams are the major component of a model. All activities and flows are represented as boxes and arrows on diagrams. Each box has a unique number and a descriptive name and a note describing it. Each flow has a descriptive name and the position at which the arrow enters a box conveys one of four specific roles :-

> **i) Controls** (or 'constraints'): these arrows enter the activity box at the top and represent an incoming flow which controls the activity in some way. It may determine how or when the activity is carried out, or it may modify the process that occurs.
>
> **ii) Inputs/Outputs:** these arrows enter the activity box on the left and represent inputs to that activity. These are usually processed or acted upon to produce outputs in some modified form. It is possible for an arrow to represent a flow which is an input, but which also has a controlling effect upon the activity. For example, an activity 'produce product' may be constrained by the availability of the input 'raw materials'. In this case, the input flow is shown as a controlling flow.
>
> **iii) Mechanisms:** these arrows enter the activity box from the bottom and represent a resource that is necessary or sufficient to carry out the activity. A mechanism is not used up or converted to an output by the activity.

These box and arrow meanings are used to relate several sub-functions on a diagram comprising a more general function. This diagram is a 'constraint diagram' which shows the specific flows which constrain each sub-activity, as well as the sources and targets of the flow constraints. For example in *Figure* 5.6., a general activity 'A0' comprises three sub-activities A1, A2 and A3. Activity A2 is constrained by the output of A1 and a further control Constraint_2 ; and produces a single output Output_2, which constrains activity A3. The term 'constraint' indicates that an activity uses the material or information shown entering the box, and therefore is constrained from operating by the arrow: the activity cannot act until the contents of the arrow is provided; and the way in which the activity operates depends upon the contents of the arrow.

An important feature of IDEFo is that it gradually introduces greater levels of detail through the decomposition of activities in diagrams. An IDEFo model starts by representing the whole system as a simple unit- a box with arrows to functions outside the system. This top-level representation is known as the context diagram and provides an initial viewpoint of the system indicating a particular emphasis or in the way that the system is to be decomposed into sub-activities. For large models, this also provides a means of bounding the modelling process around functions of interest and importance so that they may be covered in greater detail. This description is then decomposed using a top-down approach to any desired level of detail. For example, Activity A2 in *Figure* 5.6., may be decomposed into four sub-activities; these would be numbered A21, A22 and A23, as shown in *Figure* 5.7. This hierarchical structure of the description allows it to be developed in a controlled piecemeal fashion. However, in order that descriptions are consistent with each other, through levels, the flows which enter and leave a higher level activity must be represented in the lower level diagrams. If a flow is an output from A2 in *Figure* 5.6. for example, it must be shown as an output leaving one or more of the sub-activities A21, A22 or A23 shown in *Figure* 5.7. Thus the parent activity A2 provides a 'bounded context' within which these sub-activities operate.

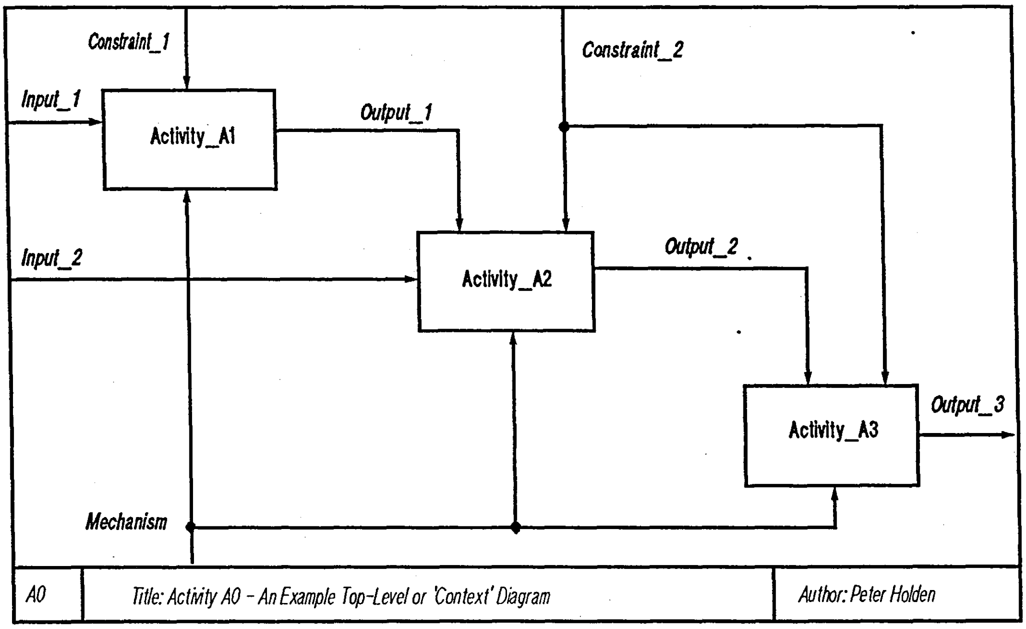


Figure 5.6 : An Example Context Diagram

Within IDEFo, there are numerous symbolic notations and drawing procedures, most of which are difficult to understand and diminish the strength of IDEFo as a communication tool. In placing this factor at the forefront of modelling requirements, a number of simplifications were made from the formal model which were used frequently in the construction of IDEFo diagrams in the client organisation. Two in particular improved the clarity of diagrams:-

a) *'Tunnelling'*: when the flows in the model are complex, it was often difficult to show them all in the higher level diagrams. In these cases, flows were tunnelled between activities. A tunnelled arrow is shown by the 'local constraint' in *Figure* 5.7. This involved ending the arrow which represents the flow away from the edge of the diagram and enclosing the description of the arrow in brackets (or a full stop in brackets). This indicates that the flow 'local constraint' reappears somewhere else in the model.

b) *Sequencing:* activity boxes are drawn along a leading diagonal and numbered from the top left corner. Although IDEFo theory does not support sequencing explicitly, the diagrams were structured in the client model so that those nearest the top-left hand corner were expected to take place before those at the bottom right hand corner.

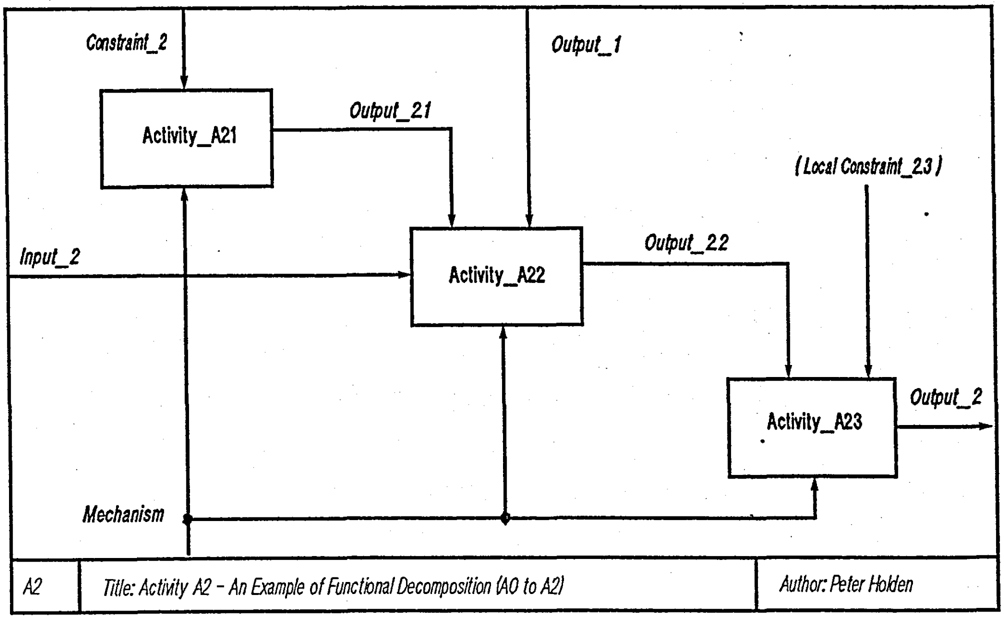


Figure 5.7 : An Example of Functional Decomposition

5.6. IDEFo Implementation

Appendix II describes how the client organisation is split between three sites: Site1 is the project management organisation dealing with the commercial and sales aspect of the business; Site2 is the head office and main administrative centre; and Site3 houses the main manufacturing facility. Due to the restrictions in time, it was decided that Site2 and Site3 would be fully modelled and Site1 functions would be modelled only when they directly interfaced with activities at the head office at Site2. The model itself would be restricted to an 'as-is' representation of the organisation, from which 'statement of requirements' and 'issues and conflicts' documentation would be produced indicating business and organisational problems and the possible means by which these may be resolved, including a potential role for expert systems.

5.6.1. Project Plans and Set-Up

The IDEFo project plan is outlined in *Table 5.1.* below and will be referred to in other sections. The project was intended to last 3 months, a large proportion of which was taken up in interviewing company staff. In all, 95 people were chosen for interviews spanning executive management and directors down to line and supervisory management on the shop-floor. The selection of interviewees was based upon an analysis of the company's organisational chart.

A team of four (including myself) was set-up specifically to develop the models, with backgrounds in systems analysis, operations management and software development. The team was to be divided between Site2 and Site3 with weekly review meetings to ensure consistency in modelling technique. Diagrams were drawn on a basic drawing package (at the time IDEFo software tools were not available) and documentation such as activity and flow dictionaries and diagram descriptions was typed out by a secretary assigned to the project.

The project itself was 'championed' in the organisation by the Company Computing Manager and the Technical Director. It is clear from this therefore that the motivation for the model and the people involved in modelling had a strong engineering and computing background. A necessary first step in the project was therefore communicating a general organisational modelling role for IDEFo through 'awareness presentations' to senior managers in the company. For the model to be successful, it was important that directors' support was forthcoming and visible to participants in the study.

Table 5.1. IDEFo Project Plan

Activity	Month -1	Month -2	Month -3
Awareness	▨		
Interview Directors	▨		
Draft Model & Verify	▨	▨	
Define Modelling Viewpoint		▨	
Interview Senior Management		▨	
Draft Model & Verify		▨	▨
Interview Relevant Staff	▨	▨	▨
Draft Model & Verify		▨	▨
Compile 'First-Pass' Model		▨	▨
Verify Final Model & Document			▨
Present Final Model & Reports			▨

5.6.2. Defining a Common Viewpoint

Because of the size of the organisation, it was impractical to attempt to model all functions. Instead, it was proposed that the model adopted a focus or common viewpoint which would define the operational boundaries and therefore limit the scope of the model in a structured and acceptable way. The implication of modelling top-down using IDEFo is that a business viewpoint of the organisation is expressed first. Thus, it was agreed that business goals should constrain the direction of the model. The logic of a business-driven modelling exercise was appealing to directors, but they were reluctant to formally present strategic information to the team from the onset. Furthermore, business priorities were not well defined and differed amongst directors. Since there was some confusion over precisely what business information was required and how it should be used to constrain the formulation of an IDEFo model, it was decided that the actual process of interviewing directors and modelling top-level activities may allow a principal viewpoint to be defined.

5.6.3. Interviewing Directors

The structure of interviews varied greatly in scope and presentation. For directors, questions asked were at a company level and aimed at defining external business relationships, business strategy and organisational problems. The interviews were arranged in advance and followed a series of formal questions set-down in an interview proforma which was distributed to directors prior to the interview. Each proforma varied in style and contents according to the role and personalities of the directors. However, a number of common questions were asked, as shown in *Table 5.2.* overleaf.

Table 5.2.: Interview Proforma For Directors: Basic Layout
a) What is the nature of the relationship and what are the principal interfaces between your company and other sites;
corporate head office; suppliers; market and customers?
b) How are resources committed to the company ?
c) What are the main constraints placed upon the company ?
d) What are the key strengths of the business ?
e) What are the key criteria used in monitoring the performance of the organisation ?
f) How do you view the development of the company ?
g) What changes and/ or developments do you consider most important ?
h) How is company policy implemented in terms of the strategy adopted and mechanisms used ?

Despite the formality, the information elicited at this level was abstract and the IDEFo model was subsequently high-level and more a conceptual than a physical representation of the organisation. Furthermore problems expressed at this level were of a business or organisational nature. The value in modelling top-level activities was that it provided a legitimate basis by which to identify key business activities, critical flows and organisational problems. These differed in emphasis amongst directors, but there were recurrent themes and priorities which through continual verification enabled a principal business viewpoint.

5.6.4. Verification of Top-Level Model

Verification of the models is by means of a 'readers/author cycle' where the author is a member of the IDEFo team, and readers are managers and engineers whom have been interviewed. Draft diagrams are first generated and distributed to readers for review and comment. IDEFo procedure requires that each reader is expected to make comments about a diagram and submit these to the author through writing or discussions. This cycle continues until the diagrams, and eventually the entire model, are officially accepted.

IDEFo includes procedures for retaining written records of all decisions and alternate approaches as they unfold during the project. Copies of the diagrams created by an author are critiqued by 'knowledgeable commentators' who document suggestions directly onto the copies. At a high-level, this process contributes a separate and distinct viewpoint, whilst at lower levels it is useful in verifying technical detail. Authors then respond to each comment in writing or through discussions on the same copy. Suggestions are accepted or rejected along with the reasoning used, so that ultimately it is the discretion of the modeller which dictates the form of the model. As changes and corrections are made, outdated diagrams are retained in project files. The diagrams are changed to reflect corrections and comments. More detail is added to the model by the creation of more diagrams which also are reviewed and changed. Thus the final model represents an agreement on a representation of the system from a given viewpoint and for a given purpose. On the occasion where there were irreconcilable differences between modelling viewpoints, it was common practice to seek the viewpoint of readers of higher and lower in status to the original reader in order to define the organisational setting more completely and thereby allow a 'reasonable' judgement to be made by the author, to be verified by the original reader. Continual iteration of this cycle provided a dual educational process. The author learnt more about organisational functions and processes, whilst the reader gained insights of how his or her area of responsibility interfaced with other functions from an agreed viewpoint. It was therefore essential that the reader became fully acquainted with the IDEFo model because it allowed the author to communicate in terms of functions and constraints, and also in that it allowed the reader to construct diagrams independently and reflect ideas in a common graphic language.

The way in which IDEFo was used at the business level was similar in concept to Checkland's Soft Systems Methodology(SSM) in that both are based upon what Wang and Smith (1988) call 'a learning paradigm' from finding out about a situation to taking action to improve the situation . Thus, IDEFo provides a means by which each individual model describes some viewpoint of the total organisational setting and consequently different models of the same organisation are possible to satisfy each of the key business functions involved. However, while in the Checkland methodology the 'root definition' must be *explicit* since subsequent model development is dependent on it, in IDEFo such a definition is absent and instead assumes that definitions will evolve informally and implicitly during the modelling process. This implicit soft approach was politically and culturally more preferable to the company than SSM although the same results could be attained. Moreover, as a language rather than a conceptual model, IDEFo contains many features which could enhance the quality and consistency of the diagrams and thereby improve productivity. Although this may be considered as a covert and 'hidden' use of IDEFo, other studies (see Wang & Smith:1988; and CAPM:1988) support such a role and suggest that IDEFo is a useful vehicle by which to apply soft principles in a rigourous way without sophistication nor 'intellectualisation'.

5.6.5. Delivering a 'Top-level Model'

As a result of the above efforts, a top-level model of the organisation as a business entity was reached made up of forty diagrams. This defined a common viewpoint and an implicit emphasis which influenced the way in which business activities were structured and interrelated. The context diagram for the client company is shown in *Figure 5.8* below.

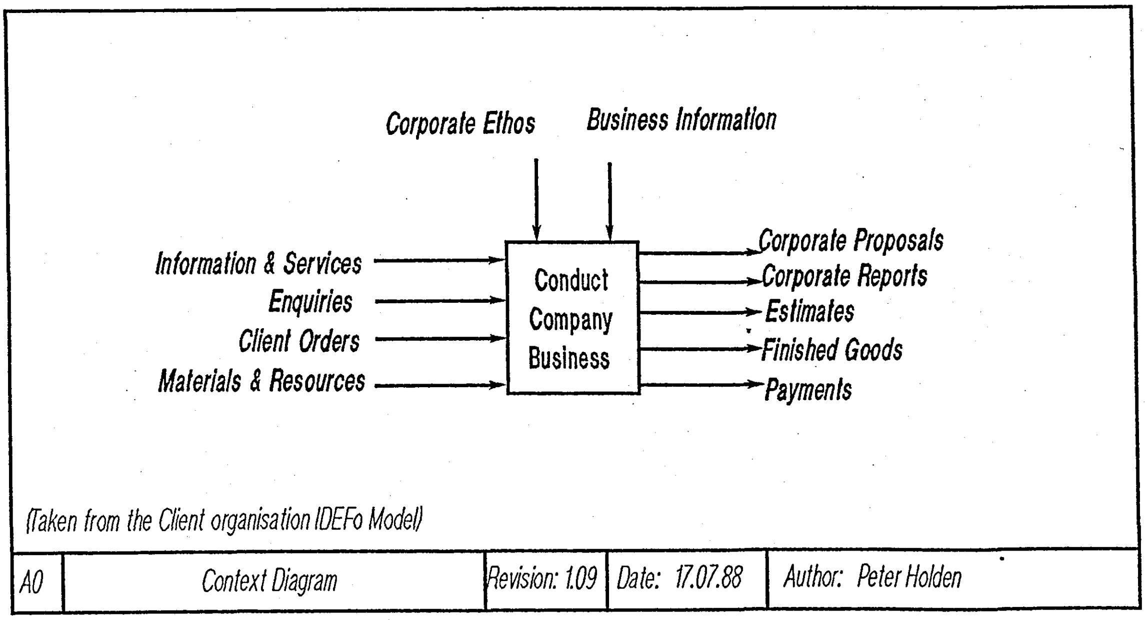


Figure 5.8: The 'Context Diagram' for the Client Organisation

In defining a context diagram for the company, it was necessary to show interrelations between each of the organisation's sites. These generated a number of 'default flows' such as company policy, corporate directives for instance which occurred in every activity, and for clarity, were not shown in the models. Default management constraints were also adopted in the model to suggest an implicit management control over a function. These include financial information and services as controlling inputs to an activity and non-contractual requisitions and budget proposals as outputs.

The principal viewpoint adopted by the context diagram was that of *'servicing a contract'* and the primary flows in this process are shown in *Figure 5.8.*. As a large contract orientated organisation, business effectiveness is determined by the way in which a tender is serviced through the organisation from initial project evaluation phases through to shipping of the finished goods.

From the context diagram, *A0* in *Figure 5.8.*, the model decomposes into four sub-activities as shown in *Figure 5.9*. Consistent with the 'servicing a contract' emphasis, the decomposition reflects the orientation of the organisation around the provision of management control structures (activity A11), services (activity A12) and needs-driven development (A14) in order to meet the demands of a particular contract or engineering project represented by the sub-activity 'conduct operations' (activity A13). At this level, the 'manage company' function is concerned in setting company policy and implementing corporate instructions based mainly on financial controls and consistent with corporate ethos and business strategy. Policy is implemented through directives which constrain the way in which company services, operations and development is undertaken. Policy is adjusted on the basis of internal reports generated by functional heads and external information on markets trends from the sister project organisation.

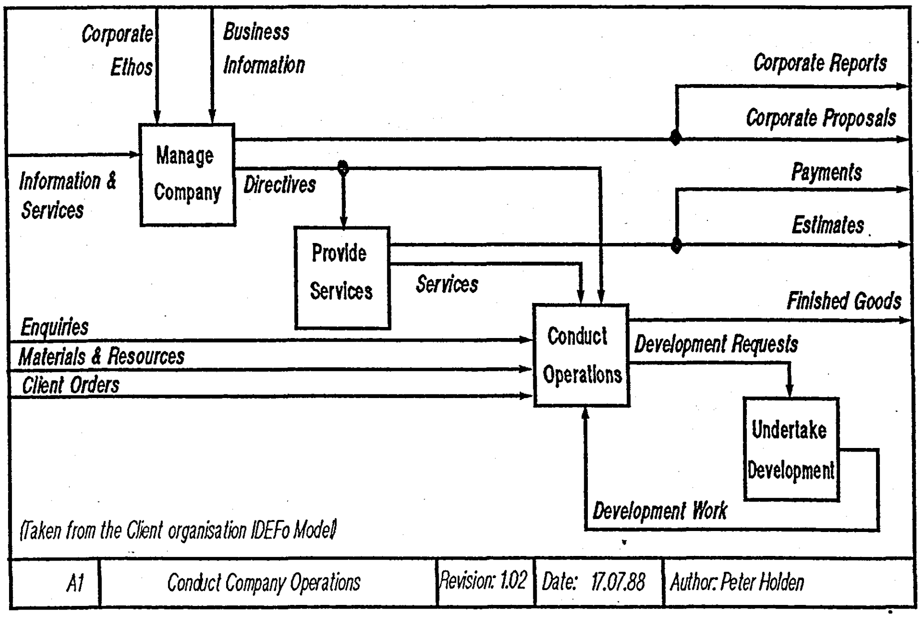


Figure 5.9 : 'Conduct Company Operations '

The necessity of allocating direct and overhead costs to specific projects is highlighted by the separation of services from operations in *Figure 5.9.* The company provides five basic cost-centred services to a project; financial, personnel, computing, quality assurance and work services (which can be anything from plant purchases for new process requirements to maintenance). These are shown as controlling inputs upon the 'conduct operations' function. Development is also cost-centred and driven by the design specification and process needs of a particular contract. This requires that the function 'undertake development' is able to meet development requests to time and budget: this may require subcontracting development work.

The first decomposition of the company shows the predominance of formal communication mechanisms at this level in terms of written directives, financial reports and a well defined hierarchy of command between management and operations. What the IDEFo model cannot do is show explicitly the organisational 'sub-culture' which co-exists within this representation, although it is possible to reflect this implicitly in the nature of the functional decompositions. In this sense, the model defines a 'minimum critical specification' (Markus: 1984) of the organisation indicating the minimum necessary for the organisation to function. Additional informal information gained through the process of interviewing readers, such as "grapevine' information and personal insights are recorded in an 'issues and conflicts' log. This enhances the organisational model by taking cognisance of the P(personal)-pespective; both are necessary and complementary in technology assessments.

5.6.6. The Business Value of IDEFo

In order not to lose sight of why the organisation is being modelled, it is instructive to make use of the business tools mentioned in the last chapter and covered in greater detail in Appendix III. to describe the possible use of IDEFo as a business model. Two particular business tools enhance the business value of IDEFo: Porter & Millar's Value-Chain Concept (1985) and Rockart *et al* analysis of Critical Success Factors (1984); both of which allow current and prospective technologies, value and impacts, to be viewed in business terms.

a) *Identifying a Value-Chain of Business Activities*

Porter & Millar's 'Value-Chain' concept describes the identification of a chain of activities which are required to be delivered internally to deliver the company's product to the customer. The resources identified with each activity can be quantified and the contribution or 'added-value' to systems operations is determined through value-chain analysis. Also the linkages between activities can be evaluated to identify opportunities for improving the delivery mechanisms and productivity of resources. Porter and Millar distinguish between primary and support activities. Primary activities are those involved directly in the physical creation of product and associated services. Support activities provide the inputs and infrastructure that allow the primary activities to take place.

From the context diagram and a defined viewpoint of 'servicing a contract', a consequence of modelling subsequent business activities is that an 'internal value-chain' is generated. It has been shown already that the organisation is structured towards providing services and infrastructure for servicing a contract. The actual operations which determine how a contract is serviced therefore define the 'value-chain' and are shown by the decomposition of box A13, 'conduct operations' as shown in *Figure* 5.10.

It can be seen that five key chain elements are involved: front-end commercial activities; the co-ordination of manufacturing; engineering; manufacturing; and accounting. By focusing technology developments in these areas, and positively changing the way these functions operate, there will be, according to Porter's value-chain theory, direct business benefits. Alternatively, by addressing support functions, shown by functions A11, A12 and A14 in *Figure* 5.9., indirect business value may be gained (e.g. infrastructural investments) by improving the way in which value-chain activities, shown collectively by function A13 in *Figure* 5.9., are serviced and supported by the organisation. The IDEFo model thus provides a means of evaluating the strategic impact and value of possible technologies, like ES, and its use in this capacity will be described in greater detail in the next chapter.

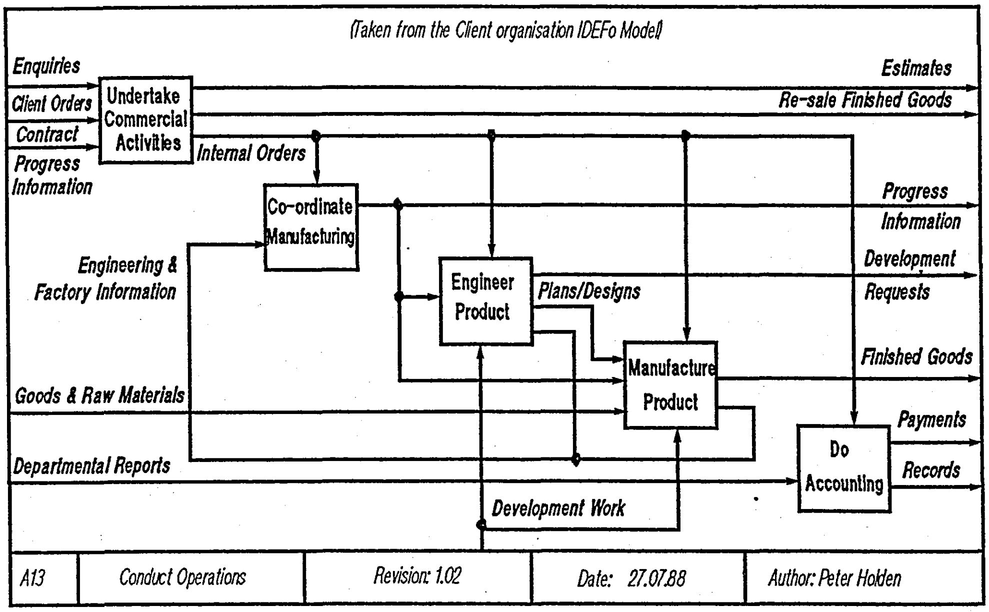


Figure 5.10 : A Decomposition of Company Operations Showing Principal 'Value- Chain' Activities

A shortfall of the value-chain approach though, is that the underlying concept belies the true complexity of 'real' company operations. (Ward: 1988) and in fact a company will likely engage in many businesses, each with distinct customers, product strategies, strengths and weaknesses and opportunities. The top-level IDEFo model shows clearly three essential *Lines Of Business* (LOB); each with separate and also shared information, resource and control structures. These are electro-mechanical control gear manufacture; the manufacture of locomotive motors; and the service and repair of locomotives. There are others, such as printed circuit board manufacture, but these may be regarded as support lines of business, for control gear manufacture in this case. What IDEFo cannot show explicitly is the strategic importance and business performance of each LOB; but it does shows, implicitly, a prioritisation through the allocation of resources, computer systems and infrastructure to each. However, because each LOB is still accommodated by the 'servicing a contract' orientation of the company as defined earlier, Porter and Millar's value chain concepts can still be utilised to emphasise the activities that bear directly on each LOB, the details of which are defined by the IDEFo model.

b) Defining Critical Success Factors

This technique is complementary to the value-chain approach and defines IDEFo activities and functions which are required to perform especially well in order to assure line of business or company wide success. For each LOB, information technology investments like ES can make a contribution to business performance and in each this contribution is specific to the needs of that business. In the service/repair line of business, for instance, the critical success factors are after sales service, inventory management and maintenance contracting. In the Control Gear line of business, the critical success factors are different and include lead-time planning, quality assurance and design specification.

A cost reduction in the 'support' areas of the company may have little bearing on the success of a particular line of business, whilst technology investments in primary activities or critical success activities will show direct impacts. Thus both approaches coupled with the use of IDEFo provide a basis for investment decisions which help to maintain a business wide perspective. A next step discussed in the proceeding chapter, is to provide the means by which to specifically evaluate the potential business roles (e.g. primary or support) and business impacts (e.g. strategic effectiveness or operational efficiency) of ES technology in affecting the company's value chain .

5.6.7. Interviewing Line Managers and Engineers

IDEFo reflects the principal O- and P-perspectives at any one level and viewpoint in the organisation. Therefore as interviewing progressed through the company down to line managers, these perspectives changed as did the structure and role of the IDEFo model. Generally, this transition was from a business model to an information processing model, although as *Figure 5.2. & Figure 5.4* show, there are five discrete levels of transition, each mirroring the organisational role of the individual.

As the focus of modelling changed, the process and design of interviews changed accordingly. At the level of middle, junior and line managers, although the interview process became significantly less formal, the questions asked became more standard and structured. This was because generally, the more junior the manager, the more approachable he or she was likely to be, but also because at lower levels in the organisation, individuals adopted more tangible viewpoints. For example, when interviewing engineers, the focus of the interview shifted from abstract questions on business strategy and opinions of organisational needs to concrete and well defined questions orientated towards identifying specific functional needs and describing physical layout and established information flows. This is apparent from sections of a proforma designed for manufacturing managers and supervisors shown in *Table 5.3*. A modelling benefit of IDEFo was that through functional decompositions, both viewpoints could be modelled using the same notation and within the same organisational model.

As modelling reached the task or information level, it was apparent that more of the problems were expressed in technical and human terms rather than organisational or business terms. Many of the technical problems were evident from the model; but human problems and personal insights could not be communicated in this way and required processing by 'softer' means (one of which was to record these problems anonymously within the 'issues and conflicts' document and use them as a basis for informal discussions on departmental improvements, re-organisation, relocation etc.). Personal needs were also considered as a criteria for ES technology selection, depending upon the criticality of the individual role in the organisation.

Table 5.3.: Interview Proforma For Operations Management: Sample Questions
Q3. Describe the operations of your area of responsibility.
Q4. Define the main information and documentation inputs into your area.
Q5. Describe the main information outputs from your area
Q7. Describe how activities, projects and information are controlled.
Q8. Which computer systems are used and how ?
Q14. What are the principal constraints on your functional area- e.g. budgetary, time, policy resource, etc.
Q18. What are the main problems in your area

5.7 An Evaluation of IDEFo

Many of the benefits experienced in using IDEFo were borne not from any special attributes of the methodology itself but from the processes of interviewing, gaining feedback and learning about the organisation. IDEFo was thus a legitimate and politically acceptable means by which to acquire in-depth information about the company. Its value as a modelling exercise should therefore be evaluated from two perspectives: firstly, the formal and direct benefits that IDEFo generated as a functional methodology; and secondly, the indirect benefits (and problems too) that IDEFo, as the first company wide investigation, permitted.

5.7.1. A Formal Evaluation of IDEFo in the Company

This evaluation can be divided into two aspects ; an evaluation of the choice of IDEFo and the information it provided the company with; and second, the effectiveness in which the IDEFo project was undertaken.

5.7.1.1. The choice of IDEFo: As a deliverable, the IDEFo model was a static representation of the organisational situation. Despite their original intentions, senior management had difficulty in deciding precisely what the model should be used for. More significant, was to decide who should be allowed to use the model since it contained some sensitive business information at higher levels and potentially damaging (political) information, showing for example the relationships between functions which came between departmental responsibilities. Moreover, at the highly detailed technical level, it was often difficult to understand the diagrams and subsequently difficult to verify the information they contained. Despite these weaknesses, there were also a number of direct benefits arising from use of the model, some of which are of value in the evaluation of ES. These are summarised below :-

i) The model promoted a thorough understanding of current functions and activities which induced management in many areas to critically consider how the business operated and reassess their roles with other functions. The model also provided a framework for the development of integrated functions.

ii) The functional decomposition of the IDEFo model clearly showed anomalies when compared against the company's organisational chart, such as duplication of tasks, poor communications and so on.

iii) The model helped to define the boundaries of a problem in terms of activities, resource and information flows whether these were as inputs, outputs constraints or mechanisms. This was also useful in identifying problems which cross functions and therefore departmental boundaries at the same level, and also causality of problems between levels.

iv) IDEFo provided a significant amount of business information in terms of critical business activities and value-chains which were more tangible to middle and lower management than 'motherhood statements', general goals and objectives.

v) The lower levels of the IDEFo model have since proved to be useful in identifying information flows between activities, particularly in the earlier stages of systems design.

vi) The model has proved in part to be an effective communications device between people of different backgrounds and has subsequently been used for training and presentation purposes.

vii) The ability of IDEFo to accommodate and integrate different views of the same function ensures that the modelling process is more robust and comprehensive, and allows a certain amount of discretion to be incorporated into the drawings.

It is inevitable that as the first company-wide modelling exercise, the IDEFo technique could never satisfy each viewpoints completely by the virtue that it was intended to be relevant to all of the five viewpoints. As a business model, the process of interviewing senior management and also explicitly defining business activities through diagrams, the top-level use of IDEFo generated a significant amount of business information which was of use in structuring lower level decompositions of company activities, and as the next chapter shows, was invaluable in evaluating the possible business value and impacts of potential ES applications. Moreover, it is unlikely that this information could have been obtained had business tools been used explicitly from the onset. IDEFo thus provided a declared and legitimate framework from which these business tools could be applied.

As the IDEFo model was further decomposed down to information and data levels, its use is analogous to structured analysis techniques and data flow diagrams such as Ganes & Sarson and Demarco (see the account by Maji:1988) mentioned in earlier sections. In common with techniques at this level, IDEFo could be used for the specification of information requirements since the diagrams are highly structured and well defined. In that the analysis is top-down though, IDEFo has the additional value of defining information flows and transactions but relating these to higher level business activities, so that the significance of information flows can be measured in business terms. However, as a tool for graphically representing information, IDEFo has a number of shortfalls compared to data flow diagrams. For instance, IDEFo fails to represent data sources and destinations (or 'sinks') clearly and does no distinguish well enough the differences between different types of data. These are notational and representation problems however, and it was recognised that once a problem was bounded and well understood using IDEFo, IDEF1, or other dedicated information modelling tools, would be more appropriate.

In discussing alternatives to IDEFo, it can be seen that perhaps a more preferable approach may have been to consider a suite of modelling tools, starting from the business level and working down to the information level, with each tool dedicated to analysis from its respective viewpoint. A similar approach is taken by the 'Multi-View Approach' (Wood-Harper *et al.* 1985) mentioned in the last chapter, in that modelling begins from a socio-technical viewpoint, and uses different tools to model information and data processes; although in commencing at the socio-technical level it fails to consider the organisation from business and individual viewpoints. Similarly, conventional lifecycle methodologies like SSADM provide system modelling techniques which commence at the physical/technology viewpoint and progress down to detailed data and entity modelling. Thus the structural difficulties in explicitly defining a suite of modelling techniques is that although it is possible to identify tools to suit each viewpoint, it is very difficult to link these tools in a meaningful way. The value of IDEFo by contrast, was that it was possible to use it in conceptually different ways to achieve certain formal and informal modelling requirements.

5.7.1.2. The Management and Implementation of IDEFo

The IDEFo project took six months to finish, twice as long as expected, and further was not complete in the sense that many of the diagrams were not verified. In evaluating whether IDEFo could have been managed and implemented more effectively, a number of issues should be considered:-

a) Volume of Work: The IDEFo model was a substantial undertaking made up of about 300 diagrams and involving over 100 managers, professional staff and selected employees. It was perhaps ambitious for four people, all of which had not used IDEFo previously, to complete the model to the scheduled time.

b) Complexity: Although the IDEFo model appears deceptively simple, it was by no means quick or easy to produce. The process of producing diagrams demanded a complete understanding of the underlying logic of the system before a model could be constructed and this could only be gained from a time consuming process of iteration through a validation-refinement cycle with interviewees. Moreover, the documentation required substantial clerical support in order that it was constantly updated with the diagrams. Added to this were problems in documenting informal systems where there was genuine confusion over the operation of a particular function.

c) Support: The project was championed by the company's technical director and computer manager and was subsequently viewed by some company functions as 'another computing methodology' rather than as a communication device and vehicle for change. Despite initial presentations and education on the role of IDEFo, there was some difficulty in gaining the support from these functions and this added to the general problem of staff being unavailable for interviewing and verification during working hours. Furthermore, since the IDEFo study was conducted concurrently with daily operations, managers and supervisors particularly, were often reluctant to depart from the daily operational environment in order to pursue what they considered as a 'long-term' project.

d) IDEFo Learning Curve: The original project estimate did not take account of the learning curve for IDEFo which was steeper than anticipated. Although there was external guidance from IDEFo consultants, they were only of use in validating the technical correctness of the diagrams. Interviewing technique, modelling and other personal and technical skills, which Godwin et al (89) identify as prerequisites for the effective use of IDEFo, were qualities which could only be acquired through experience. A second and more complex learning curve was in understanding the culture, technical layout, organisation and politics of the company. A minimum knowledge and awareness was usually required of a function or person before conducting an interview, otherwise modelling would take longer and verification would be more difficult. The process of using IDEFo itself proved to be a legitimate and accelerated means of acquiring a deep understanding of the company and its problems.

e) Drift: Veryard(1987) describes a process in the use of methodologies in which as the 'novelty' of the idea subsides, the impetus to continue the study diminishes. This was certainly the case with the IDEFo project when mid-way through the project, as resourcing difficulties were encountered, there was a lowering of visibility and a general decline in enthusiasm by both participants and team members. Senior management commitment also became less certain as on-going business problems competed for their attention.

f) Attitudinal Constraints: Throughout the study, areas of management felt threatened by the study which itself constrained the effectiveness of the model. Some managers were concerned that the study would uncover inadequacies in their area. Similarly, some thought that inefficient systems and antiquated equipment reflected negatively on their own capabilities as managers. As a consequence, activities and information flows were often described in an idealised way rather than as they actually occurred. Many of these fears arose through a lack of communication and awareness of the purpose of the IDEFo study between senior management and middle and lower management. Although many such inconsistencies could be filtered out of the model through the refinement-verification cycle, it increased the average completion time for a diagram.

g) Software Constraints: A significant amount of time was spent in both producing and updating drawings to make them coherent and consistent. This was compounded by the use of a general computer drawing package rather than a dedicated CASE (Computer Aided Software Engineering) tool. The latter had a number of distinct advantages:-

i) documentation could be associated with diagrams,
ii) consistency would be maintained when updating diagrams,
iii) a data dictionary would facilitate rapid consistency checking,
iv) data would not need to be re-keyed into many separate applications,
v) the productivity of drawing diagrams would be improved
vi) IDEFo rules could be automatically applied when producing diagrams

In view of these potential benefits a particular tool, IDEFineO, was adopted during the final quarter of the study. An example of its use is given in Appendix IV (Part B). It can be seen that the quality of presentation is improved on earlier drawings in addition to the above benefits.

5.7.2. IDEFo as an innovation: Cultural Impacts Upon the Organisation

As well as the short-term and direct effects that the IDEFo study had upon the client company, there were also longer term and less well understood impacts. Over a 100 staff were interviewed and many of the company's personnel were aware of the project and it's purpose and intentions. As the first company wide modelling exercise, IDEFo was an important innovation and was treated as a significant learning process by many of those who participated. The process of modelling the whole organisation and defining problems questioned directly and indirectly the culture of the organisation, inter-departmental roles and relationships, management attitudes, personal expectations, and business needs and priorities. The effect that IDEFo has had in these terms cannot be quantified, but its effects can be seen in they way the company has progressed since the IDEFo study. Because the resource demands placed upon a company wide modelling exercise were considered excessive, IDEFo is now used in the company at a functional level, showing as in a recent project for example, how a proposed automated tool management system could be integrated with current activities and information flows. The original IDEFo model has not been updated and therefore is increasingly of less value to the company. Despite this, it continues to be used as a context for information modelling using in-house data-flow modelling techniques. Furthermore, it has been used for training and induction purposes, and is also referred to by senior management. Indeed, the most favourable response to IDEFo was certainly by senior management who championed the project in the first instance.

Although the Issues and Conflicts report which documented organisational problems and needs gathered during interviewing, has proved to be of great use to the company, this information was discerned from the interviewing process rather than the model itself. Indeed the lessons from IDEFo appear to be that the greatest value in performing a company wide investigation are to those who actually undertake the study rather than to those to whom the model was presented. There remains therefore a preference towards modelling from a single viewpoint and also about a single function or computer system since the returns in doing so are more immediate and tangible to its sponsors. It is thus unlikely that a general company wide modelling exercise will be repeated.

5.8. Conclusions to Chapter 5.

By definition, users of IDEFo must learn more about their environment to allow them to effectively model it. This learning process proved to be a valuable basis for identifying possible enhancements and locating organisational problems that might not have been discovered had the modelling exercise not been undertaken. Modelling the company was a substantial undertaking however and although it provides an essential foundation for technology assessment, other companies may be unable to justify such an exercise for single ES development projects. In fact, it is incautious to associate organisational assessments to any particular technology project because it's usefulness to the company extends beyond single developments. Thus, the use of IDEFo should be justified on the basis that it is a valid and essential process irrespective of whether ES projects are to be developed or not.

Modelling an organisation is an innovation as much as technology development is an innovation; both have social and cultural impacts, although the effects of the former are far less visible and unlikely to be acknowledged. IDEFo provoked some controversy, generated conflict between functions in deciding what the role of systems and other functions should be, and exposed nearly half the organisation's management to a discipline of explicitly defining tasks within their responsibility. It generated large amounts of information about the organisation, including problems and areas for improvement and suggestions on how these improvements may be made (technical-a new computer system for example; and non-technical- such as rationalising office procedures).It showed how company activities were interrelated across functions and between levels and defined resource and informal flows between these activities. It also provided a forum by which to express personal problems and reflections. Often these were shared between a number of individuals in which case they were documented and formally reported.

Of the mass of information that the IDEFo project generated, only a portion of this information could be formally and explicitly represented within the IDEFo model itself. Attempts were made to diffuse this information to management through reports and informal discussions and to some extent this was successful. Certainly, the IDEFo study did change attitudes and led to discussions about organisational problems and means by which to resolve these. However, the greatest value in modelling the company using IDEFo was to those who were actually undertook the modelling exercise. It provided a means to explore and investigate the organisation, understand how and why the organisation functioned as it did and suggested priorities and an agenda for change in terms of future technical and non-technical developments. Such information may otherwise have taken some years to acquire, if at all.

The insights gained during the *process* of modelling compensated for the physically laborious task of drawing and verifying IDEFo models. Verification of activities from a number of different viewpoints is a difficult and skilled task which was only mastered towards the end of the project. It was assisted by defining a primary business viewpoint which was 'servicing a tendering contract'. This proved a highly effective means of decomposing business activities at an abstract level and enabled physical activities and information flows defined at lower levels to be related to these. There was thus a direct link between critical business activities and information and resource flows.

IDEFo fails to satisfy any of the five viewpoints of the organisation completely. However this may be construed as a strength in that by not focusing at any particular level, it may be adequately applied across all levels and therefore serves as a useful general modelling exercise. It is necessary to distinguish between formal deliverables defined by the IDEFo model, and the informal benefits of undertaking such an exercise in an organisational setting. Where business, physical/technology and information processing viewpoints were gleaned explicitly from the model itself; socio-technical and individual (human) viewpoints were acquired informally during the process of modelling.

The important issue, turning now to the value of IDEFo in terms of ES evaluation, is that some process of company wide examination should take place prior to any commitment to technology. IDEFo was used because it provided a useful bridge between viewpoints, but particularly between business models and information and data models, although individually both are served better by dedicated modelling tools. It is likely that other companies and institutions may be more suited to other modelling techniques for hard reasons- for instance what tasks need to be undertaken; or indeed for soft reasons- what is politically acceptable to the company for example. Perhaps a contingency model of organisational modelling needs can be developed which takes account of both hard and soft requirements. In either case, the necessity is that some evaluation is much better than none at all.

The choice of IDEFo as a functional methodology brought with it benefits and limitations. Wyatt(1988) notes that by being activity based, the division of systems by function is inherently more complex than division by data, due to the ambiguous nature of some tasks and the fact that many company activities are informal and cannot be well defined. A significant benefit though, is that it is flexible enough to be applied to many different situations. In its basic form, it may be considered as a combination of structured analysis and human judgement brought together as a formal discipline(Maji:1988). The importance of stressing IDEFo as a 'discipline' was important to the company for political reasons (formal methods are more acceptable to highly structured and hierarchical organisations), but also for the practical reason that such a large undertaking would require careful management and control procedures.

An original objective of the IDEFo exercise was to model the present or 'AS-IS' situation from which a desired 'TO-BE' state could be defined. During this transition, it was expected that heavy investments in computing infrastructure and plant would be required to achieve this desired state. However, a close investigation of the AS_IS model revealed that a large number of improvements could be gained through non-technical means, such as simplification and rationalisation of company activities, re-organisation and integration of functions and a re-assessment of the use of current computing systems. The imperative to innovate in new technology is often tempered when it is seen that improvements can be made by other means. An investigation like IDEFo is therefore useful because it addresses 'what is required ?' in organisational terms before it considers 'what can be achieved ?' through developing a particular technology.

Having thus defined an organisational context for technology assessment, the next chapter addresses the question of which of the company's identified problems and needs may be satisfied by ES technology, and what the likely impacts will be should an application be developed ? In this dual process of problem identification and application selection, IDEFo contributes both directly and indirectly, stemming from the formal and informal use of this tool in the company.

Chapter 6

Determining Appropriate Expert Systems Projects: From Problem Identification to Application Selection.

6.1. Introduction

Previous chapters have focused upon three inter-dependent themes: firstly, defining a conceptual basis for technology assessment and development through the adaptation of Multiple Perspective Concepts (MPC). Secondly, providing a methodological framework which allows an assessment and development approach to evolve from specific characteristics and settings of the client company. Thirdly, to impart a knowledge and understanding of the organisation, its problems and requirements, through modelling the organisation using IDEFo. It may be argued that such processes are necessary irrespective of the type of technology being considered or particular organisational circumstances. This chapter therefore narrows the argument by discussing how these concepts and approaches may be adapted and applied in the introduction and exploitation of expert systems in the client organisation. An outline of this chapter is given in *Figure 6.1*.

The 'goodness of fit' between a problem and the technology depends upon providing an effective evaluative framework by which to assess the potential for ES. This requires that a method of organisation and an explicit statement of task requirements is defined in order that an assessment programme may be undertaken in the company. The necessity of formalising problem identification and application selection procedures was in part because of contractual obligations with the company, but also because of certain organisational factors:-

i) There was no 'in-house' expertise in ES in the company,

ii) There was no immediate commitment to ES and therefore the company was more interested in understanding requirements through the provision of an evaluative framework for future ideas rather than undertaking development *per se*.

iii) The project duration was finite and therefore experiences and results required full documentation so that there was an organisational capability in ES beyond the project's scheduled completion.

iv) It was necessary to formalise ES evaluation in order to place a correct context upon this technology's development (especially in view of the pervading 'hype').

A critical stage in defining a possible role for ES technology is to identify company problems whose characteristics match those of the technology. A review of current approaches to problem identification shows an emphasis upon the technical fit between problem and technology characteristics; it is argued that although this process is a necessary aspect of problem selection, it is not sufficient in that problems are identified which are technically feasible but not necessarily desirable in business, organisational and human terms. It is thus required to place such 'technology fitting frameworks' within a wider organisational context provided in part by the IDEFo modelling exercise.

This chapter looks at how different viewpoints of key individuals such as managers and engineers favour different organisational roles for ES. A consequence of this is that each are likely to place different emphasis upon certain factors during the process of problem identification and selection. In order to ensure a more consistent and balanced approach, it was first necessary to define a central organisational function to manage and process application ideas and user requests. For reasons described, a unit in the company's computing department was created for this purpose. A first task for the unit was to provide a framework by which to educate management and promote awareness of expert systems in the company. It was important to demonstrate ES capabilities and potential to user departments in a way which stimulated application ideas whilst managing expectations. As part of this induction process, functional managers were issued with an identification check-list intended to help them identify problem areas in their location where expert systems may be of possible benefit. As *Figure 6.1* shows, it's use is as part of a wider approach of awareness and assessment, of which IDEFo plays a central role. This approach provides a listing of potential areas for development, a summary of resource requirements, and an indication of the business impact of each application idea .

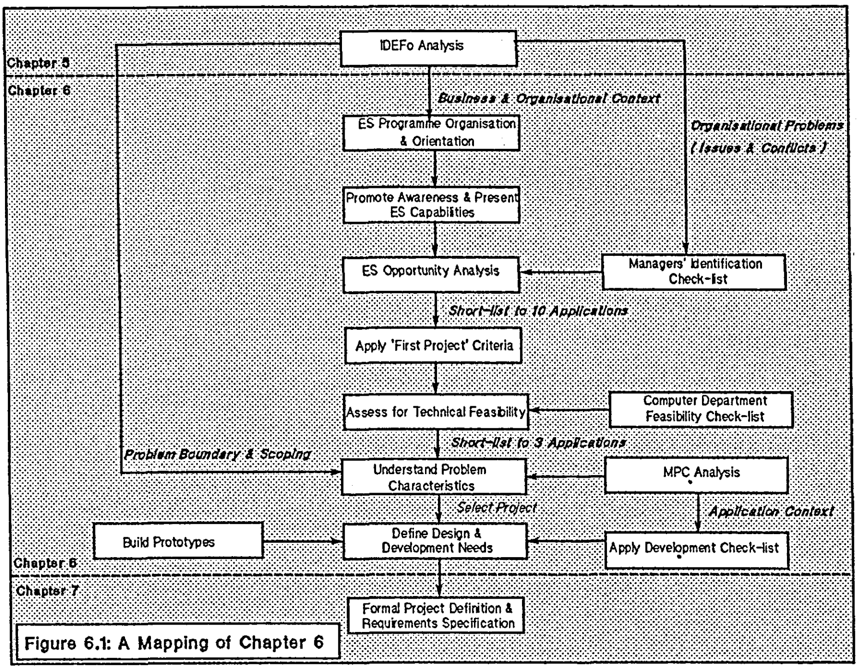


Figure 6.1: A Mapping of Chapter 6

A second assessment stage is then described which focuses upon the feasibility of the above ideas. This provides more technical and detailed criteria, again in the form of a check-list, to reduce the development alternatives to a more manageable number. It should be noted that this check-list operated within two overriding and project specific constraints: that, as a 'first ES project' it had certain organisational objectives such as promoting technology transfer which might not normally exist; and secondly that there were significant time and resource constraints associated with development. In satisfying these constraints, three candidate projects are identified. An analysis of settings and viewpoints is achieved by applying Multiple Perspective Concepts(MPC), defined in Chapter 3, and its value is described in terms of understanding the development context of each candidate project . This work is supplemented by an analysis of activities using IDEFo, which establishes the system boundary for each of the projects and their interrelationships between the domain and surrounding activities. Both MPC and IDEFo were of use in providing relative comparisons between each of the three candidate projects but could not say how well each would turn out in practice. Therefore a final stage of the selection process was to build small demonstration prototypes on a rule-based expert system from which the decision to build an expert system for trouble-shooting computer hardware faults was taken.

Although the above process of selection disclosed a lot of information about the project proposal, a lot more was required before full development could begin. Much of this centred on the need to specify development requirements and suitability as much as possible and provide a plan of action and an estimate of costs: from this, a full justification for the project could be made. Therefore development suitability guide-lines were produced and used in the design and planning of the full-scale project, the development of which is described in *Chapter 7.* These were intended as an accompaniment rather than a substitute for the company's established project planning and management methods.

There are many intermediate stages of assessment and analysis involved between problem identification and application selection phases. However, since a graphic account of each would distract the reader from the main themes of the chapter, a significant proportion of the detail has been postscripted to *Appendices 5 to 10.*

6.2 What is Problem Identification ?

Deciding whether ES are useful to an organisation is one aspect of problem identification. There are two possible extremes which may motivate an organisation to innovate in ES. Human-centred arguments, outlined in *Chapter 3*, perceive the role of technology as human facilitator. Thus, human goals and objectives constrain the selection and use of technology . A dialectic to this approach is epitomised by the 'automate or liquidate' imperative (Ingersoll:1987) in which companies must invest heavily in new information technologies and integrated manufacturing in order to remain competitive. A more useful basis by which to address problem identification is offered by Clegg who refers to 'appropriate technology' in which technology is introduced, operated and managed in a company in ways appropriate for its users and organisation (1988). Therefore, there is no imperative to develop ES, but equally human objectives and the issue of desirability is one of a number of factors including business needs and technical feasibility, in a wider process of technology assessment.

The last chapter showed that of current methodological approaches towards ES development, problem identification was poorly represented, despite this phase being defined as the most important single determinant of eventual success of a project (Skingle:1987). Sell confirms this view, "The problem identification phase suffers badly from lack of credibility, with a consequent lack of funds, support, commitment, and freedom of action that are essential for success; and it frequently adopts procrustean measures of fitting the problem to the solution" (Sell:1987;p 402). It has been argued in previous chapters that this neglect of problem identification is a product of a dominant technical perspective in ES development. This has two implications: firstly as Diaper (1988) notes, that the development process begins at a design level with little deference to problem definition and pre-project suitability; and second, that organisational and human perspectives are omitted from this process. Where attempts have been made to define an approach towards problem selection, the same focus upon technical aspects of the problem generates further difficulties in that current techniques discussed in the next section tend to be orientated about technology-fitting frameworks, which although useful, require a wider business and organisational context to be valid in a particular application context. Jain and Chaturvedi(1989) note the strong influence of vendors over user organisations in embarking on 'over-ambitious ventures' without cause for systematic technology assessment. They therefore perceive problem selection as a means of regulating the use of ES technology in user organisations.

Although a number of authors have expressed the importance of problem selection in principle, there is some uncertainty over its role and position in the development life-cycle. Differences of approach centre upon the extent to which the problem selection technique used is able to specify development feasibility and suitability prior to development itself, thus returning to the two opposing 'SPIV'and 'RUDE' models described in Chapter 4. Thus SPIV models of development presume that full specification of the problem is possible using conventional management and control techniques, whilst 'RUDE' models of development see prototyping itself as a major component of problem selection. However, in the same way that hybrid development methodologies described in Chapter 4 have emerged to take account of both these models, a similar view is taken that both are valuable in problem selection to define a method of specification and a process of evaluation. Whichever approach is taken, there should be the recognition that this phase is one of the most important of the development life-cycle. As Liebowitz states, "...By spending the time upfront in identifying the problem, savings in time and money will ultimately accrue" (1989;p3). The implication from this is that the more rigourous the selection process, the less time and effort is required in later stages in order to qualify the suitability and feasibility of a particular candidate application.

6.3. Current Approaches to Problem Selection:

The main methodological approach for selecting suitable problems for ES found in the literature features a check-list approach of listing desirable attributes for the problem. Other approaches have emerged more recently as a response to the shortfalls of this approach.

6.3.1. Check-list Approaches

The basis of this approach is to provide a check-list of attributes which describe the desirable features of an ES problem area. In its initial form, the check-list approach defined by Hayes-Roth *et al* (1984) was a simple list of rules-of-thumb, providing advice such as: seek problems that experts can solve via a telephone; choose a problem that experts can solve in a 3 minute to 3 hour timespan; choose a problem whose solutions require primary symbolic reasoning and so on. This was expanded in greater detail by Prerau(1985;1989) who offers a list of over fifty desirable attributes to check for in a problem. These attributes are organised under eight headings, and these are given in *Table 6.1.*, with some examples from each group. Prerau used this attribute check-list to narrow down an initial set of thirty possible application areas to a short-list of eight which were then subjected to further analysis and ranking. The two most promising areas were then studied in detail from which a final selection was made. Prerau's approach has been extended and modified in a number of ways to suit particular development requirements and priorities. For example, Beerel (1987) provides positive and negative *indicators* for using ES technology and provides overriding requirements for development. Similarly, Walters(1988) defines technical, economic and specific organisational and project constraints (e.g. company culture, budgetary and time constraints etc.) which should be considered during problem selection.

6.3.1.1. Evaluation of a Check-list Approach

There are specific problems associated with each of the check-list approaches associated with particular structure and focus as indicated in Figure 6.2. For instance Stow *et al* (1986) criticise Prerau's approach because the size of the check-list makes weighting the various attributes difficult , and that the two main kinds of attributes featured, those to do with technical feasibility and those to do with low cost implementation are simply weighted so that although a high cost implementation will score badly on the check-list, it may yield significant benefits to the organisation. This problem arises by imposing a highly structured and formal ranking procedure upon an incomplete set of attributes.

Table 6.1: Examples of desirable domain attributes *(Prerau: 1989; pp27-40)*

1. Fundamental Features: *e.g. There is a need to capture expertise from the expert*
2. Task Definition: *e.g. The task requires the use of heuristics*
3. Experts & Expertise: *e.g. An expert exists and is available*
4. Bounds on the task: *e.g. The task is sufficiently narrow and self-contained*
5. Domain Area Personnel: *e.g. The project is supported by senior management*
6. System Introduction: *e.g The knowledge contained in the system is not controversial*
7. The Task: *e.g. The task is such that the system can be smoothly phased into use*
8. Other Features: *e.g. There is documented information on the domain to assist*

More generally, there are a number of collective problems in the current design and use of check-lists which, as with ES development methodologies, are only evident by addressing organisational and individual perspectives as well as the technical.

a) Process and Implementation: None of the approaches address the issue of implementation sufficiently well to understand the planning, orientation and presentation of the technique to the organisation. Nor do they indicate who should manage the check-lists, or define training and resource needs Badiru(1988) offers some insight into the management requirements of initiating ES projects by emphasizing the 'triple C' principle of communication, cooperation and coordination. Although these principles were of use in sequencing the necessary tasks, they were of little use in defining how they should be performed or the issues to consider in performing them.

b) Scope of Use: A prime distinction between check-lists, as Figure 6.2. shows, is the degree of structure. In the approach of Hayes-Roth *et al*, the emphasis is upon providing unstructured and informal guide-lines as part of a wider process of analysis and investigation. However, the apparent randomness of these approaches led to calls for more formal and rigourous approaches; a view echoed by Jenkins "A variety of ad-hoc rules and check-lists have been published, but there is little evidence of attempts to draw these together or to quantify and evaluate the results of applying them" (1987). In response to this, Slagle and Wick (1988) for instance, have enhanced the approach by Prerau by systematically scoring the features of an application and combining these scores for an overall candidate value. The candidate with the highest value is then selected as the expert system application. Keller(1987) formalises this process still further by applying structured analysis disciplines in defining an evaluative approach. However, Murdock(1990) argues that check-lists should not be too structured because firstly, no list of requirements can be exhaustive for every project and secondly not every ES project can meet all of the defined criteria.

c) Scope of the Check-list: A second factor in the role of check-lists which may perhaps explain the confusion over their formal and informal use is precisely what they should be used for. Figure 6.2 shows that there are three aspects to problem selection: providing the means by which to locate organisational problems and rank these problems according to predetermined business, organisational or personal criteria; identifying which of these problems may be resolved using ES technology to define a list of candidate projects; and finally evaluating these alternatives using feasibility and development suitability criteria. Some approaches, such as Prerau's, have endeavoured to perform each of these tasks within the same check-list; whilst the situation-action framework of Jain and Chaturvedi (1988) provides a clear division between analysis requirements at each of the problem selection phases. The importance of distinguishing between each phase is that different people, organisational functions and approaches are necessary. For instance, the next sections show that during ideas generation stage, a check-list was produced specifically for functional managers and used in a highly informal and interactive way. By contrast, a second evaluative check-list was produced aimed specifically at specialist development and computer personnel and designed to systematically cover detailed aspects of ES construction. Clearly in this context a single check-list for both purposes is unsatisfactory and confusing.

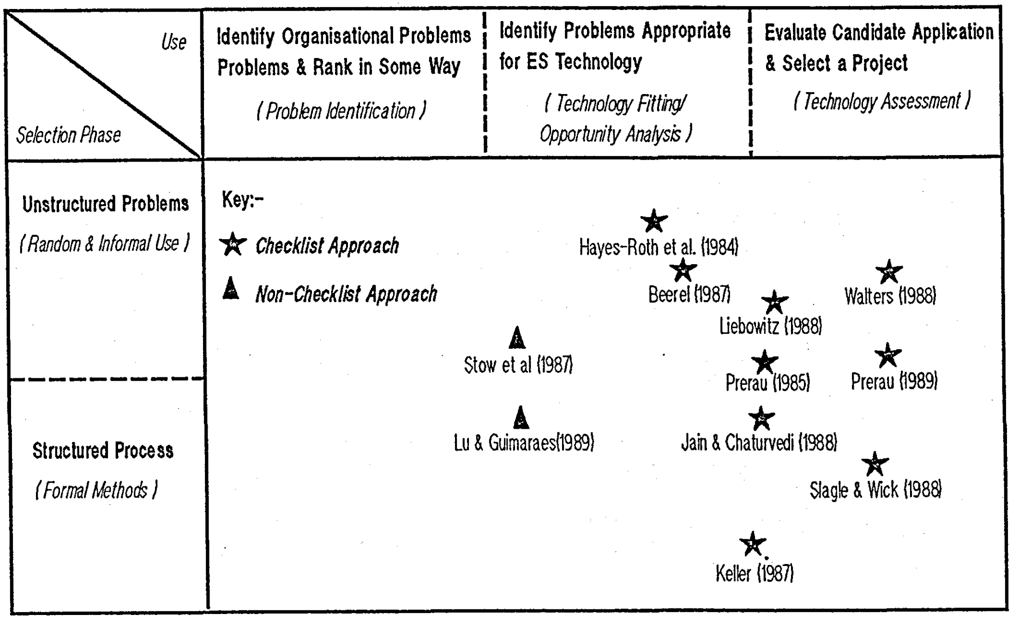


Figure 6.2: A Review of Current Approaches To Problem Selection

d) Technology Focus: *Figure 6.2* shows that of the check-list approaches reviewed which addressed problem selection, all focused at a 'technology-fitting' level, in other words they considered the suitability of problems for ES once they have been identified, rather than offering a means of finding problem areas in the first place and providing an organisational significance to these problems. The limited definition of problem identification adopted by check-list approaches has important consequences upon application selection in that the choice of project is made on the basis of the merits of a particular ES proposal in relation to other ES proposals. These techniques therefore fail to provide an appropriate evaluation context since they do not consider other possible technical solutions, such as the use of database systems for example. Equally important, they neglect evaluation of possible non-technical solutions- organisational and human changes for instance.

e) Alternatives to Check-list Techniques: Alternatives to check-lists redress the lack of attention to the first stage of problem selection, i.e. problem identification, by attributing a business value to application candidates . Stow et al for instance view ES as an extension of existing data processing tools and methods and approach identification by looking at the margins of current data processing systems. Two types of margins are identified, horizontal and vertical. Horizontal margins are described as man-machine interfaces where someone'receives output or supplies input to a system. Vertical margins occur whenever a computerised (algorithmic) process is determined by a higher-level, non-algorithmic process which is not computerised. The benefit of this approach is that it places problem selection within a wider business context so that if desired applications may be rated according to their strategic significance to the organisation. However, there are a number of shortcomings to this approach in that firstly, it does not address issues of feasibility or suitability of candidate applications. Furthermore, it defines problems according to existing technology layout and fails to consider the opinions of members of the organisation to see what they themselves consider as being important problems. Also problems may exist in other parts of the organisation beside at the horizontal and vertical margins. A second business approach towards problem identification is offered by Lu and Guimaraes(1989) who identify possible application areas by systematically locating expert-intensive areas that are considered problematic for the organisation. As with the above approach its role is in identifying areas where ES might be appropriate without providing the evaluative framework to see if this is the case. Also, the approach does not state how business priorities are identified and related to expert intensive areas of the organisation.

Although both approaches are useful in revealing the weaknesses of check-lists in failing to undertake identification from other than a technical perspective (i.e. " a solution looking for a problem"), they are only likely to be suitable as a 'front-end' to technology-fitting and evaluative check-lists. Most of the weaknesses in check-list techniques arise from their use and the emphasis placed upon each of the attributes, rather than the general principal of using a check-list approach. In fact there are a number of possible benefits in using a check-list: these include (from Slagle & Wick (1988), Beerel (1987), and Prerau:85):-

i) they provide an agenda for analysis rather than prescribe a specific one-dimensional technique and therefore may be tailored to the specific needs of the company

ii) they provide a permanent and consistent means by which to evaluate candidate ES applications

iii) the range of issues covered allows for the identification and filtering of potential trouble areas

iv) the explicit evaluation process can provide the developer with a set of questions that can relatively quickly differentiate appropriate applications from unsuitable ones

v) the process of using the check-list provides a large amount of information useful in requirements analysis and the specification for design and prototyping purposes

vi) the check-list approach is sufficiently flexible to be customised around different individuals and socio-political circumstances

Based upon these reasons and the apparent unsuitability of alternative approaches, it was decided in principle to make use of a check-list for problem selection. However, given the range of check-lists and alternative uses, it was important to define explicitly the purpose, context and conditions for their use within the company.

6.4. Project Organisation and Orientation

A criticism of current approaches to problem selection is that they provide indicators of what are likely to provide technically feasible ES projects, but they overlook how such guide-lines should be used within the organisation. A practical requirement of addressing such 'process factors' is that problem selection tasks are undertaken within an appropriate organisational framework context. In the client company this meant identifying an appropriate function which was best suited to evaluate and manage problem selection; and identifying an appropriate approach by which to introduce ES capabilities and potential into the organisation. Both issues are important in the context of technology transfer discussed in further detail in *Chapter 8.* Here, they relate specifically to the process of problem selection.

6.4.1. Organisational Structuring for Problem Selection

The section looks at the numerous strategies for introducing expert systems and the subsequent arrangements for managing problem selection.

a) External and Internal Strategies for Problem Selection: There are a diverse number of strategies by which ES may be introduced into the organisation reflecting different levels of commitment. Furthermore, each level has particular organisational effects according the implementation focus chosen, which may be externally dependent, such as relations between a division and centralised functions in a corporate organisation for example, or internal. External links have been well documented; for instance, Liebowitz identifies five levels of entry of ES in a corporate organisation, ranging from the merger or acquisition of companies already involved in ES to the development of corporate ES capabilities for use by divisions. However, internal to a particular organisation, the strategies and approaches are diverse and remain unknown and poorly covered (d'Agapayeff & Hughes:1989). *Table 6.*2 below lists a number of possible internal and external strategies which are available to an organisation. Clearly in the case of the client organisation, the external strategy was to make use of external academic knowledge: this also reflects the company's orientation towards ES as a investigative exercise prior to any company wide commitment or relationship with other external organisations.

Implementation Strategy	**Degree of Commitment**	**ES Overheads**	**Organisational Impact**	**Transfer Focus**
Develop a central corporate ES capability	*High*	*High*	*Medium*	*External*
Affiliate with ES firm or university	*Low*	*Low*	*Low*	*External*
Merge or acquire a company involved in ES	*High*	*High*	*High*	*External*
Develop in-house appreciation but rely upon sub-contracting	*Low*	*Medium*	*Low*	*External*
Develop central in-house company ES capabilities	*Medium*	*Medium*	*Low*	*Internal*
Build and distribute ES knowledge & capabilities throughout the company	*Medium*	*High*	*High*	*Internal*

Table 6.2.: Organisational Commitment & Strategies for the Introduction of ES

In formulating an internal strategy, the priority was to develop a central in-house capability orientated towards technology assessment rather than development. Implicit in this objective are a number of requirements:-

i) that the company does not become reliant upon the affiliated company, in this case Cranfield,

ii) that evaluative structures and frameworks are provided by use by the computer department rather than any external specialist body,

iii) that control for ES development should be retained by the computer department and that existing company communication networks and service relationships should be used to exploit any possible ES potential,

iv) that by distributing ES development skills company wide, it would take longer to identify and develop ES applications because resources were limited and uncoordinated.

b) Organisation and Structure for Problem Selection: *Figure 6.3* shows the project participants and set-up for ES selection and evaluation. The co-ordinating role was played by myself as ES adviser, mediating between other participants, and with the responsibility of measuring business and organisational needs against technology capabilities and potential. In defining the set-up, it was necessary to distinguish between external and internal participants to the project organisation. External participants included sponsors, university and market contacts. The sponsors included the company's technical director who added legitimacy and organisational credibility and also provided some indication of strategic priorities and needs. Support at this level was essential because the study was cross-functional and would impact the whole organisation. Further operational and political support was provided by the company's two computing managers one of whom pioneered the project and was directly responsible for its outcome. During the period within the client organisation, there was reasonably close liaison with the university. This was an advisory relationship in which ideas and technical support were exchanged. Other technical information, especially on products and services was gained through close contact with vendors and suppliers in the ES market and was consolidated through survey work, described in *Chapter 8.*

In dealing with company sponsors, it is essential to ensure that management provide a period of fairly low expectations so that time is available to become familiar with ES tools and techniques with the 'privilege' of making false starts, mistakes and errors. Two priorities in this are lowering management's expectations of ES and creating an environment where personal learning could be treated as an organisational process.

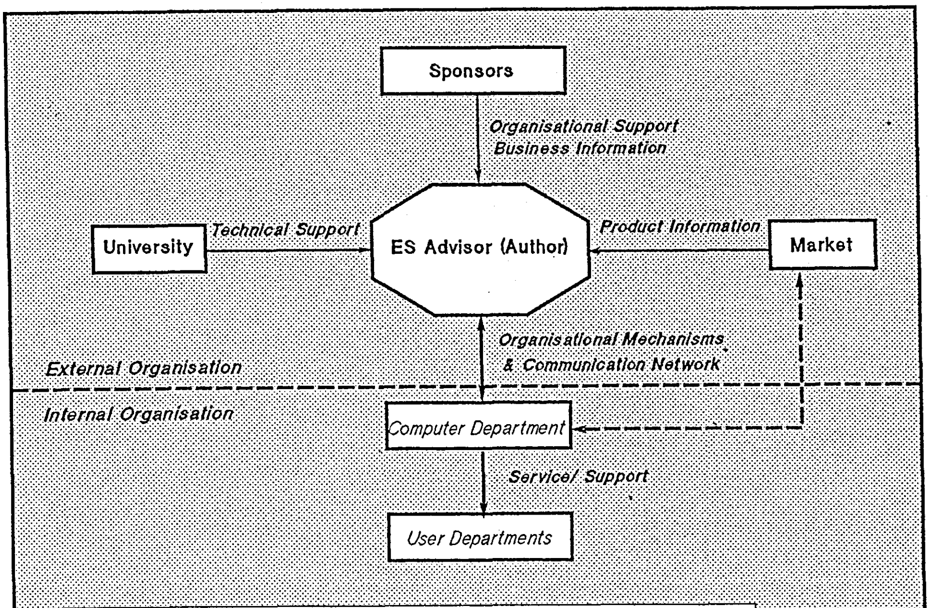


Figure 6.3: Project Participants & Set-up for ES Selection & Evaluation

Where the external organisation is concerned with providing technical, organisational and political support, the internal organisation is concerned with the means by which to undertake the project. This may be defined in terms of establishing an operational platform within the company to carry out evaluation and assessment and have access to company resources and people. Having established an external framework, the outstanding requirement in defining internal structure is in deciding which organisational function is best suited to perform problem identification This decision centred on two factors: first, who technically was suited to the tasks ? And second, who in political and organisational terms could successfully undertake such a company wide investigation ? To distribute effort into pockets of expertise in various areas of the company and have autonomous and possibly competing groups was considered unproductive and would also diminish their value at the organisation level. It was therefore decided that problem identification and future ES developments should be undertaken by a central co-ordinated function with the co-operation and involvement of all company functions.

6.4.1.1 Computer Department Structure and Role:

The choice of the computing department as the vehicle for problem identification was influenced in part by its current role in the company as a specific case, and the fact that the project champion was head of Information Technology (IT), but also by the logic that as another computing software, the IT function was best served to evaluate ES. *Figure 6.3.* shows the relationship between the author and the computer department and its relationship to user departments. There were a number of clear benefits in being associated with the computer department:-

a) Established Service/Support Role: The existing role of the computer department, as Diagram A123 in Appendix IV shows, was very much as a cost service to the company's contract based functions. It was considered appropriate that this functional relationship should be retained for the evaluation of ES, but with a increased emphasis upon support, awareness and self-help during problem identification. This enabled the use of established communication networks within and between the company's sites and an understanding of computing requirements demanded by each function (This understanding was greatly enhanced by the IDEFo study). There was also access to departmental resources (such as software, hardware, secretary etc). However, in promoting an association with the computer department, it was important to stress that the scale of ES activities would be significantly smaller than the combined resources of the computer department-in this case knowledge of expertise was restricted to one person prior to training and inductions.

b) Congenital Status: By being associated with the computing department, user departments tended to view expert systems less as an alternative to computing and more as an additional service or tool offered by the computer department. This was politically important because of the centralised role of computing and the fact that end-user computing (in which user departments manage their own computing needs) was strongly discouraged. In this case, there are also good reasons why control during early stages of ES development at least should be retained by the computer department based on previous attempts at developing an ES in the company by engineering functions:-

i) The project had been uncoordinated ill-advised and unsupported,

ii) The project was technology-driven with no apparent business or organisational value,

iii) The project failed to develop beyond a small demonstrator,

iv) Knowledge of the technology, its strengths and limitations, was localised to four senior engineers, two of whom had left the company.

v) The temptation to view shells, especially in the light of some vendors' marketing efforts, as an alternative to computer programming, induced some user departments to 'experiment' in ES with no justification or short-term motives.

c) **Organisational Expertise:** The computer department has substantial expertise and a departmental structure orientated towards the evaluation and exploitation of conventional information technology. This ranged from formal links and experience in dealing with software and hardware vendors to a technical understanding of current information technologies and future requirements, both of which are valuable in the assessment of ES technologies.

It should be pointed out that these factors are relevant to the client organisation only and the validity of using the computing department for ES problem selection for all organisations is less certain. Beerel suggests that responsibility for managing ES problem selection should not lie in the hands of the computer department for the reason that applications will be chosen which serve their specific needs. This though, reflects a poor evaluative framework rather than the inherent unsuitability of this function carry out evaluation. By contrast, Lu and Guimaraes (1989) argue that problem selection should be driven by the first organisational function to adopt ES techniques, but that for greatest success this should be the computing department or equivalent function.

A more sensible and universal approach may be derived from the work of Earl(1989) whose work refers to IT development generally but is applied here specifically in terms of expert systems. Earl argues that there is no general 'best way' for structuring the IT department. Rather that every organisation should decide which the most effective organisational function and structure is for ES problem selection in this case, contingent upon a number of situational factors and arrangements. Figure 6.4. shows there are four independent variables which generally influence these designs. These are:-

i) **Organisation characteristics:** these include corporate culture, organisational structure and management control systems.

ii) **The potential Strategic Impact of ES**: these identify significant business activities and the potential impact of ES using positioning business frameworks (such as Mc Farlan's grid) described in Appendix III.

iii) **IT Heritage:** this describes the existing inherited IT structure as a basis for understanding future requirements.

iv) **ES Technology Assimilation:** this refers to the stage of adoption and managing the introduction of ES technology.

Depending upon the state of these variables, Earl identifies five possible organisational arrangements: a centralised IT function reporting to corporate management; a business unit where IT is unified but set up as a business within a business and primarily serving the host business; a business venture where the IT function is a business unit set up to serve external as well as internal clients; a decentralised IT function where IT is distributed to each business unit and is under its own control; and finally a federal relationship whereby decentralised IT units exist alongside a centralised IT unit which is responsible for overall policy and architecture.

Of these, the development situation for ES in the client organisation most closely approximated a federal arrangement. It was important to retain control and define policy, particularly during initial phases of technology assessment and problem selection for a number of reasons:-

i) to ensure a framework and standard evaluation of candidate applications in line with business and organisational needs as well as specific functional needs,

ii) to create a central service and thus provide a correct orientation and imbue the right motives for using ES technology,

iii) to co-ordinate efforts in ES development and provide a centre for in-house expertise,

iv) to provide a strategic and company wide context for the use of ES,

v) to filter market and vendor ideas and promote the use of the technology in the company in a non-deterministic and disciplined way.

Moreover, as the authors time in the company was limited, it was important to establish a 'critical mass' of expertise in the computer department in preparation for my departure. It was therefore necessary to keep in regular contact and pass on acquired knowledge to selected IT personnel (in this case a project manager and systems analyst). However, it was recognised that such a centralised arrangement would be less appropriate at later stages of development when greater user participation and the transfer of ownership was important. A federal arrangement allowed a decentralised ES function to co-exist with the centralised function. This was particularly important with scientific and engineering functions who already had specialist computing expertise in certain areas.

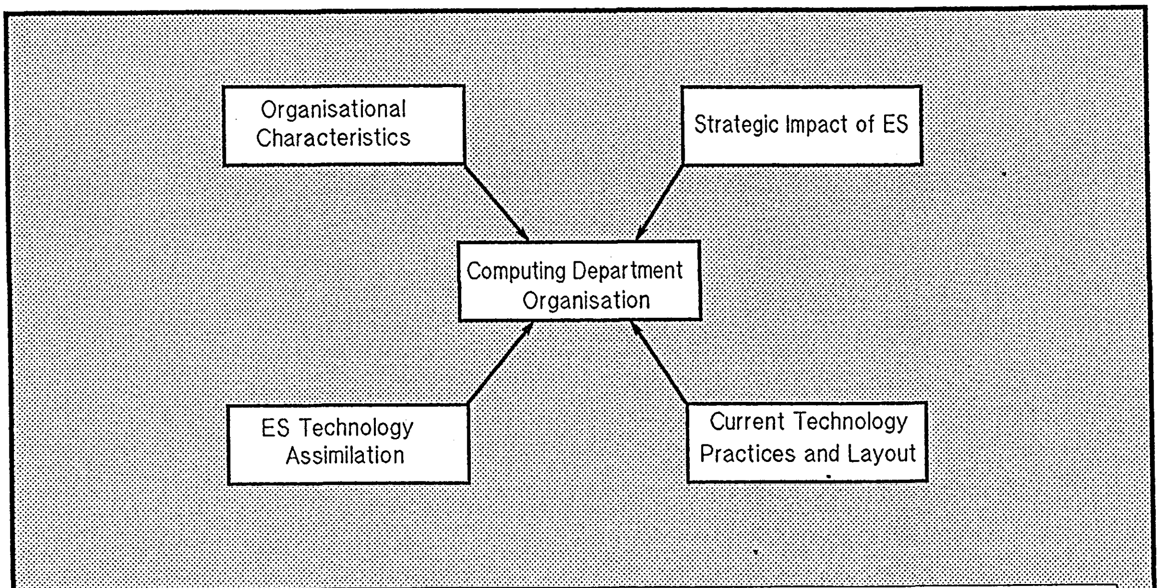


Figure 6.4: A Contingency Model of Expert Systems Organisation (based on Earl (89); Fig7.1.)

6.4.2. Invoking awareness and orientation

Early attempts at identifying possible areas for ES development showed that although a check-list approach may be worthwhile in principle, it's use would only be viable when users understood the significance of the questions and the reasons why they were being asked. This meant in practice that a programme of presentations and introductions was necessary. which described the purpose of the project as well as the capabilities of the technology.

An important issue which would later bear upon the types of applications identified was to whom should the presentations be directed. Beerel notes that by aiming problem identification at technologists when attempting to promote the use of ES in the company, it is likely that applications are suggested which are highly technical in nature .It was thus judged that a more balanced set of ideas would evolve if a programme of presentations was directed at managerial functions rather than specialist engineering and production functions. A second reason for aiming at management rather than technologists or prospective developers moreover, is that management involvement is considered critical to the success of ES in an organisation (d'Agapeyeff & Hughes: 1989).

Although presentations differed in content, their structure was broadly similar. Earlier phases were concerned with 'selling' the potential of the new technology through defining ES, the types of problem they can solve, how they solve problems and why they differs from conventional programming methods. Because there were large differences in the level of computing literacy and different information requirements between senior managers, middle managers and engineers, additional information was provided as an executive summary or in a more detailed format similar in depth to Appendix I.

The latter half of the presentation was concerned with regulating management expectations of the technology. During this phase, it was often necessary to re-focus management perceptions about the capabilities and limitations of ES where they may have had misconceived ideas about their use. Central to this process of awareness and education was to impart a constructive and realistic environment in which management would later suggest application ideas. Pedersen (1989) notes the danger that, " both management and the user of the expert system may tend to view expert systems as something different and unique- something of a cross between a person and a computer program." It was thus necessary to promote the technology to user departments as another computing service rather than herald it as something revolutionary and describe proven applications that were in use rather than possibilities. Such a re-orientation was especially useful for engineering management many of whom were already conversant with expert systems principles and had approached vendors for technical information. For these people, it was also necessary to emphasize the significant development resource and time commitments required to produce even a moderately sized expert system.

6.5. Generating Ideas for Potential Applications

Having provided an appropriate environment and infrastructure in which to consider the potential for expert systems, a next step was to generate and classify ES application ideas. During this process, there was an accent placed on information collection without too much concern towards how these ideas could be developed in practice. One advantage of separating ideas generation from feasibility and suitability assessments in this way, is that it avoids early commitment to a possibly inappropriate technical solution. Application ideas were generated from sources which were both external and internal to the company, as *Figure 6.5.* indicates .

6.5.1. Ideas from External Sources

Product information and application details were obtained from vendors, specialist ES consultants, and through a review of technical literature reflecting the development experiences of other manufacturing companies and academic institutions. This information was channelled through the author, so that it could be processed and presented to the company in an appropriate way. Without such an intermediary role between vendor and prospective user, there was a danger of applications being developed and driven for technical reasons. It also ensured that all information on expert systems was processed by the person most suited to its evaluation.

A main use for external sources of application ideas was as summary sheets. These were used during the presentations as examples and as a stimulus for managers to begin to identify their own applications. As *Table 6.3* shows using Quality Assurance as an example, the summary sheets defined a number of proven applications specific to a company function. Some of the suggestions presented in this way were often considered seriously for development, but more frequently were used as case studies. Where there was a strong interest to develop a system however, the focus of the proceeding evaluation was towards meeting organisational objectives and business needs since the technical feasibility of the proposal had already been established with great elegance and persuasiveness by vendors or sponsors of the application.

Expert Systems in Quality Assurance: 5. Summary of Potential Applications:
January 17th 1989: 10.00am Conference Room B

The following examples suggest ways in which expert systems might be used in Quality Assurance. They are intended as a stimulus for discussion rather than as solutions.

a) Printed Circuit Board (PCB) Defect Analysis: Presently Quality Assurance (QA) undertake trend analysis on the basis of a 'Snag Report'. This report however fails to describe the symptoms, cause and effect of defects. An expert system could help to formalise the reporting procedure (as a friendly database) and also help the analyst to identify trends and implications of defects. It could also be used by members of the PCB unit to diagnose faults in the first instance.

b) Archiving: The QA department is relatively small and yet quality as an issue is a company wide responsibility. Expert systems may be used to store current regulations, procedures and also the experience of applying specific company rules and diffuse this knowledge throughout the company.

c) Codes of Practice: The Test and Inspection specifications used by QA are difficult to understand and lengthy. An expert system may be used to interpret specifications and provide advice on the selection and application of appropriate codes.

d) Quality Control: Expert systems have a number of possible roles in quality control: to assist the controller in selecting suitable sample test sizes and test procedures; and fault identification in assembly and materials control. A further use might be in the selection of goods from suppliers based upon a number of legislative and evaluative criteria and also the monitoring of quality from the chosen suppliers over a fixed duration. An expert system would enable the use of more qualitative criteria to be applied such as reliability and delivery performance; a knowledge of which can only be attained through experience. Such analyses might also help to reduce the need to expedite (particularly with casting suppliers).

e) Training: In order to improve awareness of quality, expert systems could be used for training, especially with the imminent introduction of the new British Standards on quality assurance. Rather than simply convey information, some types of expert tutorial systems are interactive and are able to assess the user's needs and thereby present information in the most appropriate format. Where desirable, such systems may also be used to monitor or assess the user's understanding of a particular subject.

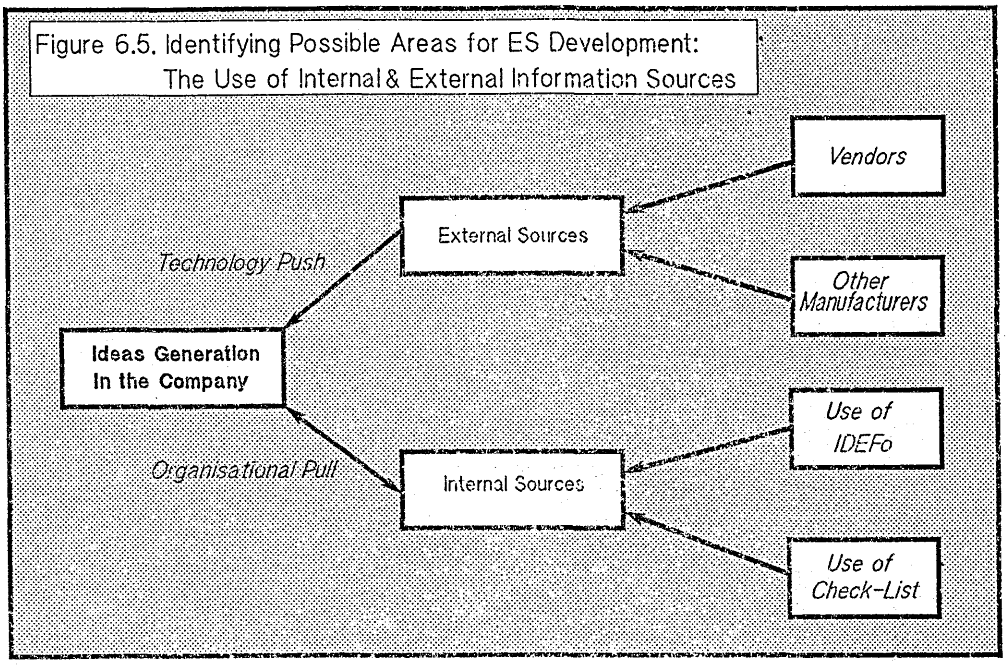
f) Data Base Analysis: Accessing information from the company databases and deriving 'meaningful ' conclusions from the screens is a further area where expert systems may be of use, particularly where the data base is used only occasionally. Using expert system techniques such as *heuristic searching*, it is possible to quickly identify relevant information from which to perform analysis or use the system to present the information in a more intelligible way. This is of particular relevance to QA who process and analyse a substantial number of fault and defect records on the VAX

(*Peter Holden, Computing Department Ex. 2073)*

Table 6.3. *Summary Sheet for Quality Assurance*

6.5.2. Ideas From Internal Sources

A second major source of ideas was generated from within the company: as *Figure 6.5* shows, these were informally through the use of IDEFo and more formally through the use of an identification check-list. Because internal suggestions were generated with a bias towards the character of the problem rather than the technology, the nature of the application ideas reflected a greater awareness of company needs and an increased likelihood of them being organisationally driven.



6.5.2.1. *Ideas Generated by a 'Technology-Fitting' Check-list*

The purpose of the check-list was to provide an aid to managers and functional heads, with a limited understanding of expert systems, to identify for themselves possible areas where ESs may be applied to their area of the organisation. It is not intended as a rigourous or definitive method of evaluation, but rather as a guide-line and as a basis for further discussions.

The check-list is made of a coarse set of identification criteria that include some high level judgements on the advantages of using ES. These factors require qualitative responses and may vary in scope to address a particular personal activity or concern the functioning of the department or organisation. The check-list is used to identify possible problem characteristics where expert systems have been successfully applied in the past. By not defining ES technology explicitly in the check-list, the assessment is not restricted to this technology alone but may also consider other technical and non-technical solutions.

The check-list is divided in three sections, as Appendix 5A indicates. *Section A* looks at problems based upon the distribution and organisation of expertise in a manager's area of responsibility. *Section B* looks in particular at the structure of decision-making; and *Section C* addresses information characteristics and requirements as a means of identifying potential ES applications. Although the concept of a check-list for identification is not new, as *Section 6.3.* has shown and indeed parts of the check-list are a synthesis of this work, its context of use and implementation is different for the following reasons:-

i) Distinguishes Identification from Evaluation: Of the check-lists reviewed in Section 6.3, most fail to distinguish in the structure and wording of the check-list between the identification of problems appropriate for ES and their evaluation for development. Many, such as those of Liebowitz, Prerau, Beerel and Keller combine the two elements within the same check-list. In the approach adopted in the company, problem identification is distinguished from application selection through the design and use of separate check-lists because the organisational priorities, processes and participants in each case are different.

ii) Defines a process of investigation: The check-list is described within a wider process of identification which includes a programme of technology transfer and business assessment. Hayes-Roth and Liebowitz for instance, suggest that the check-list may be used instead of these processes. Excepting this view, a criticism of all check-list based approaches to problem selection is that they fail define how the check-list is to be used and by whom. The company check-list was used in two ways. First, for senior management, the check-list formed the basis of a 'semi-structured' interview since more information was often acquired in this way. Furthermore, it was considered more creditable to conduct interviews rather than adopt a check-list approach at this level. At a second level, the check-list was used as a questionnaire and issued to middle and lower management. Since it was to be used without consultation, it was important to stress beforehand the context of its use and the reasoning behind the questions (*an explanation for each question is given in Appendix V.*).

iii) Non-development bias: the questions are worded and structured for use by managers rather than developers. The emphasis upon managers identifying their own applications had a number of advantages. Firstly, it provides a measure of the interest and commitment by user departments through their response to the presentations and check-list. Secondly, it maintained the service/user orientation of the computing department by allowing user departments to respond in their own time and in their own way. A disadvantage however is that application suggestions were concentrated in areas of the company where there was the greatest interest rather than necessarily being of most value to the company. This required that the use of the check-list was placed within a wider business context made possible through the use of IDEFo as the next section describes.

The check-list was designed to be relevant to all areas of the business, but in practice management of engineering and manufacturing functions (operational management functions) generally found it easier to complete than commercial, services and senior management. Perhaps this was because the decision-making needs and knowledge requirements at this lower level were more tangible and more easily expressed than the intuitive logic applied by senior and commercial management.

The check-list was also quite limited in the kinds of applications it was able to identify. The emphasis was upon addressing current organisational problems and allowing managers to suggest areas where ES may be used to enhance an operation. Thus the check-list was likely to produce application suggestions which 'added value' to existing organisational functions and practices rather than produce new organisational activities which might change, for instance, the competitive balance of the company.

6.5.2.2. Using IDEFo during Ideas Generation

As an approach, the managers' check-list was applied informally, fairly unstructured and non-analytical. Some of the questions were poorly received by managers and others caused some political commotion which leads to the conclusion that in highly structured organisational environments particularly, more formal methods may be required for problem selection.

The potential use of IDEFo as a formal method for identification arises at the lower, more detailed levels of the model. Here it is possible to systematically identify areas where expert systems are useful according to the information characteristics of an activity and the nature of information inputs, outputs and constraints about the activity or between other activities. For example, an ES might be valuable when the information input is not completely defined (in which case, the IDEFo model might show an iterative cycle between information generating activities for example); if it is missing(this may be shown as a constraint upon an activity); or if the information is not accurate (this may be shown by an information referral back loop for verification). The IDEFo model is also useful in showing activities constrained by the availability of an expert, or the 'bottle-neck' distribution of expertise as a mechanism amongst many different activities. IDEFo may also be of assistance in identifying activities that have similar inputs and outputs and therefore can be used to specify activities which are functionally similar. This enables duplicated decision-making processes to be identified. Thus, through careful analysis and interpretation of the model, it is possible to locate promising areas for ES development. The interpretation of the model in this way is similar in concept to the Information Resource Mapping approach of Harbridge in that organisational functions are defined in information processing terms from which certain information characteristics which denote ES potential are systematically identified.

Although this approach is elegant in principle, it was considered impractical to model the whole organisation to this level of detail. Furthermore, this technique could only identify a small set of problems pertinent to expert systems which is equivalent in scope to those issues covered in Part C of the Managers' Check-list (see Appendix V). However, as the check-list shows, the potential for using ES extends to wider issues of expertise and decision-making beyond the information level to all levels of the organisation. The value of IDEFo then should not be viewed in formal terms, but rather in an informal capacity, as a means of enhancing the use of the check-list. This was possible for two reasons: -

a) IDEFo as a 'front-end' to the Check-list

The process of using the check-list received company wide attention and demanded the commitment and participation of a significant number of managers. In many departments, such commitment was not forthcoming because of other work priorities and also for political reasons, such as that it was difficult for some departments to justify participating in an investigative exercise sponsored by the computing department for example. A further concern by senior manager was that the combined use of the check-list with the presentations would raise managers expectations too much and suggest the imminent development of expert systems when in fact the ES programme was more concerned with evaluation that development.

For these reasons, it was necessary to present a more 'low key' method of evaluation in many of the company's functions. IDEFo provided the opportunity to identify a wide range of organisational problems without specific reference to expert systems or any other information technology. From the list of problems identified, the check-list could then be applied independently without management participation, in order to identify problems which offered potential for ES development. Used in this way, IDEFo provided an organisational 'front' and a politically acceptable means of using the check-list.

b) Providing a context for the analysis of check-list problems

When the check-list was used as intended, it was successful in generating a large number of problem ideas and suggestions. However, by its very nature, the check-list was essentially a 'technology-fitting' exercise, in that problems were identified which were significant in ES terms but not necessarily significant in terms of their value to the company, or in relation to other organisational problems. Having identified a collection of problems therefore, a second process of analysis was necessary which

placed a business and organisational context upon these problems: here IDEFo was of use in the following ways:-
i) In Identifying Causal Relationships: The check-list gives an individual account of a problem situation, often specific to a particular function or department. These problems are therefore 'local' but may have causal relationships with other problems at a lower or high level, in the same way that a decision tree suggests a hierarchy of decision-making. In identifying causal linkages, it is possible to move from symptoms of a problem expressed at one level to a more fundamental level where the true cause or 'root problem' is originated. Greater benefits are gained by addressing root problems because improvements in both efficiency and effectiveness are attainable.

ii) In Defining Cross-functional Problems: IDEFo as a hierarchical model, was of use in identifying not just root problems, but also the relationship between problems and their interactions between functions and across levels. It also helped to define the boundaries of problems and by being activity based, identified problems which were cross-functional.

iii) In Attributing an Organisational Significance to Problems: A problem will have different weights attached to it according to the level at which it occurs in the company and the people concerned. In that the check-list was used by managers, problems were identified and implicitly rated according to the individual's perception of the problem relative to the needs of the department or function for which the manager was responsible. Therefore the more junior the manager the more provincial were the problems at an organisational level but just as important to the individuals concerned. By distinguishing between different levels of organisational impacts, ideas generated from the check-list could be classified and evaluated relative to other ideas in the same class. IDEFo provided the means by which to make this classification in business terms. The next section describes how some of the business tools described in Appendix 3 were used as a basis for this classification.

6.6. Documenting Application Suggestions- An 'Applications Portfolio'

Having generated a plethora of application ideas, the next stage was to omit repeated problems and remove clearly unattainable suggestions from which a document was produced outlining the purpose of each suggestion and defining organisational roles and impacts. The 'Applications Portfolio' so produced , and shown in *Appendix VI*, provided a formal source of information about expert systems technology potential which was specific to the organisation and reflected the company's needs and situation at the time. The document is divided into two sections:

6.6.1. Applications Listings: This section gives a brief account of application suggestions. Each is classified by function (as defined by the IDEFo model), although there was a group of application suggestions, such as skills archiving, which were relevant to all functions of the organisation. Since ideas continued to flow into the computer department after the check-list exercise, new applications were suggested each month so that the portfolio was kept up-to-date.

6.6.2 Applications Classification: In order to provide further insight into the suggestions listed above and also provide a high-level means of assessment, this section gives an indication of project length, resource needs, costs and impact for each suggestion.

i) Project length: this is expressed in terms of development man years of effort and is taken to include the whole development life-cycle up to implementation of the operational system. This was found to be the most difficult estimate to make in terms of human commitment and quantifying costs, and has proved to be the most erroneous in other projects (d'Agepeyeff and Hughes: 1989). There are rough guides to estimating project length (see Hayes-Roth *et al* as an example), and a number of 'rules-of-thumb' are used such as "...allowing a month for every 100 rule of programming". Estimates made in the Application Portfolio were intuitive rather than analytical and as such, more of a guide relative to other applications. However useful these methods are in providing a relative measure for evaluation, they are woefully inadequate as a planning specification because they generalise across application domains (design tasks for instance are more complex than diagnostics tasks), and take no account of the diversity of tool and resource requirements (for instance, the programming productivity of a shell may greater than AI tools, but AI tools are more functional).

There appear to be two main reasons why no structured and formal methods exist for estimating the project length of ES projects. Firstly, that there is not enough experience in developing ES to provide definitive guide-lines (MI:1989). Secondly, the development life-cycle is at present considered too unstructured to allow formal estimates to be made (Jenkins:1987). However there are other reasons which only become apparent by addressing the problem from different perspectives other than purely technical. The significant focus given to organisational issues in this chapter stem from the belief that project estimates can be made more predictable by thoroughly understanding the problem domain and carefully and systematically managing the selection process so that appropriate ES solutions are chosen. Thus, although it may not be possible to predict the duration of all tasks that make up ES development, at least the uncertainty over feasibility and development suitability will be minimised therefore, it is hoped, improving the predictability of time estimates.

ii) Resource Needs: This provides a relative indication of resource requirements prior to more detailed feasibility studies. Four key criteria were chosen:-

a) Hardware Requirements: this refers specifically to the delivery system (the hardware on which the completed expert system operates) rather than development hardware. The hardware alternatives and criteria for selection are based on the results and work associated with the vendor survey described in detail in Chapter 8. Four classes of hardware are defined: Personal Computers (PCs); Workstations (WS); dedicated artificial intelligence Workstations (AIWS); and mainframes (MF). More information about each class is given in Appendix I.

b) Software Requirements: this refers to the primary software used as distinct from coding which may be carried out in order to interface or integrate the proposed expert system with other software. Four categories of software are identified (again from chapter 8, and discussed in detail in Appendix I). These include expert system Shells (S), Toolkits(T), Languages(L) and Environments(E). There is also a fifth class of ES software which is intended for a specific application and comes complete with a knowledge base of the application domain. This category has been termed Application Specific (AS) software.

c) Integration. Integration commitments are divided into three levels: stand-alone systems where the ES software operates independently and in isolation from other systems; embedded or linked systems where the ES software drives or is driven by other software; and fully integrated systems where there is open communications and interaction between different software systems.

d) Expert Involvement: It was difficult to quantify the level of commitment required by the expert or specialist because at this stage, the depth and scope of knowledge in proposed systems was uncertain. At best, all that could be provided was an estimate of the degree of expert system participation necessary to acquire the required amount of knowledge for the system to function. Clearly the more that information could be acquired from secondary sources such as codes of practice, reports etc., the fewer the demands placed upon the expert. The Portfolio used a rating system ranging form no expert involvement (rating 0) to intensive involvement by an expert (rating 5). The ratings were based upon intensity of involvement rather than total number of hours to account for the occasions where there was more than one expert.

e) Costs: The greater the reliance upon prototyping as a design technique, the greater the uncertainty over the costs of the system, because the number of iterations required to produce an acceptable solution will not be known beforehand. Chapter four described a 'hybrid' approach in which importance was attached to managing and controlling the prototyping process and using it alongside conventional software planning and specification techniques. Despite this restraint upon prototyping, difficulties in quantifying costs remain (Born: 1988).

As with expert involvement, the problem of quantification was overcome in part by attributing a relative cost estimate to an application suggestion based upon the above resource needs and expert commitment. The rating used was from 0, indicating a low cost to a rating of 5 signifying high costs. This was successful in identifying costs which were unacceptably high: however, the cost value alone is insufficient in justifying ES applications because it has a different meaning according to the purpose and need for the system in the organisation. For example, a system may be proposed which has significant strategic and long-term implications and yet may yield a high cost value. Conversely, a system developed as a demonstrator will yield a very high cost value with no visible economic benefit; yet it is an important part of the company's learning and development process.

*f) Measuring Business Impact*s

To provide some indication of the organisational effects of application suggestions and in part to overcome the limitations of a cost value alone, business information acquired from IDEFo (such as critical success factors and value-chains described in the last chapter), enabled some measure of business impact to be made for each suggestion. To add significance to this information a business model, what Earl (1989) generically calls a 'positioning framework' in Appendix III, was developed.

This framework is outlined in *Figure 6.6.* It has been influenced from Hammer *et al.*'s distinction between organisational value and business impact in the planning of information technologies (1987). The effect is to provide a model which compares the current business value of an activity against the improved organisational contribution of developing an expert system in the domain. As Figure 6.6 shows, the value of current activities to the organisation span from being low (operating as 'system' activities using Porter's nomenclature); to high which signifies the company's critical success factors (Rockart *et al.* 1984) and primary business or value-chain activities. The positive organisational impacts (perceived) of developing an ES has been divided loosely into three categories; by improving efficiency at a personal or task level; or by improving effectiveness at a functional level. Organisational value is increased according to these authors in two ways: by making better use of the existing situation through improvements in efficiency (increased productivity for example) or through improvements in effectiveness(for instance through better systems management). A third way is to 'add-value' by creating a new organisational activity based upon the innovative used of technology or through distinctiveness in the operations of the company itself. As *Figure 6.6.* shows, the former are more concerned with the internal operations of the company whilst the latter adopts an external business focus.

An assessment of business impact begins by understanding the business significance of the organisational task in which the proposed ES project resides. From the IDEFo model it is possible to define whether this task is a primary or system business activity and whether it is critical to the success of the company. From this, four basic regions of business and organisational importance are defined using a derived classification adapted from the work of McFarlan(1984) and Ward(1988). In the 'system' quadrant, the current business significance of a function is low and the proposed expert system is designed to improve personal or task efficiency. This may have the subsequent effect of improving the effectiveness of a wider set of activities which are dependent upon this task. Using the application suggestions for Quality Assurance outlined in *Table 6.3*, an example from this quadrant is an ES which provides advice on the selection and application of appropriate codes of practice.

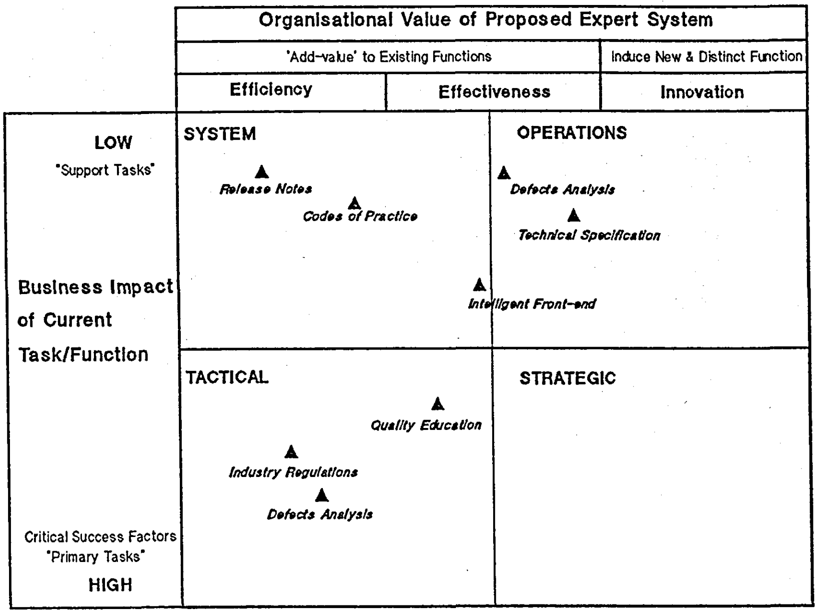


Figure 6.6. A Framework to Define the Business & Organisational Impact of Expert Systems

In the 'operations quadrant shown in *Figure 6.6.*, the business importance of the domain remains low, but the nature of the proposed system in terms of organisational value is different. Proposed applications may improve the effectiveness of whole functions rather than tasks alone, or induce new organisational functions through innovation. In Quality Assurance, a typical example from this quadrant is a proposal to develop an ES to analyse faults in the manufacture of printed circuit boards. In identifying an innovative use for ES technology, it became apparent that a new organisational function would be necessary in order to collect and record defects in the first instance. In this case, the organisational changes were more significant than the technical ones.

The 'tactical' quadrant was so named because improvements in the efficiency or effectiveness of discrete and bounded functions, identified as being of high business value, may improve the effectiveness of the whole organisation. Thus, by providing the effective means of identifying a company's critical success factors for instance, modest improvements in the efficiency of a task may provide significant business returns. Useful examples from this quadrant, again using the Quality Assurance domain, include proposals for an ES interfaced to a database recording customer reported defects in order to identify trends and possible generic design faults; and a system which defines and helps recipients to apply new standard regulations (BS5750) to quality control. Here, the technology innovation was considered more important and complex than the organisational effects. Applications which are included in this quadrant are important in sustaining existing business.

In the fourth quadrant of the framework, 'strategic' applications are positioned. These applications have the potential to generate a new organisational function which is of high business value to the company. This may be achieved in two ways: by improving organisational effectiveness (internal focus), by enhancing internal competitive advantage for instance as Porter & Millar suggest; or generate a competitive advantage through innovation (external focus) by changing the way in which the company interacts with suppliers and customers and other external sources. By the nature of Quality Assurance as a function in the company, no strategic applications were identified.

6.7. An Analysis of Applications from the Portfolio

The existence of a formal document indicating resource requirements and business impacts provided a unique opportunity to analyse the types of problems which are addressed by ES technology and their organisational significance. Furthermore, it was hoped to investigate whether the different processes of identification influenced the particular types of problems being identified. A full graphical analysis of the findings gained from the Applications Portfolio is covered in Appendix VII. The main conclusions drawn from this investigation are discussed below.

6.7.1. An Analysis of Support/System Application Suggestions

Figure A. in Appendix VII shows that nearly half (46%) of all applications identified were support or 'System' expert systems. In other words, most were intended and perceived as supporting personal roles or unit efficiency rather than as being of more important value at an organisational and business level. Clearly individuals in the company will have different organisational roles and therefore a senior manager will be of more business value to the company that a junior manager. However, the point is that managers associated the use of the technology with specific tasks and individuals rather than for general or integrated use. Furthermore, the use of the technology was targeted at the short-term needs of the manager and his or her area of responsibility.

A breakdown of System applications in Figure B of Appendix VII shows that despite some differences, there is a reasonably even distribution of applications across organisational functions. This suggests that for system ES, no organisational activity is particularly more receptive or innovative than the others. The range of applications suggestions per function was from the lowest value of 12% for front-end services (which included sales, marketing senior management and commercial functions) to the highest value of 28% for Operations functions (which included manufacturing, purchasing and planning). A possible explanation for this variance is that operations and engineering functions are computing intensive and make substantial use of current company information technology facilities; consequently they were more able to see the potential for ES than commercial functions who tended to be 'occasional users' of company computing systems. Figure B also shows a privation of System applications in 'pre-shop' functions: this reflects the complexity of pre-shop activities such as planning and scheduling and the relative unsuitability of System type ES to perform these tasks.

Figure C in Appendix VII, provides a decomposition of System ES by development requirements. A typical application of this class is likely to be developed and delivered on a personal computer; using shell-based software in preference to languages or toolkits. Furthermore, the application is likely to take no more than a year to develop and be of comparatively low cost, with 72% of all System applications having a cost rating of between 1 and 2 (1 indicating low cost, 5 high). It is significant however, that despite the relatively small physical resource commitment, the personal commitment of the expert to project development was considered high, with 46% of applications indicating a rating of between 4-5 (1 indicating low expert commitment, 5 high).

6.7.2. An Analysis of 'Operations' Application Suggestions

A breakdown of Operations ES by function is given in Figure D., and shows that the greatest number of applications were located in services and support functions identified by the IDEFo study, with computing services in particular being the subject of a number of suggestions. This focus upon service and support functions reflects that the effectiveness of an operation, as perceived by managers, may be improved by addressing those indirect business activities which sustain or add-value to current operations; what Porter and Millar would call support activities as distinct from primary business activities. This viewpoint is reinforced by the fact that many of the company's services, like computing, are shown as a controlling input (and therefore a constraint) upon business activities and functions.

Figure E shows that Operation ES require further integration (either by being linked, embedded or fully integrated) with conventional information systems and databases. The poor functionality of shells is mirrored by the increased use of languages and toolkits such as prolog and poplog in order to satisfy these increased integration needs. The effect of using languages and the necessity of integration is that there is a subsequent increase in hardware needs, with a greater use of workstations and Mainframe systems; an increase in costs (most applications in this class had a cost rating over 3); and an increase in the average duration of development.

The expected use of Operations expert systems differed from System types: from encoding experts' rules of thumb, for diagnostics and configuration for example, which requiring intensive expert involvement, to moderate expert involvement with a trend towards ES used for data interpretation and information analysis.

6.7.3. An Analysis of 'Tactical' Application Suggestions

The progression to Tactical ES refers to the way in which certain ES applications were perceived by managers as having a significant impact at the organisational level. It is not unreasonable to assume, as Figure F in Appendix VII affirms, that ideas generated by a senior manager will be of greater organisational importance, but also that this individual is more likely to think in organisational (cross-functional) terms. The concentration of Tactical applications upon front-end services, as *Figure F* shows, reflects the centralisation of senior management in these functions. It also suggests the extent to which changes in these commercial functions have the capacity to impact the whole organisation. By contrast, *Figure-F* shows that service and support functions (accounting for less than 4% of application suggestions in this class) are unlikely to affect organisational effectiveness.

Figure G shows a breakdown of tactical application suggestions according to development requirements. Other than a slight increase in integration levels, requirements are broadly similar to System type requirements. This highlights that although the technological complexity may be the same, the organisational and business impacts can differ significantly according to the positioning and settings of the ES in the company.

6.7.4. An Analysis of 'Strategic' Application Suggestions

A feature of Strategic applications, as earlier sections have identified, is that they improve business effectiveness externally by redressing the competitive balance between customers, rivals and suppliers (see Porter & Millar's 'Five Forces Model'(1985).) or internally by identifying critical business success factors and evaluating the way in which potential ES applications affect the company's value chain. Figure H demonstrates the importance of front-end services in dealing with external (market) factors and it is here where over 80% of strategic applications are located. A second implication from Figure H, is the importance of engineering activities in the company as being critical in retaining business effectiveness such that innovations (technical or organisational) brought about by the use of ES technology are likely to have strategic implications. As a large tendering and contract based organisation, the effectiveness of company operations is bound by the speed and quality of design in response to tender requests by clients.

As with tactical ES, a distinguishing feature of strategic expert systems is how and where they are used in the organisation rather than necessarily the scope and complexity of the technology *per se*. Indeed, *Figure I* shows that almost 80% of the proposed applications could be developed on a personal computer. However, there was also a band of applications, approximately 20% of all suggestions in this class which demanded significant resource and organisational commitments: each took well in excess of five man years to complete at relatively high costs; required full integration with company Mainframe computer systems; and require complex environment-based software to be feasible.

6.7.5. An Evaluation of Identification Techniques

A stated objective in the analysis of applications was to investigate whether from the internal and external methods of ideas generation summarised in *Figure 6.5*, each method favoured a particular type of problem or organisational situation. Of the relatively small number of strategic applications identified, all were observable through IDEFo rather than the check-list. Similarly, many of the tactical applications were identified from this source. IDEFo allowed root problems rather than symptoms to be identified which arose at higher levels in the organisation, often across functions and company-wide.

The focus of the check-list by contrast was at a personal or task level and was therefore most effective in identifying support and operations type expert systems. The failure of the check-list to identify tactical and strategic applications highlights its limitations as essentially a 'technology fitting' approach. More positively, it shows the clear role of organisational studies like IDEFo to complement incremental and technology orientated problem identification approaches.

Application suggestions derived from external sources proved technically to be the most complex. Those from ES software vendors were usually integrated systems aimed for manufacturing and shop-floor applications. Subsequently, most were support and operations type ES. Ideas elicited from other manufacturing companies were also predominantly operations expert systems although there was a more general spread of applications relevant to a broad range of company functions.

6.7.6. Enhancing the Positioning Framework

In applying the positioning framework to over 150 application suggestions, a number of anomalies were identified. For instance some applications exhibited the characteristics of a 'System' ES and yet had the potential to provide greater than expected business returns. In addition, a number of trends were observed such as that many strategic applications would be considered as being extreme in terms of organisational innovation but yet were of only moderate business value. However, there was a minimum perceived business value allowed for an application to be considered 'strategic'. To take account of these and other discrepancies identified through an analysis of applications, a more fluid classification evolved and is presented in *Figure 6.7.* below.

The use of this revised framework allowed a number of generalisations to be made about the organisational commitment to expert systems. In that 'system' application proposals were concerned mainly with improving task efficiency, the organisational commitment to the technology was likely to be localised to a particular individual or a specific task as *Figure 6.7.* identifies. Where an application improved effectiveness, the organisational focus shifted from a personal level to a functional level, with a number of company activities and whole functions being affected by the change. When a business significance is added to this formula, applications which focus upon improving task efficiency will nevertheless have wider implications at a functional and even organisational level. The capacity to innovate or improve effectiveness in areas of business importance moreover will not only be of company wide importance, but it may change the external market structure in some way by changing the rules of competition for example. The classification therefore provides a measure of the technology diffusion and scope of its use and effects in the company.

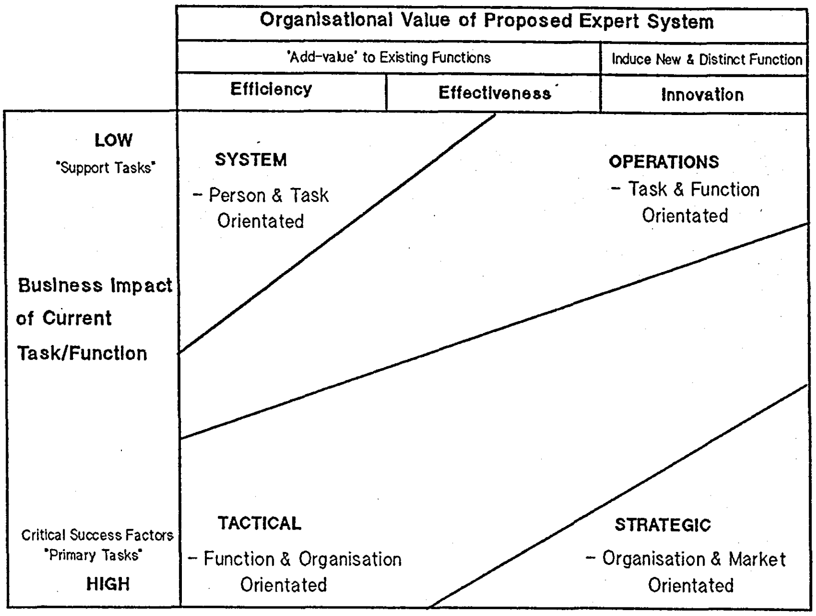


Figure 6.7: A Revised Framework Defining Regions of Business Impact & Organisational Value

The framework in *Figure 6.7.* may also be used to provide an indication of the resources necessary for the development of a proposed application and the development approach taken. In the Strategic quadrant for example, the planning of the application by definition has to be integrated with business planning and is strategic as well as long-term. The two are likely to be interdependent; as ES becomes embedded in business operations and pervasive in business thinking, ES take on more complex organisational forms. Unlike System applications which are well bounded and adopt neat functional responsibilities and authority, more dispersed organisational structures are planned or emerge. Thus where the development approach to System applications can be 'ad hoc' to the extent that the computer department can react to users' problems adopting an eclectic approach to the utilization of ES tools, Strategic applications require a detailed pre-planned development policy which is concerned more with organisational structure, systems integration, compatibility and manageability issues.

6.8 From Ideas to Application Selection.

The analysis so far has been concerned with providing an organisational framework which will allow the company to identify and classify different types of problems and thus define areas for potential ES development. Where this process was characterised by an open exchange of ideas within the company, and from external sources also, the next phase looks towards successively reducing the large number of application ideas down to a more manageable number of candidate ES projects which are more realistic given certain technical and organisational constraints. This next phase also looks more closely at the means of reduction by addressing the feasibility of proposals and how appropriate they are for development within the company. Preceding consideration of technical feasibility and development suitability however, are constraints arising from the fact that the technology was being used for the first time in the company. This factor imposed certain restrictions on the choice of application based upon resource limitations and organisational commitment specified by the company. There were also other motives which constrained the selection of a first project which related to the effectiveness by which it would be able to be promote, educate and diffuse knowledge of the technology through the company.

6.8.1. Identifying 'First Application' Constraints

The importance of a first application for a new technology like expert systems is that the effect of a success or a failure is significant, mainly due to the habit of reaching a conclusion about the technology from the one experience (Beerel:1987). As indicated above, there are particular requirements of the first application which limit the number of viable projects, and therefore provide a useful first 'filter' for the selection process. These constraints reflect the extent to which the company is committed to the use of ES and the risks it is prepared to make; the subsequent time and resources it allocates to the project; and the technology transfer, educational and awareness needs.

A number of attempts have been made to specify the useful attributes of a 'first project'. IBM (1989) for instance specify certain critical success factors for a first project such as: the application should not require a lot of database access; the application should not use a lot of extended routines; the expected development time for the first prototype should be reasonable (between six months and 1 year). d'Agepayeff & Hawkins(1987) suggests that 'simple' rule-book, fault-diagnosis or procedural type applications are appropriate where development takes place in a non-critical business area. Applying these ideas and others, in addition to the specific constraints laid down by the company, the following selection criteria were applied: -

a) That the application suggestion should reside in a company function whose business value was of support or systems status,
b) That the perceived organisational impact would therefore be to add-value to an existing function rather than be innovative,
c) That the suggestion be developed in one year or less,
d) That the system can be delivered on a Personal Computer,
e) That the software used was of relatively low cost and could be learned and applied within a short period of time (no previous experience of A.I. programming is assumed),
f) That the system should not require a lot of data-base access or the use of external routines,
g) That the project was not considered costly and set-up costs were low,
h) That the chosen experts would not required to be committed full time to the project,
j) That the application domain was of general relevance company-wide and subsequently of high exposure (high diffusion and visibility),
k) That the benefits of the system are tangible,
l) That the initial risks are considered low,
m) That there was some potential for following on the project should it prove successful

The result of applying these criteria was that the application set was very quickly reduced to ten candidate applications which are summarised in Table 6.4. A more detailed account of each application is given in Appendix 8A. It should be observed that not all of the candidate applications satisfied all of the first project criteria completely. For instance, Application_1 in Appendix 8A., an expert system to configure client spares orders, had a very high cost rating because of the difficulty in acquiring the source knowledge. However, in this case, such was the interest in this proposal by the company, that these costs were acceptable. Therefore the use of these constraints was in attempting to provide a complete picture of an application, rather than considering each criterion to the exclusion of the others.

Application No. Constraint	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_10
Can be constructed in less than one year	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
Relates to 'system' or support functions	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
Can be developed on an ES shell or application software	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Can be delivered on a Personal Computer	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Is the system Stand-alone ?	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
What is the Expert Commitment	5	1	3	4	3	4	3/4	2	2/3	2-4
What is the Cost Rating ?	4	1	2	2	1	2	1	4	2	1
Is the application of 'Universal Relevance'?	YES *Basic Config. Model*	NO *Clarificat'n Model*	YES *Basic Diagnostic Model*	YES *Basic Design Model*	YES *Basic Planning Model*	YES Basic Config. *Model*	YES *Basic Diagnostic Model*	YES *Data Interpret'n Model*	YES *Basic Clarificat'n Model*	YES *Basic Selection Model*
What is the exposure / diffusion of the application ?	LOW	LOW	LOW	LOW	LOW	MEDIUM	HIGH	LOW	MEDIUM	LOW
Is there potential for a 'follow-on'?	✓	✗	✓	✓	✗	✓	✓	✓	✓	✓

Table 6.4: A Short-List of Candidate Applications

Having defined a set of applications which broadly met first project constraints, the next phase was to undertake a more detailed evaluation which would assess the technical feasibility of developing an ES for the problem domains identified .

6.8.2. Assessing the Feasibility of Application Suggestions

Despite a reduction to ten applications, a detailed evaluation of each would be difficult and time intensive. A second filter was therefore applied to consider basic issues of feasibility. This filter was present in the form of a check-list of what was considered to be fundamental criteria necessary for an application to be considered technically feasible. The detail and depth in which each application was analysed against these criteria was dependent upon the relative size and complexity of each; however the *discipline* of using them provided a common structure for describing and comparing these applications. As with the manager's check-list, it was not a definitive listing, in fact a number of revisions and additions were made as the feasibility check-list was used.

The feasibility check-list was divided into five sets of questions as *Appendix 8B.* shows. These include questions on: aspects of the problem (how a problem is bounded for instance); validation of expertise (e.g defining the components of expertise and the ease of knowledge elicitation); user implications (e.g. identifying user needs and possible impacts); development prospects (e.g anticipating costs and maintenance requirements); and justification (e.g. what are the intangible benefits). Many of the questions were qualitative and required an understanding of ES theory in order to understand the significance and interpret questions correctly. Furthermore, a number of the questions used in the check-list are to be found in current literature on the selection of applications (see Waterman:1986; Prerau:1985; Liebowitz:1989; and Badiru:1988). The check-list combines what were considered to be the most useful items from the literature. It differs from these approaches however by the way in which the check-list was used- not as a 'self-help' guide but as part of a wider feasibility assessment intended to be managed by the computing department in future, but with the implicit participation of potential users and experts, and the incorporation of an organisational viewpoint through the use of IDEFo.

During the first phase of feasibility assessment, the check-list was used exclusively by the author, although computer department personnel observed its use in practice and were 'talked-through' the decisions and actions taken during consultations. Rather than ask questions directly, the check-list was used more as a prompt and questions were chosen selectively according to whom was being interviewed. Unlike the manager's check-list, interviews were held on an informal basis with potential experts and users as well as functional managers. Interviews could take up to an hour, but frequently the feasibility of a candidate application could be judged from one viewpoint fairly quickly. A final decision was made by combining information and judgements from different viewpoints in the hope that a more balanced and thorough evaluation was possible. In broadening exposure to the proposed system in this way, there were potential conflicts and political problems arising from concern over the role of the technology in the department. In order to remove this threat, the participation of relevant experts and users was planned early on in the assessment and preceded by a informative meeting by the functional manager concerned. Where potential conflicts remained, despite such planning, this was indication itself that the proposal was inappropriate. Indeed the 'political factor' was perhaps one of the most important criterion in the selection process.

As a result of this analysis, a general picture of feasibility for each of the ten candidate applications was gained and could be summarised; as Appendix 8C shows, into factors relating to the relative strengths and limitations of the proposal. No attempt was made at rating these factors because many could not be quantified and moreover, weightings would differ according to the emphasis placed on them by each participant. Instead it was left to the discretion of the author in rationalising each proposal and from the list of candidate applications, the number was reduced down to three 'feasible' projects (the basis for this selection is again discussed in *Appendix 8C*). These are:-

a) A capital investment appraisal adviser,
b) A maintenance adviser for a Flexible Manufacturing Cell or 'FMC'
c) A fault trouble-shooter for computer systems hardware.

6.8.3. Defining Problem and Development Contexts

From issues of feasibility, whether it is possible to develop an expert system, the next consideration (presuming that the three applications chosen are feasible) is how appropriate they are for development in the organisation and what the detailed development requirements will be? It was proposed that a second development suitability check-list be used for this purpose. However in order that it could be used effectively, a more substantive and detailed understanding of the problem situation and the development context for each of the applications was necessary. The process of analysing the problem would help to define the requirements of each application more accurately. Both IDEFo and Multiple Perspective Concepts provide a useful basis in gaining this appreciation and this was supplemented by the development of evaluation prototypes. This allowed the number of proposals to be systematically reduced until a final selection was made. Each step of this 'filtering' process is described next, although in order to retain clarity much of the detail of specific project proposals is enlarged upon in Appendix 9.

6.8.3.1. Using Linstone's Multiple Perspective Concepts

Multiple Perspective Concepts were useful in two ways: first to define 'what' the problem was by analysing the settings that make up the problem (see Chapter *3.5.1* for a reminder of these); and second, to understand 'how' the problem was being viewed by investigating the development context for each of the proposals. The latter proved useful because it animated the reasons why managers and experts thought ES proposals would be useful and revealed individual and organisational expectations and requirements from the system. A full analysis of the three candidate systems using MPC is given in Appendices 9a, 9b and 9c respectively. MPC was especially useful , as Appendix 9c shows, in highlighting the importance of political and cultural factors in reducing the likelihood of success of the capital investment appraisal adviser, which was subsequently rejected.

6.8.3.2. Using IDEFo to define the Problem Context

A danger in the use of check-list approaches irrespective of their design, is the tendency to focus at the level of the proposed system or problem rather than understand the cause of the problem and define it in organisational and business terms as well as in technical terms. The value of IDEFo was as a basis for 'organisational prototyping' in which it was possible to understand how each of the projects interfaced with current activities within the problem domain and the environment in which the problem was embedded. IDEFo provided a simple but complete description of the problem, the environment and the people involved. At a high level of abstraction, IDEFo defined the boundaries of the problem and through successive decompositions defined its activities and components. At lower levels, IDEFo was able to define equipment and machine interfaces together with information inputs, processes and outputs. In this case, both the IDEFo model and the process of undertaking the IDEFo study provided an insight into the problem situation and made it possible to define the problem more completely. IDEFo was of most use, as Appendix 9a and 9b show in particular, in outlining the physical boundaries and scope of trouble-shooting by providing an inventory of machines, equipment and plant, together with an indication of how they are used and by whom.

6.8.4. Building Evaluation Prototypes

Two evaluation prototypes were developed and are discussed in detail in Appendices 9a and 9b. The first was an off-line shell-based system for debugging and recommending maintenance action on faults arising in the company's Flexible Manufacturing Cell (FMC) to be used by maintenance engineers. The second prototype was a trouble-shooting aid for users of Mainframe terminals, network terminals and linked printers.

Discussions on the use of prototypes very much assume two factors: first a large systems perspective; and secondly, that a full scale system will evolve from the prototype (e.g. Harmon & King and Waterman). In this project, there was exception to both factors in that the scope of the project was restricted by 'first project' criteria to a small-scale, shell-based application; and that the objective of prototyping was to provide design information based upon a given design proposal rather than necessarily evolve a design concept. In this capacity, the prototypes were of value in defining four principle knowledge base and interface requirements (after Kahn and Bauer: 1989):-

a) **Scope and granularity of knowledge:** this is the knowledge required to solve a problem to a certain level of detail. The two prototypes provided information on how the expert generates a sequence of decision events and defines an explicit representation of order. They also verified the sufficiency of a rule-based programming environment in representing these decision-making processes.

b) The degree of procedural regularity in the use of knowledge - The FMC prototype underlined that the same diagnostic solution was not always achieved in the same way since there were alternative methods and techniques for different modes of failure that occurred. In the Computer Hardware (CH) prototype, diagnostic solutions were much more consistent and unique and therefore easier to program and verify.

c) The need and availability of run-time data: Both prototypes require substantial non-permanent knowledge, i.e. information that is acquired 'run-time' during the course of the consultation. For the FMC prototype, this is in the form of machine monitoring either automatically in which case the prototype is embedded within the control and monitoring equipment of the FMC; or manually whereby the maintenance engineer interprets machine sensor information. The user of the CH prototype requires greater interpersonal and communication skills in order to acquire the necessary run-time information required to make a decision. This places greater demands upon the design of the human-computer interface since unlike the FMC prototype, there are different end users with a range of skills and competences to consider.

d) The degree of accuracy acceptable in a situation assessment. The purpose of the CH prototype was to solve a broad-band of high-level user problems. In this case, it was acceptable, within certain limits, for the system to recommend likely causes of faults without necessarily considering and evaluating all possibilities because at this level the consequence of a recommendation not yielding a positive result was not critical. For the FMC prototype by contrast, it was important that a correct situation assessment was made and that the subsequent debugging advice was valid because of the criticality of the problem domain and the possibility that a mis-diagnosis may cause catastrophic damage to the FMC. Since there were no resident experts in the company, there was an additional responsibility upon the prototype that it performed this function effectively.

Although the purpose of the prototypes was to define functional requirements rather than explore solutions, the actual process of development changed the functional goals of the proposals. In the case of the CH prototype, its development led to the re-design of the nature of the interaction between the user and the system than originally intended. The development of the FMC prototype indicated that more fundamental changes were necessary than simply the design of the interface. These changes were organisational rather than technical and evolved from a shift in focus by management as greater familiarity with the technical possibilities and the particular limitations of expert systems led to a re-assessment of current human resources and capabilities. As the technical costs and complexity of operating a full scale version of the prototype became known, the decision was made to upgrade the skills of the maintenance engineers by investing more heavily in training and educational programmes so that they could assume an expert role. Thus from an initial understanding of the problem from a technical perspective, the apparent failures of a technical ES based solution led to a shift in viewpoint and subsequent calls for human and organisational changes which have since been implemented.

By contrast, the CH prototype reaffirmed rather than contradicted the initial conception of the problem, as defined by the MPC analysis, and thereby not only validated the design concept but vindicated an appropriate *process* of development. The prototype thus reinforced the information acquired from the feasibility and development check-lists and also provided additional design insights which have added to the formal design proposal described in the following chapter.

6.8.5. Applying Development Suitability Criteria

With the decision to construct a full scale computer hardware fault trouble-shooter, the next phase was to consolidate the information acquired through the above stages in order to specify design and development requirements as completely as possible without actually commencing construction of the system. Defining the scope and requirements of the project was facilitated by the use of a development suitability check-list described below. It was considered important that both the design and use of the check-list should be multi-dimensional so that human and organisational, as well as technical, design issues were considered.

6.8.5.1. The Design of a Development Check-list

It has proved difficult to compare the resource, time and effort estimates with other projects (MI:1989), for example through the use of statistics from which some form of parametric estimation can be made, because of the early state of application experience generally, and more specifically within the client company. The aim of suitability assessment of some form was to provide a bottom-up means of making such estimates more possible by systematically defining project requirements and of specifying the methods and processes of development. In the company, this process was formalised through the design of a development suitability check-list intended for use by the computer department and as a complement to earlier feasibility and prototype analyses.

The design of the check-list was motivated by Linstone's multiple perspective theories which state that in order to define a problem, it must be represented and understood from different settings (*what* are the components of the problem) and viewpoints (*how* is the problem being looked at ?). This provided a basic framework for the check-list into which a diverse set of technical, organisational and human criteria were incorporated, as Appendix 10a (an exhibit of the development suitability check-list) shows. Each criterion has associated with it a detailed description of what is required and how it should be carried out. Appendix 10b explains the motives and back-ground behind the use of the check-list. It also indicates the association of each criterion to each of the settings (*see Figure I., Appendix 10b*); and, where appropriate, discloses the documented sources of questions used in the check-list.

6.8.5.2. Using the Check-list

The check-list was not intended to be used directly by User departments as a 'self-help' guide, but rather designed to be applied by selected members of the computer department as part of a wider process of association between manager, user, expert and development representatives. In order to impress upon the company the context for its use, the check-list was applied by the author to the computer hardware trouble-shooting proposal. This along with earlier analyses formed the basis of a development proposal which is described in detail in the following chapter.

6.9 Conclusions to Chapter 6.

The purpose of this chapter has been to place problem identification and application selection within a wider context of human social and economic evaluation and with greater emphasis upon the *processes* involved in evaluation (how to use a check-list, how to manage assessments and so on) as well as simply describing in a linear fashion which tasks are involved. In applying this approach, it was essential to provide an organisational framework which could support the company and progress it from a position of no understanding of ES to one where it could begin to identify feasible and desirable ES-based solutions independently of the author. There were two basic problems in using this approach however:-

i) The company was expected to set-up an evaluation framework without seeing first the tangible benefits of an expert system. Some managers regarded this as an undue expense although there was general agreement that viable ES projects would not be identified if such a process was not undertaken.

ii) The framework stressed the importance of non-technical as well as technical issues in identification and selection stages. Some managers failed to acknowledge the importance of social and human factors in these stages, although it is significant that there was more resistance from lower management than middle to senior management.

The approach adopted in this chapter separates analysis of the problem from the evaluation of particular ES based applications. The former is independent of a solution and does not pre-define a particular technical option such as expert systems. The second stage by contrast, looks specifically at the feasibility and development suitability of applying ES based techniques to these problems. The response to this approach has been largely favourable, although there have been criticisms that some of the earlier phases particularly are regarded long-winded and difficult to justify for a single ES project. However there are two responses to this criticism:-

i) The business and organisational assessments are of value to the company irrespective of whether ESs are being developed or not.

ii) The focus in approach has been in constructing a process-orientated framework for ES evaluation which may be used continually. It is not restricted to a single project.

This made the choice of problem selection framework of great importance. Current approaches to problem selection combine initial problem identification with application selection. However, experiences in the client organisation show that these processes are distinct in terms of approach and requirements. By drawing attention to the organisational, political and human factors involved in exploiting ES technology in the company, four discrete stages may be identified in problem selection and these are: creating an organisational infrastructure for problem selection; generating an appropriate climate for ES development; identifying problems appropriate for ES technology; and selecting an appropriate application. The principal limitation of current approaches is that they only address the latter two phases and therefore fail to provide a business or organisational context during evaluation and limit discussions to technology-fitting, technical feasibility and task definition. Moreover, they omit issues of desirability and suitability and the means by which these tasks may be accomplished within a particular organisational setting. A further shortcoming of current approaches is that evaluation is comparative with respect to other candidate ES applications such that the selection of an ES project is self-fulfilling. More preferable is to consider alternative technical and non-technical solutions in the evaluation process. Again this requires wider organisational and needs assessments rather than one pertaining to a particular type of technology.

The dual process of top-down organisational assessment and bottom-up technology-fitting provided different types of ES potential. The use of the manager's top-level check-list for example tended to attract ideas which, although feasible in ES terms, were restricted to within functional areas and as such tended to serve the interests of the function rather than the business as a whole. Moreover, without a business or organisational context, priorities tended to be expressed from an individual viewpoint. The subsequent ES applications which were identified tended to be aimed at improving personal efficiency or unit efficiency in accordance with personal goals of the respondent. Clearly, there were exceptions: for instance senior personnel distinguished less between personal needs and organisational needs; engineers ('rational actors') often expressed needs in terms of new equipment and computer technology. Significantly, it was a strata of middle management who most frequently defined problems and needs in personal terms. Moreover, the potential benefits of using ES were frequently cited in political terms.

The value of a top-down strategy using IDEFo was that in addition to generating a large number of application ideas itself, it also provided a context by which to assess the organisational and business importance of ideas generated bottom-up. It is noteworthy that those application ideas identified by the IDEFo report tended to be cross-functional and often company wide and, moreover, their business significance was greater than those identified using a check-list. It is conceivable that by undertaking a combined approach to problem selection in which both bottom-up and top down strategies are adopted, areas may be identified which meet personal needs and expectations whilst also accomplishing business needs and are acceptable to the organisation.

The philosophy behind the use of the feasibility and development suitability check-lists was that although it is not possible to define all development needs completely, a great amount of design information may be acquired . This detail is of use in planning the development and implementation phases; setting time and resource estimates; and in providing a full justification for the project. It will also help to constrain prototyping and iterative design to within certain defined targets. The focus in prototyping was as a complement to the above check-lists. The prototypes were useful in testing the premises made in the check-lists, for example 'is there an expert committed to the project', and verifying that the assumptions made were correct in practice. The prototypes also generated additional design information which was necessary in making the final selection.

The strengths and weaknesses of the approach described in this chapter should be measured by the extent to which it is able to identify and select feasible and appropriate expert system based applications. The next chapter therefore describes the authors experiences in developing the chosen application, a computer hardware fault trouble-shooter.

Chapter 7

The Development of an Expert System for Computer Fault Trouble-shooting: Design and Operations.

7.1. Introduction

It is perhaps useful to reflect one what has been done so far. The approach used in former chapters may be subsumed within the label 'pre-project analysis' and represents a business and organisationally driven method of firstly identifying problems which are of value to the company in some way, irrespective of any particular technology . From this, measures were taken to identify those problems where expert systems technology might provide a solution using broad criteria of technical feasibility. More detailed methods were then applied to evaluate the development suitability of proposed ES applications using a set of multidimensional criteria. Following further investigations of development requirements through evaluation prototyping, a final application selection was made.

This process differs greatly from current approaches to ES development in the preeminence attached to pre-project issues. Although commentators (see Harmon & King: 1985 for example) acknowledge the importance of problem identification and application selection, Chapter 4 has shown that these activities are defined principally as 'technology-fitting' exercises which look at the feasibility of the match between the problem and the technology without addressing issues of business planning and organisational needs for example. Moreover, as well as failing to consider pre-project analysis seriously, current ES approaches fail to define the *means* of undertaking such processes. The way in which ES is introduced and managed in a company, as Chapter 8 will show, has important implications upon the relative success and outcome of a technology project, involving as it does, wider issues of technology transfer and the management of 'change'. The response to these needs has been the development of a framework for technology assessment in the client company. This provided the tasks for problem and technology evaluation in terms of a company specific programme of parallel assessments as well as the organisational mechanisms which enabled such analyses to take place.

The main output to the above activities is the selection and specification (as far as possible) of a particular expert system application which aims to diagnose and debug faults in the company's computer hardware. This chapter centres on the design, development and operational experiences in producing this system. As with all previous chapters, the author has endeavoured to describe all settings and account for each of the perspectives of a particular problem situation or decision-making process. The chapter begins therefore by specifying the problem in technical, organisational and human terms from which the system's objectives, together with design and project management requirements, are defined. Of the latter, an attempt has been made to apply conventional software engineering techniques such as time and resource estimating, project planning methods, and functional modelling of the proposed system's design and its desired features. Because of an absence of economic analyses in the justification of projects, particular emphasis is placed in this chapter on project costing and evaluation; benefits are defined in terms of cost savings and reductions as well as the added-value that the system contributes.

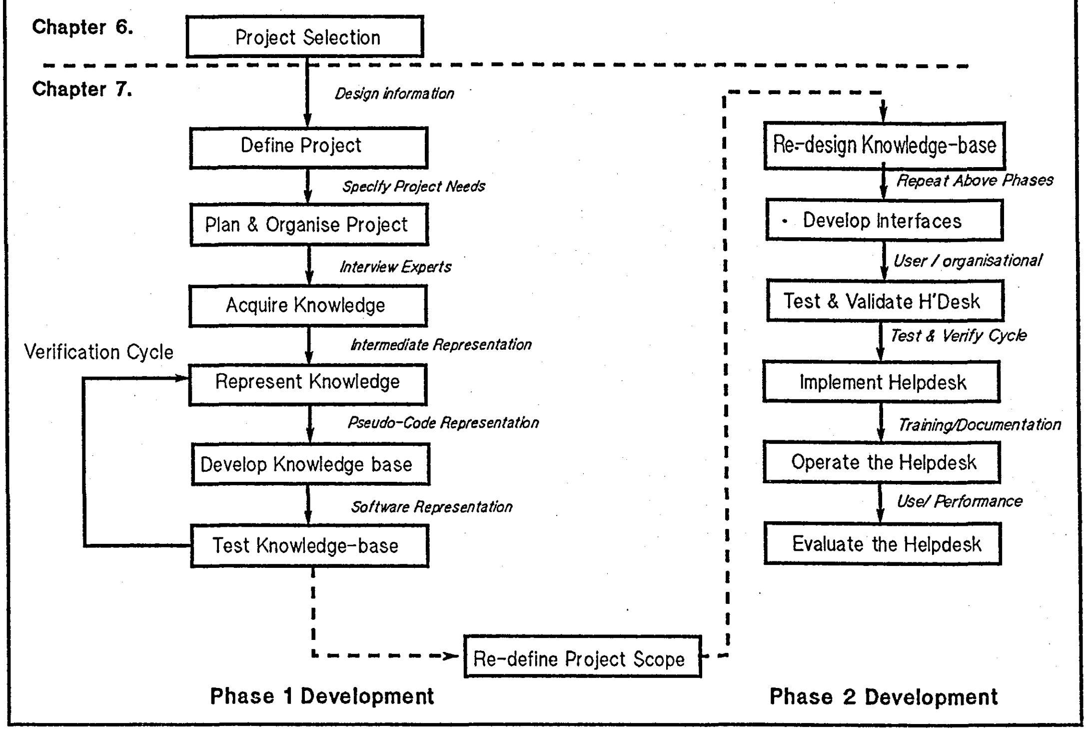
Earlier experiences in this domain during the development of an evaluation prototype helped in understanding the structure of expertise from which knowledge representation needs could be defined. This provided the basis for the selection of hardware and software for the project. The ES software chosen, its attributes and capabilities are described in the appendices.

Development itself begins with the process of acquiring knowledge from the expert and representing it in an appropriate form so that it may be understood and verified before being encoded in a suitable knowledge-based programming format. This process is collectively known as *knowledge acquisition*. Following a short survey of the various approaches to knowledge acquisition, this chapter identifies that many of its processes are similar in formalism to that of the IDEFo methodology. The use of IDEFo is described in terms of resolving factual and personal conflicts between experts; providing a common communications device for use by end-users and managers as well as between experts; and as a discipline in providing a structured approach to the documentation, updating and verification of knowledge. IDEFo was thus used as a first stage in providing an intermediate representation, that is a representation of experts' knowledge which is independent of any particular type of software tool or programming method.

A second phase makes use of knowledge mapping to represent the decision-making processes which characterise the expert's knowledge. Following verification of this intermediate representation by the experts, a final stage of representation, which was used by the author before actual programming, was 'pseudo coding'; this attempted to formulate the structure and programming requirements of the knowledge base without operating to the syntax constraints of a particular software tool. From this, an account of the experiences in developing and programming the system's knowledge base are described, with the bulk of programming examples and technical features being appended. This process is incremental and modular in that small areas of expertise were represented and validated manually from which pseudo code and finally formal code were produced. It was some way into the development of the knowledge-base in this fashion that it became evident that the original time estimates were ambitious. Moreover, a number of design features, particularly concerning the relationship between the system and end-users, emerged as being wholly impractical. From this realisation, two significant changes were made: firstly, that the scope of the project was reduced to what was considered achievable given the amount of time left to complete a system of some form and implement it; and secondly the organisational role for the expert system was simplified, and in doing so, the technical complexity of the system's user interface was rarefied. The structure and design of the revised system is described from which a more detailed account of the knowledge-base, inferencing and interfacing features of the system is given; again making use of appendices for some of the more technical details, such as rule and variable listings.

Following system's validation, testing and documentation, this chapter continues by describing the implementation of the helpdesk in the company and associated activities such as training and re-organisation. The remainder of the chapter is then devoted to two issues: operational experiences in *using* the helpdesk over a six month period-here the viewpoints of different individuals, such as managers, experts, end-users and developers, is considered important in attaining a balanced assessment of its role and use. Secondly, in providing a full·and similarly balanced evaluation of the helpdesk using a set of multidimensional criteria. A mapping of the chapter in full is given in *Figure 7.1*. It is clear from this figure that two distinct phases emerge in the development of the system reflecting a change in personal and company expectations of the technology and also a greater understanding of the technical difficulties and organisational constraints which acted upon the project during the processes of development. In order to reflects the experiences and insights which led to a transition between these phases, the structure of this chapter has been logically divided into two parts. **Part One** looks at all pre-development activities such as design and requirements specification. cost estimating and implementation planning. **Part Two** looks at the development, implementation and evaluation of the help-desk application and begins its analysis with the knowledge acquisition process.

Figure 7.1. Outline of Chapter 7: The Development Lifecycle of the Helpdesk



Part One: Consideration of Pre-Development Issues

The development of the computer fault trouble-shooter is principally a technical issues; indeed, the expert systems literature abounds with similar such analyses. However, earlier chapters have repeatedly shown that the validity of a design rests not only upon its technical excellence but upon the processes and mechanisms of development where the determinants of success are human and organisational factors. This chapter therefore begins by providing a four tier structure of user participation and human involvement with a view to keeping the system very close to users' needs during subsequent stages of development. This structure also provides a pretext for the inclusion of human and organisational factors during design and specification, such as anticipating legal and ethical effects of the system; of appreciating the effects of *not* doing so, as with reported concerns over acceptability and responsibility for instance; and finally, in using these factors as performance criteria in the evaluation of the system during operations.

7.2. Towards A User-Orientated Design and Development Process

Although this chapter is essentially a technical one in describing the development of the trouble-shooter, it is structured according to the practical attempts to incorporate a human viewpoint into the development process. This is not new to this approach however; indeed, the earlier stages of problem identification and application selection of ES applications, although not involving future end-users directly, is implicitly constrained by human factors (albeit from a managerial viewpoint). When it comes to the design through to implementation stages of development however, as authors such as Markus(1984), Mumford(1983) and Cooley(1989) argue in Chapter 3, it is imperative that a user viewpoint is represented explicitly. However, Bright & Stammers(1989) note that very often, the individual point of view is only referred to during the design of the user-interface as a process of validation. Furthermore, Ackermann *et al* (1989) notes that however sympathetic and idealised the developer is towards a human-centred approach, there are great difficulties in involving the users and other non-technical members of a development team in the processes of the design lifecycle. This is exacerbated by a focus upon individual tools- task analysis during knowledge elicitation, a human factors approach during design and so on- rather than in promoting general human concepts. Finally, Diaper(1988) notes the limitations of discussing the contribution of an individual in terms of the user-interface alone; firstly because of the variety of people involved in development- for the computer troubles-shooter for example, this includes the developer, expert, project manager, senior manager, system operator and as such 'user' should be taken to mean all these people- as well as the end-user. Secondly, an emphasis upon involvement at the interface stage precludes participation of the individual at all other stages of design and specification of the system.

Rather than focus upon specific tools, and limit the potential contribution of all types of people to narrowly defined fields of design, a more useful basis is suggested by Young(1989) who identifies four levels of human involvement spanning goal setting and functional design to participation in physical design. When this structure is applied to the client organisation, it helps provides a mapping, as Figure 7.2 shows by which appropriate forms of user involvement are expressed and may be incorporated into the development process. The four levels are:-

i) Contextual Level: these include the social and organisational factors that affect how the expert system should be integrated into the company as well as broader questions about the evaluation of the system; and the legality and ethical responsibility for the advice the system gives. As Figure 7.2 shows, contextual issues help in defining individual roles during the project and the function of the system in the organisation. Furthermore, it shows attention towards mechanisms and processes, such as appropriate project organisation and planning structures, which facilitate user involvement at later stages. The contextual level therefore provides the user terms of reference for the rest of development.

ii) Conceptual Level: at this level, human participation centres on defining the personal sets of tasks to be performed and task relevant concepts around which the system is organised. In terms of the project, Figure 7.2 shows that this is manifest through user participation during functional design: by focusing user involvement at a functional level rather than a physical level, there is likely to be more empathy between users, developers and experts. In achieving a common mental model, users are able to express requirements in 'logical' terms, specifying the knowledge the system has to be provide for example, which is also coherent to the developer.

iii) Communications: user involvement at this level is 'techno-personal', using Linstone's nomenclature used in Chapter 3, and focuses upon defining appropriate interactions between the operator-machine interface and between the helpdesk service and the user community. Figure 7.2. shows that the design issues covered at this level include creating appropriate syntax and dialogue features, and the design of appropriate error handling and explanation facilities for instance.

iv) Physical: as development approaches the physical level, such as with programming the knowledge base, it is expected that the scope for end-user involvement will be much more limited, formal and technical in context; with more prominent role for the systems developer. At this level, user involvement thus restricted mainly to dealing with the spatial layout and 'hard' characteristics of technical and organisational interfaces.

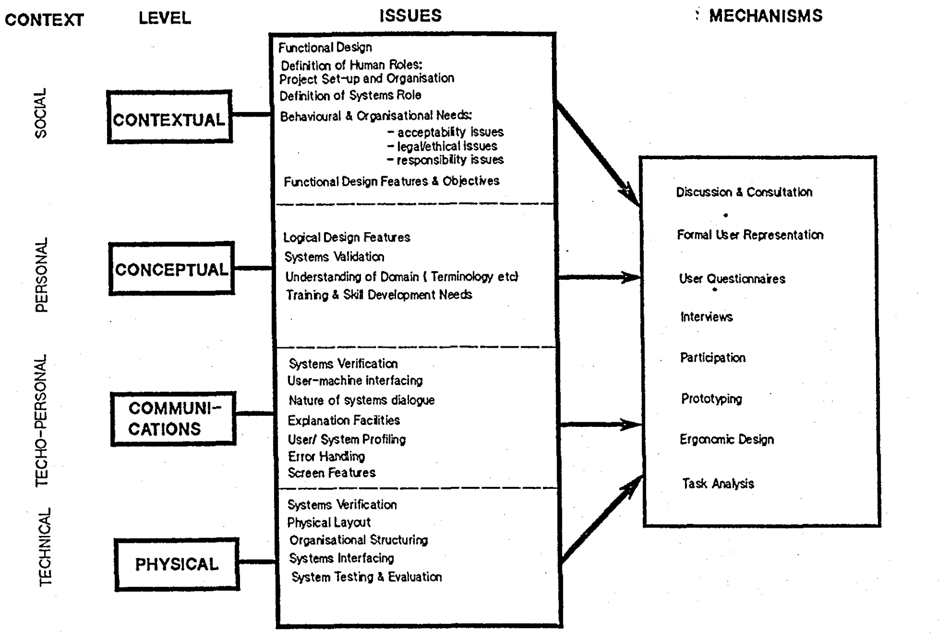


Figure 7.2: A Proposed Mapping of Levels of User Involvement Upon Development

The relationship implied by Figure 7.2 is not fixed however, and the form of involvement will vary according to the organisational and political settings of the company. Spinas and Ackermann (1989) for example describe how in a formal and highly structured organisation, the degree of participation is restricted to an exchange of information only between users and developers; whilst in a similar sized company users were involved in decision-making and design at all stages. In the former organisational setting, involvement could be encouraged by using formal methods of participation such as Mumford's ETHICS method described in Chapter 4., or analytical tools such as task analysis, questionnaires and structured interviewing. By contrast, in more functional settings informal methods of participation such as group discussions, prototyping and direct forms of participation such as user-driven design may be appropriate. Thus, at each level, the issues, mechanisms and objectives of human involvement are different, with each level being more relevant to certain phases of development. The remainder of this chapter describes how human involvement was facilitated at each of these levels during the development of the computer hardware trouble-shooter.

7.3 Project Definition and Terms of Reference

This section relates to the contextual level of Figure 7.2 and is interested in defining the role and objectives of the project; it's performance measurements; the roles and requirements of all individuals involved or affected by the proposal (including skills impact); and in defining an appropriate project organisation which will facilitate user involvement. This section also covers numerous factors such as equipment availability, gradings and calls per hour, which allow for the comparison of current services against the newly created terms of reference. They also provide definitions and measurement of the above objectives.

7.3.1. Objectives

The purpose of the project was to construct an expert system which would function as a 'helpdesk' for trouble-shooting computer hardware faults. The term helpdesk was important since it was hoped the system would invoke a better relationship between the computer department and the user community by providing a better computing *service*, aimed at reducing the downtime of computer systems in the company. Indeed, computer department management were eager to stress that the helpdesk would be a dedicated, user orientated facility in the sense that it would be operated full-time as a service to User departments. In this sense, the helpdesk provided a contact point for users and it was hoped that this attempt at accountability would improve the image of the computer department. Furthermore, the more freely users contacted the computer department when there was a problem, the less likely they were to attempt to solve the problem themselves or ask a colleague: such 'dabbling' often caused further problems and would be discouraged if an quick, alternative source of expertise was available elsewhere.

Within the computer department, the motivation for developing the helpdesk was that it would help to relieve the experts of routine diagnostic decision-making and thus allow them to undertake more specialised and complex development work. The term 'expert' here denotes senior personnel with specialised knowledge of a specific computer system. The expert system based helpdesk would be manned by a non-expert operator who would advise end-users on their problems over the telephone. The helpdesk was only intended to address a broad-band of high-level problems which could be rectified by users themselves, and where the solution was straightforward enough to be communicated effectively to the end-user. For more complex problems, those which take the expert longer than ten minutes for example, or where self-repair or rectification by the user was technically demanding or dangerous, the helpdesk operator would be expected to refer these problems to the expert directly. Here the role of the operator is as information gatherer, acquiring as much detail about the problem from the user as possible and placing a priority on the criticality of the problem before handing it over to the expert in an appropriate form.

7.3.2. Problem Analysis

In addition to understanding the problem by distinguishing between types of users, computing equipment and impacts, it is useful to attempt to define a generic set of problem classes (not all queries are equipment failures) in order to learn more about the types of service demanded of the helpdesk. Table 7.1 defines six basic classes of user query together with their concentration for each set group of computer equipment. The information on which Table 7.1 was based was acquired from an analysis of incident report forms which documented users' reported faults. This showed clearly that the nature of the query varied according to the computer equipment and also according to the different types of users that made use of this equipment- secretaries, engineers, managers, line-operators and so on.

From a user perspective, Table 7.1 shows that a high proportion of the calls made from IBM and Vax systems' users concentrate upon operating queries and special requests: very seldom is the source of a query due to equipment failure. This reflects the division between the management of these systems by computer specialists and their remote and controlled access by end-users using terminals. By contrast, for Personal Computers, the complete management and operations of the system is under the control of the user and subsequently, the bulk of user queries centre upon equipment and service failures since the user is responsible for more computer equipment. More generally, users of Personal Computers ask proportionately more queries than any other type of user because of the degree of autonomy that such stand-alone systems bring.

Table 7.1 only provides an indication of how the helpdesk should be designed because it reflects what users requested rather than what they actually wanted; the difference being that some users approached experts only when it was essential whilst others did so for what may appear as trivial reasons. Clearly the more approachable and helpful the expert, the more likely the user is to contact him for all types of queries. It is expected that for this reason, the helpdesk will actually stimulate a demand for less critical queries. This is valuable because it may prevent more critical problems from arising later on.

Table 7.1: User Query Types and Their Distribution According to Computer Equipment
(Based on a similar analysis by Fry:1989)

Query Class	**VAX**	**IBM**	**PCs**
1. Equipment Failure *e.g. faulty printer*	*	*	***
2. Service Failure *e.g. All equipment not working*	*	*	***
3. Lost Data *e.g. lost print-outs*	***	**	*
4. Special Requests *e.g. special user feature*	***	***	*
5. Operating Queries *e.g. meaning of screen message*	***	***	**
6. General Queries *Basic need to ask a question*	**	*	***

Key: * -Low; ** -Medium; * -High proportion of User queries**

The helpdesk should not aim to solve all user problems but address upto 70% of calls which are of a routine nature and may be solved in less than 4-5 minutes over the telephone. The performance of the help-desk should therefore be measured on how well the help-desk operator is able to decide whether a problem may be diagnosed using the help-desk, or whether it requires referral to the expert.

7.3.3. Problem Control

Many problems will require the user to carry out a simple operation (keyboard, power checks and so on) under the instruction of the help-desk operator. As the computer awareness and literacy of the user community increases, the complexity of the debugging tasks entrusted to users may also increase. However in the first instance, the more complex problems will require attendance by the expert himself. It is desirable on these occasions that a log is made of expert assistance requests with, when possible, a description of the problem so that the expert knows precisely the required action to take. The help-desk should also recognise that some problems are more critical to the company than others and it should therefore define criteria by which these priorities are to be identified. These include type of user, service level agreement of the software used, and problem area .

7.3.4. Help-desk Organisation & Layout

The intended organisation of the Help-desk is shown in *Figure 7.3.* User calls are received by the help-desk operator who then consults the Personal Computer (PC) based trouble-shooter for technical support. The operator may also refer to a network terminal (which is able to connect up to both IBM and VAX systems and provide data about the status of the network itself) for systems information. Depending on the complexity of the problem and the information provided by the end-user, the operator then decides whether a response can be made directly or whether the problem should be escalated to one of the four experts. In either case, a record of the fault is made.

End-users may require verification of the operator's information and organisational role: this is expected at the beginning, particularly, when certain users may have a preference to contacting one of the experts. It is important therefore that accountability and support is provided by an organisational reference and, as Figure 7.3. shows, this is provided by the project manager.

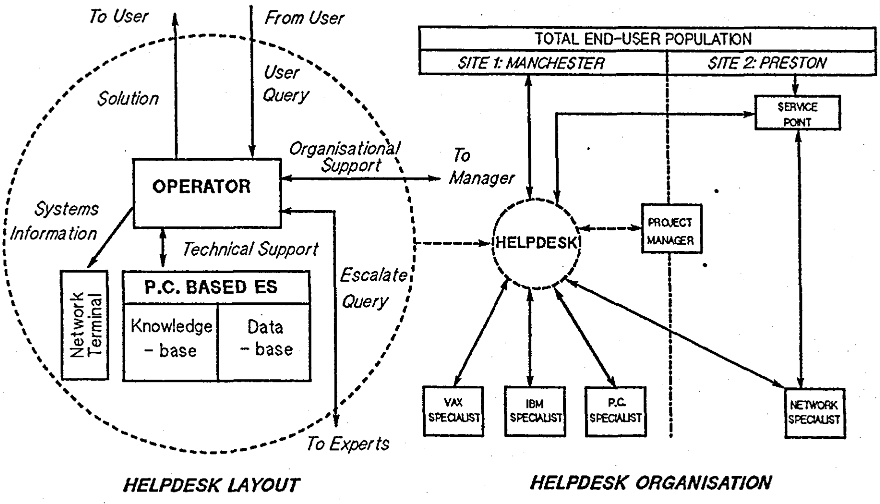


Figure 7.3: Proposed Layout and Organisation of the Computer Department Help-Desk

A structural problem faced by the help-desk was the division of users between the company's two manufacturing sites. This physical split also mirrored differences in end-user profile (for instance, the Preston site were generally more converse with using PCs than the Manchester site, but were less familiar with the IBM system), and more importantly, the help-desk had to accommodate differences in organisational practices and procedures and computing management style between sites. Although the ideal solution was to have two help-desks at each site, this was difficult to justify for the number of faults received daily and the estimated costs of manning two help-desks. Rather than have two help-desks, one at each site, is was decided that all calls should be referred to Manchester where the help-desk would be located. However, in order to retain the status of the computer department at Preston and maintain a service department role with users at this site, a service point was set up, as Figure 7.3. shows, which would channel all Preston calls, process them, and refer them to the Help-desk. The service point would be absorbed with the computer department's daily operations and therefore no extra personnel or facilities were required. Besides the organisational benefits, this arrangement ensured that all faults could be logged at a single, central source and enabled the status of faults to be monitored .There are two further issues associated with this structure :-

7.3.4.1. Help-desk Responsibility: accountability for decision-making rested with the project manager. This provided the operator with an organisational contact point and a definable responsibility level. It also ensured that the project manager checked that the operator understood completely the functioning of the help-desk and when and how to refer to the expert; that new faults were recorded in an appropriate manner; and finally, that the expert was satisfied by the performance of the help-desk and that it was being used.

7.3.4.2. Call Ownership: this refers to the status of calls escalated by the help-desk operator to an expert; or from one expert to another when the problem at issue crosses the boundaries of individual expertise. By attributing ownership to the user at all times, a help-desk call would then not close until it is confirmed that the user is satisfied.

7.3.5. Individual Roles and Responsibilities

As mentioned earlier, Diaper (1988) has argued that in order to enlarge the scope of human involvement in project development, it is necessary to re-define the term 'user' to include not only the end-user of the help-desk in this case, but also all other individual roles involved together with the nature of their interactions.

A consensus on these roles and interrelationships was achieved through open discussion and consultation at the contextual design stage. This is reflected in the organisation of the project and structure of the development team, as Figure 7.4. shows. Each member has different roles according to the phase of development: this may be defined formally as a responsibility chart (Badiru:1988) which specifies which member of the team is *responsible* for a particular activity; should be *informed* or *consulted*; should *approve* or simply *support* the activity. This is a useful approach since it defines explicitly the extent of operator, end-user and expert participation. Moreover, it provides a discipline for co-ordinating project members; this was especially important when users and expert representatives were distributed between the Manchester and Preston sites.

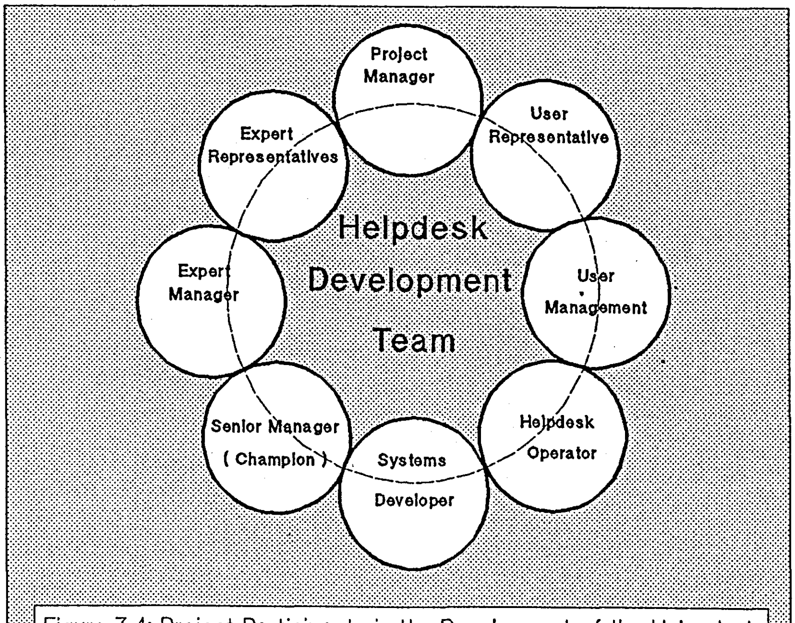


Figure 7.4: Project Participants in the Development of the Help-desk

The significance of including end-users' and experts' management in the development team, as Figure 7.4. shows, is in order to maintain an organisational viewpoint in determining the role and effects of the help-desk. For instance, while the end-user representative is important in defining personal needs and interface requirements, the end-user's manager, in charge of a large population of other end-users, was more able to rationalise the layout and implementation requirements of the system.

7.3.5.1. Help-desk Operator Role

The help-desk would be 'dedicated' using Fry's classification and be manned by a permanent staff member providing a full-time dedicated service to the user community. All calls would be processed via the operator who would retain responsibility for each call until the query had been resolved (from a user's point of view). The help-desk operator has three basic roles therefore: firstly, to elicit and interpret details of the query such as information about the user, the equipment used and the context and settings of the query itself. The operator will also be expected to assess the criticality of the problem. Secondly, based on the above details, the operator may be able to provide a direct and immediate response to the end-user's query using the help-desk. Thirdly, where the operator is unable to resolve the query, it is necessary to decide on which of the experts is most able to solve the problem and to escalate the query with accompanying details.

It was recognised that inter-personal skills (a good 'telephone manner') and a knowledge of the organisation were more important than an in-depth knowledge of the technology. For these reasons the computer department's administrator was earmarked for the role. She had worked in the computer department for many years and was respected and well known. Moreover, she was familiar with computer technology nomenclature and also, as a user, had experience in using both terminals and office systems. The operator was informed of the project and her possible role early on, which provided the opportunity to reflect ideas and suggestions to her during the design phase. It also enabled on-going testing of user interface features during prototyping. Because of the early participation of the operator, it was expected that training would take no more than two weeks once the system was implemented, although nevertheless there would also be full supporting user and technical documentation. The emphasis in training was upon developing communication skills and familiarity with the help-desk system in order to use it effectively rather than upon developing particular personal troubleshooting capabilities. However, it was anticipated that the operator would undergo an explicit learning process in which she becomes proficient in applying the diagnostic rules encoded in the knowledge-base and may in fact begin to apply the rules without consultation with the help-desk. In order to prevent what might be an over confidence leading to possible errors, it is necessary that the nature of the helpdesk's interface changes to mirror the increased competence of the operator.

7.3.5.2. Help-desk Expertise

The help-desk 'experts' used in the project refer to staff of the computer department who are specialist in the mechanics or operations of the company's computer systems. The help-desk required the participation of four specialists, representing Vax systems hardware, the IBM Mainframe system, Personal Computers, and Communications and Network equipment. It was agreed that where possible, the help-desk should include trouble-shooting routines for computer hardware only since software diagnostics was complex, variable and difficult to represent in an expert system format.

Each expert decided the scope and depth of troubleshooting which could be delegated to the help-desk and this varied according to the complexity of the domain and individuals involved. For instance, almost a third of the knowledge for the Personal Computer (PC) and network domains could be acquired from secondary sources, such as manuals and documentation. By contrast, the IBM and VAX domains required almost exclusively upon the experiental knowledge of these systems' experts. A further feature of expertise which dictated the depth of troubleshooting' allowed, was what may be termed 'expertise causality. A difficulty for the Network and VAX experts particularly was in defining problems which could be bounded and identified as isolated faults with a well defined and simple debugging procedure; rather than being the result of a multiple order of problems linking different sets of equipment and levels of operation.

Despite these differences in domain characteristics, there were also a number of common features in the structure of knowledge which was represented by the help-desk. These features were constrained by the organisational role defined for the help-desk and by development and implementation considerations outlined in the last chapter:-

i) the experts considered the problems as 'trivial' although to the end-users they may have been significant,
ii) the expertise was in the form of structured rules rather than heuristics, and therefore of relatively straightforward representation,
iii) The span of problems covered was large but very shallow,
iv) The expertise was not exclusively about debugging faults but included advice about common queries and difficulties in using systems- understanding screen displays for example.
v) Each expert was not required to liaise with other experts or perform physically or mentally skilled functions in order to express his expertise.
vi) The nature of decision-making was such that it could be communicated by a 'non-specialist' intermediary (namely the help-desk operator) to end-users on a telephone; and therefore presumed that the rectification procedure or any other subsequent action could be undertaken by the end-user directly.

7.3.5.3. Help-desk End-users

The end-users of the help-desk are employees who have access to company supported end-user computing facilities such as terminals, printers, and personal computers. It also deals with more complex user roles in areas such as communications and systems development. As such there is a spectrum of user competences and needs which are addressed by the help-desk. Fry(1989) recommends a user questionnaire to determine the required level of support . However for the help-desk, this was impractical since there were nearly 700 users and so instead user support levels were identified from an analysis of reported queries. This produced a profile of targeted end-users for the help-desk which is shown in *Table 7.2.*

It is necessary to distinguish between direct and indirect users to qualify the figures shown in Table 7.2. Direct users are equipped with their own terminal or PC and are likely to be frequent users of their respective systems. By contrast, indirect users have access to but do not own computer hardware and are likely to be occasional users only. Table 7.2 has therefore sub-divided the end-user population using three measures of frequency according to : the number of terminals or personal computers in the company (and therefore direct users); the number of actual users (which includes both direct and direct users); and finally according to the number of queries or user problems received from each domain.

Table 7.2 provides a lot of information about the end-user population. It shows that IBM and VAX users tend to be direct user whilst PCs have a proportionately greater number of indirect users. Thus, although there are more IBM terminals than

PCs, the latter are used by a greater number of people. This is reflected by the greater number of PC related faults to IBM related faults shown in *Table 7.2*.

Table 7.2 Computer End-Users' Profile

Domain	**IBM**#	**VAX**#	**P.C**$	**Network**«
Direct Users	166	70	180	33
Total Users	211	95	365	40
Reported Faults*	26%	19%	49%	. 6%
Range of Users' Queries º	1-5	2-4	1-3	3-5

Notes: # includes related equipment such as terminals, printers keyboards, cables etc.

* average figures taken over the first three weeks of fault logging.

« this includes network terminals in addition to communications equipment

$ includes the use of PCs as network terminals

º A qualitative measure of the diversity in complexity of end-user calls

A further observation, verified by the experts, is that occasional users (indirect users) tend to know less about the system they use and therefore telephone more often about relatively simple problems (hence the low rating of complexity for the PCs). These problems moreover, relate to difficulties associated with operating the system, 'the keyboard has locked' for example, rather than because of systems failure. By contrast, direct users tend to be more familiar with their system and also have a knowledge of the system's history in terms of past faults, erratic behaviour and so on. Therefore direct users request expert assistance less often and when they do, the problems tend to be of a more complex nature than those of indirect users.

The implication of different levels of support, to satisfy the distinct needs of direct and indirect users, is that the help-desk arrangement of expert system and operator is able to identify individual needs and orientate the nature of the telephone consultation around these. This requires that the operator has special interpersonal skills and that certain features and facilities are provided by the expert system. These are discussed next.

7.3.6. Planning For User Acceptability

Mumford (1989) has shown in previous chapters how user acceptability of a technology may be increased if the user is involved directly in the development of the project. It may be implied from Young's analysis (described earlier in Section 7.2.) moreover, that participation at higher orders of abstraction (e.g. during contextual or conceptual levels) will improve the likelihood of acceptability than participation restricted to communications and physical levels alone. A problem in using acceptability as a performance criteria is, as Rouse and Morris (1986) argue, that it is associated with 'impacts' and the 'effects' of technology, when in fact acceptability is equally important as a front-end analysis factor, as it is during systems operations. Therefore user acceptability is understood here as a process of forward planning rather than a measure of reactions to the technology alone.

In the design of the help-desk, there were two levels of user acceptability as *Figure 7.5.* shows. At the level of the help-desk technology, it was essential that the operator was satisfied with the logical design of the system in terms of features, nature of the dialogue, explanation facilities and so on. This was achieved through direct and open consultations with the operator at an early (contextual) stage of design. This also provided an opportunity for the operator to establish and shape a personal role in the organisation. In doing so, a number of social issues were raised which would have been difficult to appreciate without the operator's implicit involvement:-

a) Security: that the operators's financial and social status in the company remained the same or was enhanced.
b) Accountability: that the operator received organisational support from management. This would take the form of political backing from functional managers in ensuring that their staff used the help-desk rather than rely on informal networks and communication channels.
c) Competence: the operator would require formal training and practice sessions in order to feel personally competent before using the help-desk in a live situation.
d) Legality : the quality of the information provided by the operator depends to a high degree on the integrity of the knowledge held in the system. In order that the operator is not held responsible for mis-diagnosis, the project manager is held legally responsible, in organisational terms, for the service.

Figure 7.5: Levels of Acceptability

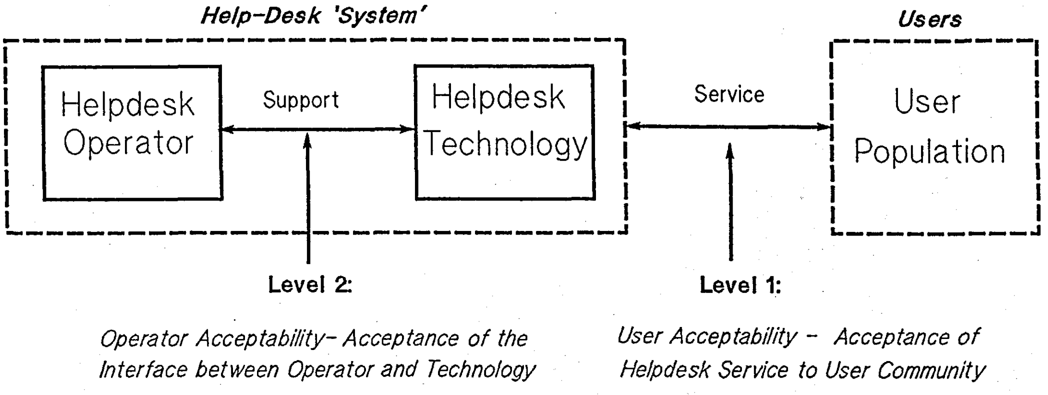


Figure 7.5 also defines a second level of user acceptance which looks at the relationships between the user community and the help-desk function. Acceptability, in this case was in terms of the response end-users gave to the help-desk as a troubleshooting *service* with little concern as to whether it used an expert system or not so long as it performed satisfactorily. User management defined the criteria for acceptability rather than experts or users and included : the range of services (handle queries as well as faults and offer other general information to end-users); the speed of response; the availability of services; the effectiveness of debugging; the degree of 'professionalism'; and whether there was clear support from computing & expert personnel.

7.4. Logical Design Features of the Help-desk.

Having outlined a specification for the Help-desk and state performance criteria, the development team began to define, in logical terms, design needs which would be required in order that these requirements could be met. The result was the stipulation of a number of design features which ought to be included in the help-desk. These are outlined in Figure 7.6. and include the following:-

7.4.1. End-User and System Profiles

Each end-user makes use of different printers, terminals and other hardware which may be used in a particular fashion. As well as differences in hardware configuration, each end-user will have very specific troubleshooting needs reflecting differences in technical competence, skills and ability, and also status in the company. This information is required by the operator in order to shape the nature of the consultation accordingly and moreover, present a suitable response to the end-user which is both technically valid and organisationally context sensitive. Although much of this information could have been acquired using a front-end 'question and answer' session before each knowledge-base consultation, this was likely to be time-consuming. Instead, a database of records profiling both the end-user and the systems that he or she has access to was proposed. This information would be imported automatically at the beginning of each consultation in response to a user key-code or identification number. Database profiles were constructed for the VAX, IBM and PC systems users; each were different according to the criteria each expert judged important in determining the detail and level of consultation. As an example, a few entries from the IBM users' profile are given below in **Table 7.3.**.

Name:	John SXXXXX	Userid/Username:	JES
Systems Used:	IBM	Ratings (1-5):	1
Department:	Technical Planning	Site:	Manchester
Terminal:	Telex 278	Connected Via:	Controller
Name:	Vincent HXXX	Userid/Usernames:	VJH VJH
Systems Used:	IBM VAX	Ratings (1-5):	1,2
Department:	C.A.D.	Site:	Manchester
Terminal:	VT220, VT320	Connected Via:	Network
Name:	Albert RXXXX	Userid/Username:	JES
Systems Used:	IBM MECCA	Ratings (1-5):	1,1
Department:	Estimating	Site:	Manchester
Terminal:	VT220	Connected Via:	3274-41D

Table 7.3: *Example from the Helpdesk's User and Systems Profile for IBM Users*

7.4.2. Addressing Terminology

A limiting factor in communicating technical concepts to users of mixed competence is that there is some uncertainty whether the end-user will understand the message or that the help-desk operator will be successful in communicating it effectively. In both cases, the use of esoteric terminology will compound these difficulties. Early attempts at interviewing the experts revealed a great number of abbreviations and acronyms which to the layman were unintelligible. In order to resolve this difficulty, it was first necessary to assess the current level of understanding held by both end-users and the help-desk operator. From this, it could be determined which terminology the user was familiar with and what kinds of explanation formats the user found effective; and then to operate the consultation at this level rather than the perceived level judged to be appropriate by the expert.

This created a central consultative role for the operator and user representative and meant that the shape of the interface was defined by the user rather than the expert.

It also meant that certain features were incorporated into the help-desk design which might not have been otherwise. These included an on-line glossary and list of acronyms; a pull-down menu of operational instructions; and context sensitive help facilities.

7.4.3. Help-desk Operator Response

The help-desk operator was required to identify root causes of problems from the key words and observations provided by the end-user. Two distinct problem-solving routes followed from this:-

a) **Mixed initiatives:** the user's keywords provided a direct means of input into the help-desk and enabled the operator to focus upon a particular level or aspect of the knowledge base (e.g. printer paper jam, how to produce a screen print etc.). In such cases, where the operator knows precisely what information is required, a 'mixed initiative system' (Morris:1987) is required so that the operator can volunteer information and thereby speed up the interaction and access the relevant information in the knowledge-base more quickly.

b) **Systematic tests**: the end-user's keywords and descriptions are symptomatic of a fault without providing clear evidence of the actual cause (for example, 'the screen is flickering' or 'I can't log-on to the system'). Here, the operator is expected to follow a more systematic route, starting from fundamental questions ('is the terminal switched on ?' for instance) and progressing down through successive levels of the knowledge base asking more specific and detailed questions until a fault is identified or it becomes evident that the problem should be referred to the expert. In such situations, the operator will require information from the end-user in order to test hypotheses put forward by the help-desk system. A number of help-desk facilities are required if this iterative process is to be effective. These include context sensitive explanations so that the operator understands the reasoning behind a particular decision and may communicate this to the end-user. It is also useful if the operator is able to change answers at a higher level in the decision-tree and therefore change the direction of decision-making logic, or re-trace the previous steps taken. This helps when end-users provide erroneous information or change their mind; and also refreshes the operator on what decisions have been taken so far. It is also important that the operator can save and recall particular consultations since an end-user may have to leave the phone or may want to find out more about the fault before contacting the help-desk again.

The operator may also wish to browse the help-desk. This would make it possible to peruse through the knowledge encapsulated in the system in order to get an overview of which issues are critical and also to determine the scope of a particular problem. In this role, the help-desk is operating as a stimulus to the operator rather than acting in a prescriptive capacity.

7.4.4. Prioritisation

It is intended that all end-user calls are diverted through the help-desk so that when problems are reported which are significant and obviously important, the operator should escalate them forthwith to the expert. All other problems are resolved directly by the operator or referred to a log of faults awaiting response from the relevant expert. A prioritisation system is important so that certain fault categories, or areas in the organisation where they occur, or even when certain end-users report, are given precedence over all other queries. When problems are escalated to the expert for this reason, it is useful to present a history of the fault to the expert, as a print-out, based upon the information gathered from the end-user and profile details.

7.4.5. Development Needs

As well as addressing end-user and operator needs, it was also important to consider features which would help the systems builder to maintain, update or modify the help-desk according to future needs and developments. One such need included the provision of error handling facilities which could detect obvious errors, provide error messages and the means to correct errors entered and detected by the operator or developer. The process of defining development features also provided valuable criteria for the selection of the development tool (described later in Section 7.7.).

On the basis of the above recommendations and the subsequent design of a logical or functional specification outlined in Figure 7.6., it was possible to progress to a next stage of pre-development planning which looked how the design might be accomplished in practice whilst conforming to certain project, organisational and cost constraints. This process is described next.

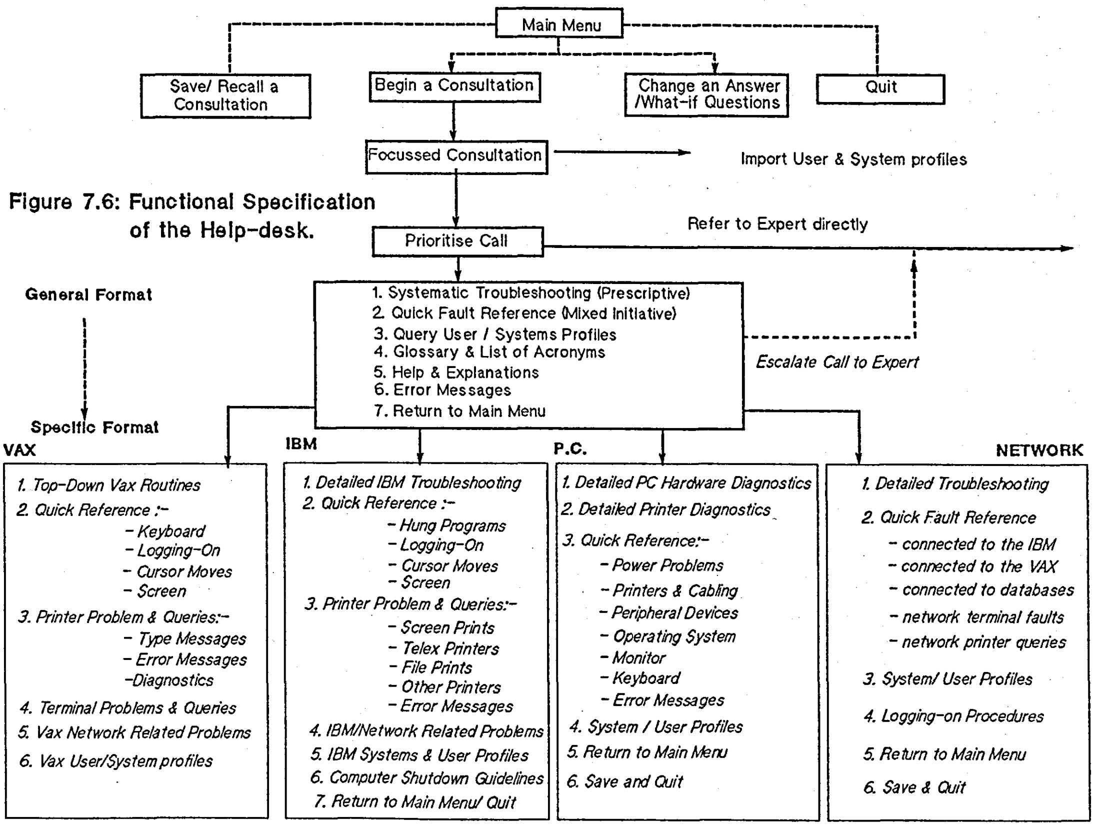


Figure 7.6: Functional Specification of the Help-desk.

7.5. Project Management and Development Costing

An essential part of project justification, whether expert system based or not, is the process of defining and attempting to quantify the total costs of development. In doing so, the cost structure should take account of not only development costs (labour, hardware, software, overheads etc), and the variable and organisational costs associated with operations, but also the future costs of the system, such as maintenance and anticipated project enhancements. However, this aspect of ES development is the least well understood and covered in the literature (Slatter *et al.*:1989). One reason for this, suggested by Bryant(1987), is that it is difficult to estimate completion times and effort because of the uncertainties of prototyping. Such uncertainties exist though, according to Hickman (1989), because no proper attempt has been made to define requirements prior to construction of the prototype. The approach adopted in this study favours the latter view, and it is because of the attention paid to pre-project issues in this and earlier chapters that such a costing exercise has been made possible.

The expectations from time, resource and cost estimates produced in this and subsequent sections are not that they will provide exact accounting figures, but that they provide working cost guide-lines from which resource and project cost deadlines may be set. The principle adopted is that some costing exercise is much better than none at all; however incomplete or inaccurate to the real world situation the figures prove to be. Moreover, it is argued that to base a justification for an ES project in 'value-added' terms alone (see for example Harbridge:1989) without attention to cost structures is incautious and, in terms of industry practice, commercially unacceptable (Lunn *et al*:1988).

7.5.1. Defining Development Timescale Estimates

Defining the estimated time to completion of the project requires that the process of development is broken down into sub-tasks from which the duration of each is evaluated. The assumptions made in estimating timescales differ according to the development methods adopted (which as Chapter 4 has shown, may vary considerably), the estimating criteria used (man-months of effort or timescale until project completion for example), and even the status of the project in the company (for instance first-time projects may operate to slack estimates to allow for learning). As a result, there is no standard or formal means of estimating timescales, as a recent Department of Trade and Industry report confirmed:-

> " There appears to be wide variations in the estimates of time and effort required to develop and deliver knowledge based solutions to problems in manufacturing. It is still true that in the majority of cases, systems have been developed as exploratory research projects and many of the true costs have not been recorded. " (MI:1989,pg64).

This report showed moreover, that many timescale estimates are made on the basis of the time to construct the knowledge base only (and associated activities such as knowledge acquisition), and omit later phases of verification, interface design and implementation. Thus empirical guide-lines which offer advice on developing systems provide estimates which are orientated around knowledge-base construction; for example, Hayes-Roth *et al.* adopt a 100 rules-a month basis for time estimates. Cutter argues that such guide-lines are simplistic and furthermore, do not take account of the variances in time estimates according to the class of expert system (in MI:89). For instance, Cutter found that for large planning systems (classed as having 4-500 rules), the range of development completion times was between 11 - 22.5 man-months of effort, whereas for diagnostic systems this was lower at between 10.5 - 17 man-months.

Since there appeared to be no creditable method of deriving timescales, estimates for the help-desk were made on the basis of experiences in developing the evaluation prototypes. From this, a project completion plan was produced and is reproduced in *Figure* 7.7. The assumptions made in generating these estimates are listed in the accompanying notes to Figure 7.7. The plan provides upper and lower estimates producing a total timescale range of approximately 9 - 15 months; in both cases there is a contingency to take account of personal and organisational learning curves. Estimates are measured in man-months, however since only one member, the author, was directly involved in developing the help-desk, this also indicated the total time to completion. This also accounts for why most activities shown in Figure 7.7 are in sequence and not in parallel.

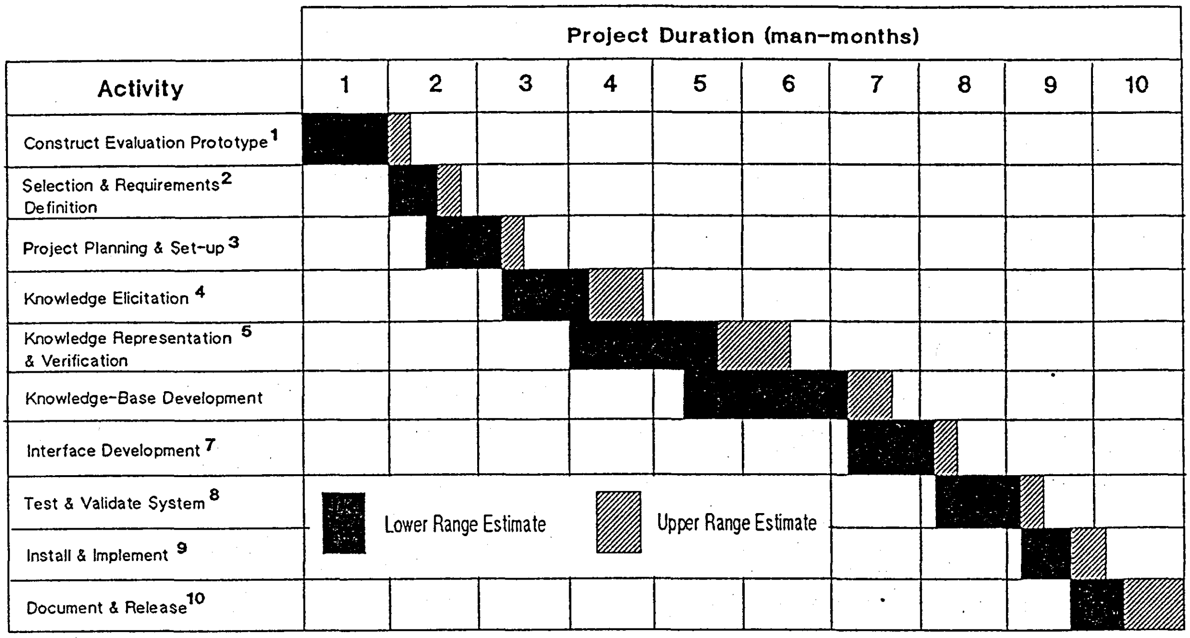


Figure 7.7: Project Plan for the Computer Hardware Fault Trouble-shooter

Notes to Figure 7.7:

1 Although the evaluation prototype was used as a stage of the selection process, it would be improper to distinguish it from other development activities because of the valuable interface and representation information it provided.

2. Requirements Definition is a full specification of the proposed project, as much as is possible that is, using among other tools the development suitability check-list

3. Project Planning and Set-up includes the organisation of the project, purchase of hardware and software, resource allocations etc.

4. Knowledge elicitation is taken to mean the processes of acquiring formal and informal information about the structure of the domain and aspects of decision-making using such techniques as interviewing, fault-logging, documentation and other secondary sources.

5. Knowledge representation refers to the intermediate representation of knowledge using IDEFo and fault-trees formalisms and, following verification, the preparation of pseudo programming logic for knowledge base development.

6. Knowledge-base development involves the physical design and programming of the helpdesk's knowledge-base. This phase also includes the selection and familiarisation of an appropriate tool based upon the knowledge representation needs.

7. Interface Development is classed as a distinct phase to knowledge-base development to reflect the importance attached to user interactions, dialogue requirements, explanation and other interfacing facilities. Considerable user involvement and iterative design may make this phase protracted.

8. Test and Validation includes the testing the interface with the operator and the 'service' to the user population as well as more detailed internal verification of the system itself.

9. Implementation includes training and awareness, learning and testing.

10. Before the system is released, full technical and user documentation is made available and advice on updating and maintenance should be provided.

7.5.2. Utilization and Operational Resource Estimates

Fry(1989) has looked at the resource implications of corporate information and advisory centres and much of this work has been adapted in the evaluation of resource needs for the help-desk. Using the work of Fry, it is possible to provide estimates of help-desk loading and performance characteristics. Firstly however, it is necessary to provide a number of definitions and assumptions from which formulae may be used.

a) Call loading: this is the total time taken up by an operator and other help-desk resources in attending to a user query. It is made up of two primary components as *Equation 7.1.* shows:

Call Loading = Maximum Duration + Call Administration **(Eq:7.1.)**

Maximum Duration refers to the time that the operator is in direct consultation with the user. A ceiling of *four minutes* was set for the help-desk and it was established that should the query take longer to resolve, then the problem was too complex anyway and should be passed onto the expert directly. During trials with the expert, the range of telephone calls lengths was from 30 seconds to 3 minutes for the types of problems to be represented by the help-desk. The figure of 4 minutes was therefore generous and made contingencies for the operator's learning curve, slow or obtrusive users and other factors. It is also necessary to take account of what Fry calls '*call administration*' which is the time the operator takes in dealing with a user query outside the phone call. For the help-desk, this may include: logging onto the help-desk; using user or system profiles; referral to experts; saving and recalling consultations; recording new instances of faults/queries; and liaison with the help-desk manager. Fry allows an administration period of *1-2 minutes* and therefore a period of 1½ will be used for the help-desk. Thus, from *Equation 7.1,* Call Loading is set at **5½ minutes.**

b) Personal Call level: This is the maximum number of calls that can be handled by a help-desk operator in a day; and, since it is planned to have only one operator, it is also the maximum loading of the help-desk. The personal call level is obtained from the number of working hours in a day divided by the call loading as Equation 7.2. shows,

Net Working Day / Call Loading = Personal Call Level **(Eq 7.2)**

From which it follows that

Maximum Loading = Personal Call Level x No. of Staff **(Eq. 7.3)**

The full working day at the client company was 8.5 hrs; however allowing for breaks and variances in work output, the actual productive work is different to the this figure. Fry provides a figure of 75% of the allocated working day as the level of productive output; and so,

Working Day = (75% of 8.5 hours) = 6.38 Hours
= 6.38 x 60 = <u>383 minutes</u>

Personal Call Level = (383 ÷ 5.5) = <u>70 calls per day</u>

Therefore Maximum loading of help-desk = *70 calls/day*

This is not a measure of success since it is conceivable that all these calls may have to be referred to the expert; but it does show the maximum possible number of calls which may be handled by the help-desk arrangement.

c) Peak Loading: The level of 70 calls per day is also an aggregate figure, since many of the helpdesk's queries are concentrated in peak periods. Therefore an hourly call rate should be calculated and expanded by the duration of the peak period. In the client company, there was a fairly even distribution of problems throughout the day with peaks if any over the periods 9.30-11.00am. and 2.30-3.30 pm - a total of 2½ hours or 150 minutes in all. From this, it is possible to calculate the maximum peak loading using the formula given below:

$$\boxed{\text{Max Peak Loading} = \frac{\text{Peak Period Duration x No. of Staff}}{\text{Call Loading}} \quad \textbf{(Eq 7.4)}}$$

Thus from *Equation 7.4,*

Max Peak Loading = (150 ÷ 5.5)x 1 = .150/5.5 = <u>27 calls over 2½ hours</u>

Again, this is a conservative figure and it is likely that the operator would be able to handle many more calls.

d) Staffing Levels: On the basis of the above values, the next step was to assess whether a single operator would be sufficient to handle the number of calls received and therefore whether more than one help-desk was required. A log was kept over a two week period from which the average daily load upon the experts was calculated. The results are shown in Table 7.4. below.:-

Query Area	IBM	VAX	PC SYSTEMS	NETWORK	TOTAL
Daily Load (Calls/day)*	18	13	32	4	67

* Includes calls from both Manchester and Preston Sites

Table 7.4: *Breakdown of Average Daily Loads.*

From Table 7.4., it can be seen that the operator would be expected to handle 67 calls a day on average and on the calculated value for the personal call level,

$$\boxed{\text{No. of staff required} = \frac{\text{Estimated Number of Calls}}{\text{Personal Call Level}} \quad \textbf{(Eq 7.5)}}$$

Therefore Staff Required = 67 ÷ 70 = <u>0.97 Staff</u>

Therefore despite the contingencies, one staff is still sufficient to deal with the maximum estimated loading and peak loads.

7.5.3. Project Cost Estimates

Total costs may be broken down into development costs and operating costs.

7.5.3.1. Help-desk Outlay and Development Cost Estimates

These are costs which arise from the construction and implementation of the help-desk and are expressed as a one-time cost. These costs do not include the costs of selection and programme development in the company, which should in principle be amortised over present and future development projects. Development costs have not been discounted because the purpose of estimating costs at this stage is to measure 'investment risk' rather than provide a quantification of actual benefits. This is elaborated upon in the forthcoming evaluation in Section 7.16.

a) Hardware Costs for Development: Since development could make use of existing hardware, and therefore there was no explicit capital investment, a share of the capital cost was attributed to the help-desk project as a percentage utilisation of the lifetime cost value of the machine. Since the accounting lifetime of personal computers is 5 years and the duration of development was one year, then the cost of using the machine as a development host is 20% of the initial cost of the machine. Thus,

If capital outlay of the machine (Compaq 386) = £2300
Then Hardware development costs = (20% of £2300) = **£460**

b) Maintenance Costs of Development Host: On the basis that hardware maintenance costs to the development machine over the year of its use is 8% of initial costs, using the company's financing assumptions, then:

Maintenance of Development Hardware = (8% of £2300) = **£184**

c) ES Software Tool Costs: A budgetary limit on the purchase of software was set at £2000. Since it was almost certain that a shell-based tool would be used and these ranged in price from between £1000 - £2000, the average of this was used. Thus,

Price of Software Tool = **£1500**

d) Systems Development Staff: Although the project made use of a number of people in the company, management and users notably, only the systems developer and the experts costed their time directly to the project. All other costs were carried as overheads of the company. Thus,

One research student working 10 months on the project = **£6000** *

(* *note: commercial rates would be significantly greater than this, but presumably the lead-time for development would be less since no training would be required. It should also be noted that many other development costs such as for training and documentation are also subsumed within this cost*).

e) The Cost of Expertise: There are different means of defining the costs of expertise from the most simplistic which is on the basis of income, to the most accurate, but most difficult to quantify, which is based on the real value of the expert to the company. Adopting an halfway position, the cost of expertise may be expressed in terms of the cost of replacement. This accounts for the individual income plus the organisational costs of training, induction, orientation and other factors which make up the period it takes for a new expert to become fully effective. On this basis,

Annual Cost of an expert = **£30,000** (*This is taken to be the average cost of expertise since the value of each of the four experts varies in the company.*)

Assuming 48 weeks/year , five working days a week and an effective working day of 6.375 hours (*this is the effective working day of the expert and is 75% of the full working day of 8.5 hours.*) then,

The hourly cost of an expert = (30,000 ÷ (48 x 5 x 6.375))
= **£ 19.61 per hour per expert**

Referring to *Figure 7.7*, the project plan shows that the duration of **direct** expert involvement in the project spans two levels, knowledge elicitation and knowledge representation and verification. Taking the 'worst case scenario from the estimates in *Figure 7.7.*, expert commitment to the project covers 17 weeks. However because of the nature of knowledge acquisition, from experiences in developing the prototypes,it is very difficult to spend more than five hours a week with an expert for the practical reasons that an expert seldom has much this much time to spare and five hours work with an expert generates three fold the amount of work. Moreover, since there were four experts a maximum commitment of 3 hours per week per expert was set. Thus the total expert commitment in hours for 4 experts over 17 weeks at 3 hours each per week is,

Total Expert Commitment=(3 x 4 x17)= **204 hours** and,
Thus the total cost of this commitment = £19.61 x 288 = **£5,647.06**. These costs are summarised in Table 7.5 below and add up to a total of **£ 12,144.44**

Table 7.5. *A Summary of Outlay and Development Costs*

Factor	**Cost (£)**
Percentage of Hardware Capital Cost	460.00
Maintenance Cost of Hardware	184.00
Software Tool	1500.00
Development Staff	6000.00
Expert Involvement	5647.06
Total	**12144.44**

It is significant from the structure of development costs shown in Table 7.5. that the software costs (i.e. the expert system component) accounts for only 12% of the total development costs and much less of total costs.

7.5.3.2. Help-desk On-Going Cost Estimates

These are costs which arise through the use of the Help-desk and are expressed as costs per annum. Since operating costs depend upon the utilisation of the help-desk, this section will make use of the call loading figures calculated earlier in *Section 7.5.2.*

a) Help-desk Hardware Delivery: The help-desk is to operate on the development machine and so the same costing assumptions may be made: Thus,

Cost of help-desk Hardware Delivery= **£ 460 per annum**
Maintenance cost of hardware host = **£ 184 per annum**

b) Help-desk Operator: On the basis of the grading characteristics of the help-desk operator defined in *Section 7.3.5.1.*, an annual charge of **£8000 per annum** is assumed.

c) Call Loading Costs: From the analysis of call loadings earlier in *Section 7.5.2.*, it was established that the average number of telephone calls per day was 67. On the assumption that the cost per call is £0.35 (1988 figures) then call charges amount to,

Call Costs per day = £0.35 x 67 = £23.45 per day
Therefore per annum the cost is £23.45 x (48 x 5) = **£5628**

d) The Costs of Expert Referral: Although the role of the expert diminishes, the help-desk as a service to the User community in the company still has to make use of expert resources for the more complex problems. A performance objective set for the help-desk, and described in *Section 7.3.2.*, was that it should be able to cope with 70% or more of all incoming queries. Expanding the analysis of calls shown in *Table 7.4.*, it is possible to see how accurate this goal is. *Table 7.6* shows the breakdown of calls according to those which the respective experts thought could be accommodated by the help-desk and those which emphatically should be handled by themselves. It shows that 48 calls in all were considered appropriate for the help-desk making roughly 70% of the total (48 - 67 = 72%).

Table 7.6. *A Revised Breakdown of Calls*

Query Area	**IBM**	VAX	**P.C.**	**Network**	**Total**
Total Daily Load	18	13	32	4	67
Appropriate For Help-desk	12	8	27	1	48
Suitable for Expert Only	6	5	5	3	19

The implication from this analysis is that if the help-desk is costed as a service to the user, then if for 30% of the time they require the support of the expert, then the cost of this support should be added to the operating costs of the help-desk. Thus on the assumption that one expert costs £19.61 per hour and, from *Box 1, Appendix 11*, that the total number of hours spent troubleshooting is 5.025 hours per day by all four experts then,

Daily Cost of Troubleshooting = 5.025 x £19.61 = **£98.53 / day**

Therefore as a yearly cost, this amounts to £(98.53x48x5) = **£23,647.1 per annum**

(This corresponds to the assertion in Box1 Appendix 11 that approximately one-fifth of the experts' time is taken up in troubleshooting and therefore one-fifth of the expert's total costs should be the costs of trouble-shooting; such that one-fifth of £120,000 = £24000).

Thus if the cost of troubleshooting all faults is this sum, then for the help-desk to refer 30% of all problems to the experts will cost **£(30% of £23647.1) = £7094.12**

e) Help-desk Maintenance Costs: This is one of the most contentious areas of costing expert systems. For large systems or where the domain is volatile, maintenance can be up to 30% of the initial development costs per annum. In the case of the Help-desk, the knowledge base is fairly stable for the VAX and IBM systems, but less so for the PC and Network systems since new equipment and configurations are added. As such a figure of **10 %** was considered reasonable.

Thus, costs of help-desk maintenance = £(10% of £12144.4))
= **£1,214.4 per annum**

Table 7.7 brings all help-desk operating costs together and it can be seen that the total operating cost amounts to **£22,580.564 per annum.**

Table 7.7. *A Summary of Help-desk Operating Costs Per Annum*

Cost Factor	**£ / annum**
Rule-book Hardware Depreciation	460.00
Hardware Maintenance	184.00
Help-desk Maintenance Costs	1214.44
Help-desk Operator Costs	8000.00
Expert Referral	7094.12
Call Costs	5628.00
TOTAL	**£22580.56**

f) Future Costs: In addition to maintenance, there are other future costs which may restrict the effective lifetime of the help-desk or may render it uneconomic after a short period of time. Future costs clearly depend too upon how widespread the system is in use. For instance if the call rate increased dramatically then more than one help-desk operator may be required: this would double the operating costs and would also generate additional costs such as licence costs and royalties on the software, increased training costs and other organisational costs. For the sake of costing, the lifetime of the help-desk was taken to be three years since no major technical or organisational changes were expected, although clearly no definite assurances could be made.

7.5.3.3. Costing the 'AS-IS' Situation

The principal alternative to using the Help-desk is to continue to perform diagnostics using expert resources (referred to as the 'as-is' situation). It does not account for the additional functions of the help-desk though, such as fault logging, skills archiving and so on. Furthermore, there are other non-technical alternatives to the help-desk such as improved training commitments, and these are discussed later. However, for accounting purposes, costs and benefits listed below are with respect to two alternatives, the 'as-is' situation or the help-desk. The costs in continuing the 'as-is' situation are on-going and are made up of two components:-

a) Cost of Expertise : The total annual cost of expert troubleshooting was calculated earlier to be **£23,647.1 per annum.**

b) Call Costs: The call costs for the expert are the same as for the help-desk and amount to **£5628 per annum**

Thus the total yearly costs for the 'as-is' situation are the sum of call costs and expert costs.

Total Costs for the 'AS-IS' situation = **£29,275.06**

7.5.3.4. Cost Comparison

A cost comparison of the help-desk proposal with respect to the as-is situation is given in *Table 7.8*. From this it is possible to calculate the payback: this is the duration after which the costs of the project are recovered by using the help-desk. At Payback, Total Costs of the Help-desk = Total costs of the as-is situation: thus over Á years,

Help-desk Development Costs + (Operating Costs).Á = (.As-Is Operating Costs).Á **(Eq. 7.6)**

Therefore £13, 791.06 + £22,745.23.Á = £29,275.06.Á

$$\text{from which}\quad Á = \frac{12144.44}{(29{,}275.1-22580.56.2)} = \mathbf{1.814\ years}$$

After the first year of operations, the help-desk is operating at a loss of £5449; much of the costs are recouped after the second year while at the end of year three there is a gain of £5284.17. This gain would be less if costs were discounted and the effects of increases in costs (principally labour) were accounted for.

Table 7.8 *A Cost Comparison for the Help-desk and AS-IS Situation*

Cost **Situation**	**Help-desk**	**AS-IS**
Development	£12144.44	----------
On-Going (per annum)	£22,580.56	£29,275.06
Total	**£34725.89**	**£29,275.06**

The robustness of these figures may be measured against the effects of changing certain key values. This also provides an interesting picture of the sensitivity of ES projects generally to changes in time and effort estimates and cost assumptions. Four main questions were asked of the help-desk:-

a) What would be the effects of increasing the costs of maintenance during the operational lifetime of the system ?
b) What would be the effects upon the justification of the help-desk if the value of expertise in the company were to change ? Furthermore, what is the break-even value of expertise ?
c) How does increasing the call rate affect the cost effectiveness of the help-desk ?
d) What are the effects of increasing the development time of the help-desk ?

In order to improve the clarity of the subsequent analyses, most of the calculations are included in Boxes 2,3,4,5 and 6 respectively of *Appendix XI.*

a) Increasing Maintenance Costs to 30% and 50% of Development Costs: If Development costs are £13,791.06 from Table 7.8., then maintenance costs are £3643.33 at 30%; and £6072.22 at 50%. Then Help-desk operating costs change to £25009 and £27438.34. These values are calculated in detail in Box 3, Appendix XI. In both cases there is an significant increase in the payback period for the help-desk from 1.81 years in the normal case to 2.85 years at 30% and 6.61 years at a 50% increase in maintenance costs. There are two practical implications from this result; firstly, it is essential that maintenance needs are planned for and incorporated in the original design of the expert system in order to reducè the uncertainties of maintenance costs. Secondly, for the help-desk to be cost-effective, maintenance costs must not be greater than **15-20%** of the development cost. Even small increases from this rate can have significant deleterious effects upon payback.

b) The Effects of Changing the Costs of Expertise by ± 25%: If the total costs of trouble-shooting to the company are £23,647 per annum, or £19.61 per hour, then by increasing costs by +25%, expertise rises to £24.51.per hour. Conversely, by decreasing this cost by 25%, expertise is reduced to £14.71 per hour. From the calculations in Box 4, Appendix XI. the following costs were generated,

Table 7.9 *A The Effects of Changing the Costs of Expertise: A Cost Summary*

Cost	**Help-desk**	**AS-IS Situation**
Development: - 25%	11144.84	----------
+ 25%	13144.04	
On-Going: - 25%	20707.14	23,363.25
+ 25%	24454.03	35,186.75
Total - 25%	31851.98	**23,363.25**
+ 25%	37721.62	**35,186.75**

From these figures, the payback was calculated to be **4.2** years at a 25% reduction in the cost of expertise and **1.2** years with a 25% increase. This is to be expected since if the value of expertise increases in the company, then there is a greater under-utilisation of the experts when they spend one fifth of their time performing routine decision-making, and subsequently the opportunity costs of not developing a help-desk are high (reflected by the rapid payback period of 1.2 years). Conversely, if the value of expertise is downgraded, then the opportunity costs are proportionately less and the value of the help-desk diminishes with the corresponding effect that the payback period increases significantly.

c) The Effects of Increasing Call Rates: It was expected that there would be a general increase in calls to the help-desk but the rate would differ according to the system. For the purpose of costing the effects of this increase, Table 7.10 provides forecasts over the three year period of planned operations.

Table 7.10 *Call Rate Changes Over a Three Year Period* *

Query Area	IBM	VAX	PCs	Network	Total
% call rate change	+ 10%	- 10%	+ 15%	+ 5%	---
At Year 0	18	13	32	4	67
At year 1 end	19.8	11.7	36.8	4.2	73
At Year 2 end	21.78	10.53	42.32	4.41	79
At Year 3 end	24	10	49	5	**88**

* *Figures based on discussions with computer department* staff

At the end of Year 3, the call rate rises to 88 calls a day which can be absorbed quite easily by the current as-is situation; but the help-desk, requiring 1.3 staff to operate it effectively, may require additional help with the possibility of an increased role for the service point at Preston. The effect of increasing call loading is to increase call costs and the costs of expert referral for the help-desk and of increasing total operating costs for the as-is situation but also to improve the operating advantage in using the help-desk in preference to experts. The precise values are given in Box 5 of Appendix XI. The main effect of these changes however is that the payback period is reduced to 1.5 years using operating costs for the first year with a net gain of £17580 after the third year comparing favourably against the gain of £7775.2 under constant call rates.

d) The Cost Effects of Increasing the Duration of Development Tasks: This factor was of the greatest concern since the literature abounds with cases of project over-runs (Bryant:1987). In this scenario, the intention was to see the effects of doubling the time taken to elicit and represent knowledge at an intermediate level and thereby program the knowledge base. This would affect the costs associated with the systems developer over this period, as Box 6 in Appendix XI shows, as well as the required involvement by the experts. By increasing the costs of development in this way there is also a proportionate increase in the maintenance costs of the hardware (valued at 10% development costs and classed as an operating or on-going cost function). This provides a new cost structure as Table 7.11 shows.

From the figures in *Table 7.11*, the payback rate was calculated at 3.1 years. this is considerable since not all the duration of tasks which make up the development lifecycle were increased. This highlights the sensitivity and importance of three principal development activities- knowledge acquisition, knowledge representation and knowledge-base verification- in determining the eventual cost-effectiveness of the help-desk.

Table 7.11 The Cost Effects of Doubling the Knowledge-Base Development

Cost(£)	Help-desk	AS-IS Situation
Development	£18544.88	---------
On-Going	£23,220.61	£29,275.06
Total	£41765.49	**£29,275.06**

7.5.4. Part One Summary: Justification for the Help-desk

There were a number of organisational difficulties in justifying the help-desk. The help-desk would require a full time operator which was considered by some in the company as an 'unproductive' allocation of staff, because as a supporting function, 'he would not load tapes or check the printers' for example. Secondly, the help-desk would effectively formalise the communication channels between users and experts which conflicted in areas with an informal network of contact points and relationships which had built up over the years. Due to the previous reliance on this 'grape-vine', it was difficult to produce reliable figures showing the need for the help-desk in terms of the number of calls, typical problems, call duration and so on. Despite these problems, the justification for the help-desk was made on two fronts: firstly that it would reduce the costs; and secondly that it would add value to the company's trouble-shooting service.

a) Cost Savings of the Help-desk

It was hoped that the costs of troubleshooting could be reduced by improving the utilisation of expertise. This was achieved by relieving specialist staff in VAX, IBM and PC systems support of the more routine and frequently occurring problems which may be delegated to less qualified staff. In doing so, the finite time of the expert is used more cost-effectively to address more significant problems and undertake developmental work. As a first project, it was important that the project was seen to offer a tangible financial return to compensate for the project risks in exploiting a new technology. The above analysis therefore represents an attempt to provide a cost justification for the help-desk. Some of the assumptions made were experiental rather than rigourous and may therefore be inappropriate for other projects. Despite this proviso, the cost comparison was favourable for the help-desk showing a reasonable payback of 1.8 years. The costing exercise however also sounded two principal cautions: first the sensitivity of cost benefits to increases in development time; and secondly the deleterious effects that the costs of maintenance can have on the lifecycle profitability of the help-desk.

b) Added Value

Although the above cost analysis proved satisfactory, it is important to note, particularly where the cost-benefits may appear marginal in the short-time, that there were additional benefits in developing the help-desk associated with *adding-value* to the company's troubleshooting service. Measuring added-value was difficult and required a longer-term, holistic approach to justifying the Help-desk using measures which are difficult to quantify in cost terms. As such the concept of added-value was used more to support the conventional cost based justification above rather than replace it. Despite this, the added-value potential provided by the Help-desk were significant and included the following:-

i) Providing an improved user service by offering a more professional, consistent response to user queries.
ii) Improve the image of the computing department in the company
iii) Better utilisation of experts: in that they are only consulted when they are really needed. Much of the expert's decisions may be described as routine.
iv) Allow experts to undertake work which is of more value to the company
v) Capturing the skills of the experts for all time and therefore reducing dependency. Also providing a useful training aid.
vi) The ability to incorporate standard company procedures and codes of practice into the troubleshooting process
vii) To demonstrate the commercial and organisational value of experts systems and facilitate its transfer into the organisation.
viii) Providing greater capabilities: for instance, the help-desk would provide a central record of user calls (having all user calls coming into one point means that each problem can be logged centrally) and a reference of problem types, occurrences etc., all of which could help to identify trends and forecasts.

The improved cost profile from such benefits results from the reduction in the opportunity costs of inefficiency in continuing to use the experts rather than any absolute gain.

c) Non-Technology Based Alternatives

Although the emphasis in justification so far has been in the possible cost benefits and added-value of the help-desk as an improvement to the as-is situation, there were also other non-technical alternatives to either of these. The most obvious alternative was a devolution of trouble-shooting knowledge through an increased commitment to training and tuition. In some areas of the company such as engineering, this took place and, providing that the turnover of staff was small, proved cost effective. However for other areas, such as administration and training, this would be impractical because of the large number of staff and the high turnover rate.

A second possible alternative was to develop a manual help-desk. This could be manned on a daily basis by each of the experts in turn: thus for four days of the week an expert would have no troubleshooting commitments other than for important problems. This was feasible since each of the experts had a good working knowledge of the other's domain. However in practice it was a politically sensitive issue to appoint an 'expert' to answer telephone calls unless one was employed from outside the company specifically to undertake this role. Moreover, it still represented a under-utilisation of expert resources and therefore the gains from adopting this approach were marginal.

A final alternative was to relieve a computer operator (who may be classed as a 'semi-specialist') to concentrate wholly on fault troubleshooting, offering a dedicated manual help-desk to users. This was a cost effective alternative to the expert help-desk but was problematic for two reasons: first that it was unlikely that any of the computer operators would be interested in the job; and secondly, none of the experts were free to spend the considerable time required to train the operator up to an appropriate competence.

Part Two: Help-desk Development, Implementation & Evaluation

Having defined project requirements and planned for change as much as possible, the processes of Help-desk development began with knowledge acquisition. It is at this point where there is a change in individual roles and responsibilities and a subsequent change in the nature of user involvement as the focus of development shifts from contextual and conceptual levels to physical and communication levels of participation as defined by *Figure 7.2.*

7.6. Knowledge Acquisition

Knowledge acquisition involves eliciting knowledge from the experts, representing it in an appropriate manner so that it may be verified and then translated into a format which is appropriate for knowledge-base programming. There are a plethora of techniques for elicitation and representation ranging from the highly formal and structured tools (e.g. Rajin *et al*: 1989) inherited from software engineering practice to the highly unstructured techniques of knowledge gauging using interactive prototyping (e.g. Harmon & King:1985). This section describes the evolution of a Knowledge Acquisition(KA) approach, which although it adopts some of the concepts and techniques of existing KA tools, is different in two ways: first, that it would make use of existing skills, capabilities and tools, like IDEFo, rather than introduce 'yet another technique' into the company. Secondly, that it would allow a full mapping of the knowledge base in logical terms and thus facilitate greater user and expert participation.

However a discussion of tools must be preceded by a full analysis of the knowledge characteristics of the domain so that these can shape the choice of acquisition and representation techniques best suited to its needs. This is also important in selecting a suitable knowledge-based development tool.

7.6.1. Defining Knowledge Characteristics for the Help-desk

In order to express the knowledge requirements of the Help-desk for the purposes of defining acquisition methods it is necessary to define properties and dimensions of knowledge which describe it accurately. Worden (1989) lists a number of knowledge attributes which constrain the processes of knowledge acquisition and representation, and these are discussed with respect to the design of the Help-desk.

a) Shallow knowledge: shallow knowledge is based on the experts experience of a domain and need not equate with an understanding of precisely what has happened. On the basis of experiences and observation, the expert constructs rules-of-thumb or heuristics which provide a means of approximation by which to identify probable faults in the case of the Help-desk.

b) Deep Knowledge: the Help-desk is designed to off-load the bulk of what now may be called shallow problems from the expert. However there is a second set of problems faced in the client company which require deep reasoning (this has also been called *model based reasoning* or *reasoning from first principles* (Milne: 1987) and *causal reasoning* (Atwood et al: 1986). Deep reasoning reflects the experts ability to understand the theory or first principles behind the domain thereby making it possible to use domain independent causal mechanisms in order to operate on a model of the system being analysed. Decision-making then proceeds by observing the differences between the 'real world' situation and the behaviour of the model. An example showing the comparison between shallow and deep reasoning in the company was when the VAX operator encountered a new problem on the disc drive system. Because it was a new problem, the operator began by tracing through the circuit diagram and carrying out tests that verify that each module of the drive was functioning properly; from this he eventually solved the problem. The operator relied on personal knowledge of electronics theory plus the use of circuit models and test documentation to systematically work through the problem. The deep-level reasoning was on the basis of applying theoretical knowledge in a practical context. Having solved the problem, the operator was able to rationalise the experience and make general observations about procedures. If the problem were to be repeated enough times, the operator would be able to refine these observations and formulate rules of thumb providing the basis of a shallow-based reasoning approach. However, a characteristic of these problems was that although there were many, they occurred infrequently and it was not possible technically , nor was it advisable organisationally, to devolve such complex decision-making processes to the Help-desk. Thus although both causal and heuristic models have been used for diagnostic problems (see Rodi *et al* (1989) and Lister (1989) for instance), the organisational role intended for the Help-desk in resolving the routine experiental based component of the experts' knowledge made deep reasoning unnecessary.

c) Procedural and Declarative Knowledge: Declarative knowledge is based on statements of occurrences, whilst procedural knowledge defines sequences of steps to perform functions. Although an expert system adopts the former approach in the structuring of rules and is necessary in defining the experiental knowledge of the expert, the Help-desk also has a substantive 'text book' component, in defining computing procedures and standards for instance, where a procedural approach may be used.

d) Meta-Level Knowledge: This is knowledge about knowledge in which higher level knowledge is used to control and understand lower level knowledge. For instance, when a printer ceases to work, knowledge about the internal operations of the printer may be useful at later stages, but it is first necessary to start at a higher level of understanding which addresses the nature of the interface of the printer with the terminal or PC, power connections etc. Meta-knowledge provides a context in which base-levels of knowledge operate and therefore defines the level of detail and direction of problem-solving. The Help-desk requires meta-knowledge in order to structure and prioritise the many sub-modules of knowledge about particular computing equipment (printers, terminals, PCs, networks, mainframes), define how equipment interconnects (some personal computers are connected to two printers via a switch and may be used as a network terminal for example) and the new operating and troubleshooting characteristics which emerge when hardware is linked.

e) Abstract versus Concrete Knowledge: Heuristics are essentially approximations based on experience and may therefore be considered as an abstraction based upon a possibly imprecise notion of the actual situation (Worden). Indeed for many areas of Help-desk trouble-shooting, especially in communicating the process to the user, abstractions are necessary - it doesn't matter how exacting the response to a query is so long as the user understands precisely what to do from it. However, where the problem becomes more procedural and systematic, the nature of the knowledge transfer process is more concrete and well defined. Both may be used in the same consultation with abstract, meta-knowledge being used to identify and select a lower-level, more concrete, knowledge structure.

f) Symbolic versus Analogue: With regard to the Help-desk, this issue is fundamental. According to Worden, a representation is analogue when "..the parts of the representation can be put in correspondence with the parts of the thing being represented." For instance an analogue watch provides a context by which the hour and second fingers may be viewed in relation to other settings. It provides much more deep and abstract information about the situation than a digital watch which provides a concrete value but without a relational context . A person with a digital watch moreover, often has to make a translation to analogue to appreciate the real time and therefore it is symbolic. This analogy is of value because it mirrors the situation of the Help-desk, as Figure 7.8 shows. The expert provides symbolic knowledge because it is a representation of how he actually performs troubleshooting. This knowledge is encoded in a digitized form in the expert system. It is the function of the operator to 'animate' this knowledge and with sensitivity to the problem, the individual and organisational contexts and present it in an analogue state to the user. This has a number of implications for the processes of knowledge acquisition and representation :-

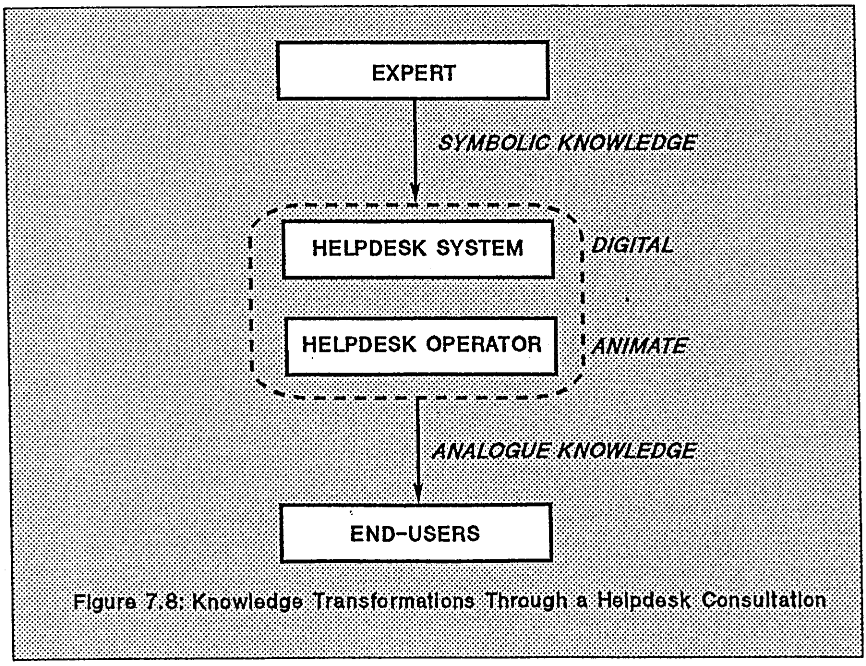


Figure 7.8: Knowledge Transformations Through a Helpdesk Consultation

i) that the expert must appreciate the organisational and personal settings in which his knowledge is to be used, i.e. think like a user not an expert,
ii) that the design process should be analogue, with pre- eminence attached to understanding functional knowledge requirements rather than fitting symbolic knowledge to the technology. The latter is a task for the systems developer after a logical representation of knowledge has been verified,
iii) that the operator has the technical and inter-personal skills to apply knowledge and add context to essentially symbolic and context insensitive knowledge,
iv) that the user interface between the operator and the Help-desk facilitates valid interpretation and understanding of analogue knowledge.

*7.6.2. Using Knowledge: Problem-Solving Strateg*ies

If the knowledge is highly causal, then a **model** for the knowledge can be identified. Conversely, if the knowledge is largely heuristic (i.e. uses 'rules-of-thumb'), then a **representation structure** is necessary, this being identified earlier as the most appropriate form for the Help-desk. As well as defining different types of knowledge, it is also important to determine how the expert uses this knowledge. As rules-of-thumb are the expert's primary form of knowledge in the Help-desk, a useful representation for such problem solving is Clancy's model of heuristic classification (1985). He proposed that there are three stages to a simple diagnosis problem which are shown in Figure 7.9.

i) Data abstraction: where a mass of low level problem data suggesting symptoms for example, is transformed into abstract aggregated features which are most likely to point towards a solution.

ii) Heuristic Match: where a 'great leap' is made from the abstracted problem features to classes of solutions.

iii) Refinement: where further problem data is used to move from broad classes of solution to the detailed solution of a particular problem.

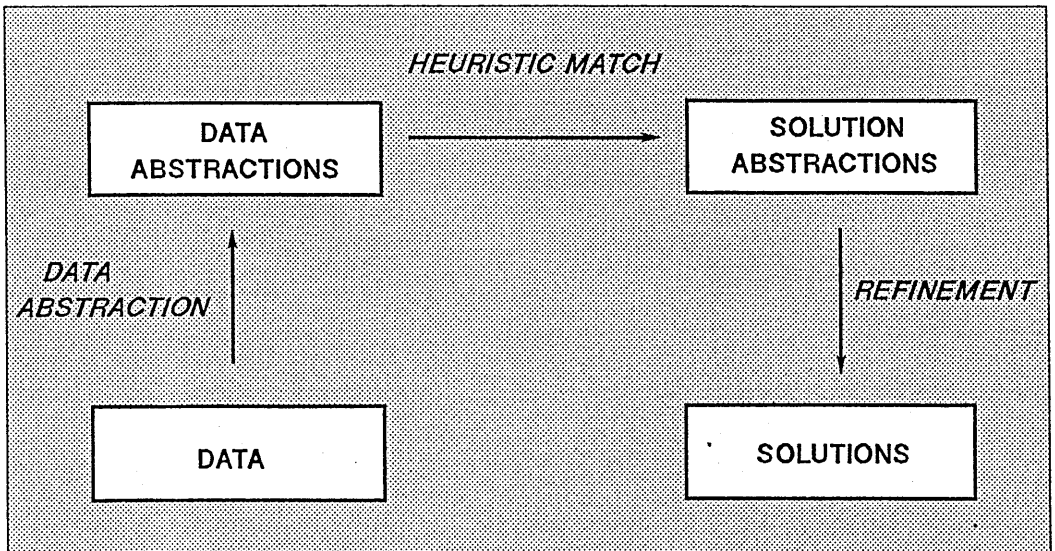


Figure 7.9: Clancey's Model of Heuristic Classification.

This model captures the general principles of how knowledge gained through experience is organised and used. The data abstraction stage equates simply to the user telephone call where the expert looks for clues and symptoms so that he can put the fault situation into a broad category. This information, at the problem solving level at which the Help-desk is to operate, is often provided by the users themselves: for example,

" ...I typed the queue entry sequence so why can't I get a screen print ? "
"....My terminal flickered and then the power went off "
"....I pressed the PF2 key and it logged me out. What did I do wrong ?

The heuristic match or leap is where the computer experts take these high level symptoms and identifies a relationship which may define a broad solution class. For instance in the second example above, the expert would likely deduce that the general solution category would be 'power problem' . Then, in the solution refinement stage the expert would attempt to gather further evidence to refine the broad solution class into a detailed diagnosis or repair. In the case of the power fault for example, the expert might instruct the user to first check that all power leads are connected and that there are no loose wires. This process is cyclical is there are different categories or levels of knowledge.

The similarity of the computer experts' problem-solving to that of Clancy's model has a number of implications on the methods of representing this knowledge which are discussed in detail elsewhere. Foremost is that it is based on shallow or compiled knowledge (Chandraskaran and Mittal:1983) and therefore is amenable to rule-based representations of the format If<antecendent> then <consequent>. This suggests that rule-based programming techniques are suitable rather than more complex causal mechanisms such as pattern matching; and moreover, that decision-trees and simple fault classification methods of knowledge representation are appropriate rather than Semantic Networks* and Frames* for instance (Milne: 1987). It also suggests that relatively straightforward searching strategies may be adopted for controlling the inferencing of the knowledge base.

7.6.3 Current Approaches in Knowledge Acquisition

Having gauged knowledge characteristics and requirements from a sample set of knowledge, the next process was to determine an appropriate approach for knowledge acquisition for the whole domain. A first task in this process was to evaluate current knowledge elicitation methods and provide a classification of manual and automated tools which have evolved from these, from which, appropriate forms of acquisition may be adopted for the Help-desk.

7.6.3.1. Methods of Knowledge Elicitation

There are a wide range of methods by which to acquire and distil knowledge from the expert. Some of the more frequently mentioned are described below (from Beerel (1987) and Welbank (1983) primarily) together with an account of their relative merits and weaknesses (ITSC:1989).

a) Interviewing: This is the most often used technique. Interviews may be structured in which case the course of the discussion between the knowledge engineer and the expert follow as a specific path of planned questions. Alternatively, they may be unstructured and conversational allowing the expert to express freely what he or she feels to be important and thus define general issues and settings.

b) Introspection: Beerel refers to a process of introspection where the expert acts as expert and knowledge engineer. By examining personal problem-solving and knowledge processes, the expert builds a system which replicates this knowledge.

---

* *Details of Semantic Networks and Frames are given in Sections 3.2.2. and 3.2.3. of Appendix I respectively.*

However it is not advised for the reason that an objective and skilled knowledge engineer is more able to elicit the expert's knowledge than the expert.

c) **Observation.** An alternative approach to asking what an expert does is to watch how it is actually being done through observation. The experts provides a commentary as he or she undergoes the problem-solving procedures. Welbank(1983) loosely defines this as protocol analysis. Observation techniques have a number of benefits in that as the expert 'walks-through' a case, he is more able to articulate his reasoning to the knowledge engineer. Moreover, many of the rules-of-thumb that the expert applies are 'second nature' and the expert may not be fully conscious of them until they are required in practice. An IBM study (see ITSC:1989) found observational techniques to be useful in capturing procedural knowledge and valuable in understanding the characteristics of the users of the application. Where the expert 'walked-through' the problem; it was also possible to capture tacit and heuristic knowledge. However these techniques were usually time consuming and may be considered by the experts and users as intrusive if they are sensitive to being observed in this way.

d) **Induction:** Here the expert tries to give an exhaustive set of examples of problems in the domain. Using a suitable algorithm, rules can then be induced by computer from these examples. These rules are referred back to the expert for verification from which a valid set of extended rules may be developed. Machine induction is useful in defining heuristic knowledge once the attributes of knowledge have been identified. However the integrity of the knowledge is a function of the number of test cases gathered. A large number of test cases are required to define a complete rule set.

e) **Procedure Animation**: If the expert system being built is only attempting to animate or make more accessible a set of written procedures such as codes of practice company standards or manuals, then the knowledge elicitation process is nearly complete. The task remaining for the expert is to provide a context to this formalised and procedural knowledge by defining meta-knowledge which provides a mapping and representation structure for this knowledge.

f) **Repertory Grid:** There are numerous manual techniques such as card sorting (Welbank) and automated tools (Gutierrez: 1988) associated with this method. The central approach is to ask experts to define a series of objects in their particular domain (such as different faults or failure modes for example). Each expert is then presented with three objects and the expert is required to say in what way two of the three are alike and different from the third. As all possible combination of objects are presented, the knowledge engineer can deduce the way in which objects are distinguished from each other. This approach is useful because it captures links, values, actions, concepts and some tacit knowledge; its limitations lie in the difficulty of statistical interpretation of the results.

g) **Prototyping:** Beerel defines prototyping itself as a forms of knowledge acquisition in that the knowledge engineer and expert generate knowledge ideas which may be tested and verified *en route* through using the system. A problem with prototyping as a form of elicitation, as Aylett (1990) has argued, is that the knowledge engineer is compelled to fitting the expertise to the representation of the tool. A consequence is that, as with introspection, the expert focuses upon a symbolic or digitised form of knowledge which is appropriate for the expert system rather than upon the analogue characteristics for its use in the organisation. Presenting an analogue picture of trouble-shooting as *Figure 7.8* has shown is essential if the expert-operator-user relationship of the Help-desk is to be successful. A further problem identified by Bradley *et al* (1989) in using prototyping for knowledge elicitation is that the time required for the knowledge engineer to create or modify the prototype program is often too long to retain the concentration of the expert causing a break in the line of reasoning.

7.6.3.2. Tools for Knowledge Acquisition

From the above approaches a number of tools have been adapted which facilitate both the elicitation and representation process. Aylett identifies four classes of knowledge acquisition tool:

i) Automatic Knowledge Acquisition Tools: the function of these tools is to reduce the knowledge engineer's role (the knowledge engineer being the person who elicits the knowledge from the expert) so that knowledge may be elicited directly from the expert who is able to structure it automatically. Tools for the automatic elicitation of knowledge include induction in which the system learns over a set of examples, (e.g. Xi Rule Knowledgemaker and ExpertEase are tools commercially available for this purpose); Repertory Grid tools which prompt the expert interactively to name significant objects in a domain from which rule attributes or constructs are identified (e.g. Aquinas and NEXPERT-OBJECT); and model-based elicitation techniques which structure the domain by modelling knowledge tasks and inferencing procedures (e.g. Test-bench).

ii) Methodologies & Toolkits: Aylett identifies a second class of KA tools which provides methods and toolkits for the knowledge engineer with the aim of making the engineer's role more effective. Examples of these tools include KEATS (Rajin *et al:* 1989) and Shelly (Anjewierden:1987) both of which are used as front-ends to the large software engineering orientated methodologies such as KADS which was described in *Chapter 4.*

iii) 'Free-Standing' Tools: Aylett uses this classification to describe a set of tools which provide free standing support for particular elicitation techniques or small-scale facilities for the structuring of knowledge. These are aimed at assisting the knowledge engineer with an emphasis upon low-cost , ease of use and ease of implementation.

iv) 'Making-Do': Aylett defines a final class of KA tools not directly intended for KA. These include the use of standard packages such as drawing packages and other software, as a low cost, low commitment option to KA based upon using existing company facilities in a distinct way for knowledge analysis.

7.6.4. Adopting an Approach in the Client Company

For practical reasons (time, resource and costs constraints) the 'making-do approach' described above was the only available option to KA in the company. However, in defining which existing company tools and methods were appropriate a there were a number of prerequisites:-

7.6.4.1. The need for an Intermediate Representation

It was mentioned earlier that a problem with prototyping, iterative programming and other system constrained methods and tools for knowledge acquisition is that the definition of knowledge itself is constrained by the representation capabilities of the tool. As well as restricting participation by users and experts this makes knowledge testing and verification difficult. A condition of the approach in the company therefore was that there should be what Hayward (1986) calls an Intermediate Representation, Hickman (1989) calls an external model and Martinez and Sobol (1989) call a logic model: all refer to the same process of representing the experts knowledge in a functional way which is independent of tools or particular techniques. A problem with all these approaches however is that they are complex, require a automated tools or bureaucratic methodology to work properly and are orientated towards large-scale integrated ES projects. By contrast, the Help-desk was a relatively small project relying on a 'making-do' approach to knowledge acquisition. As such, it was determined that although an intermediate representation was necessary no complex ES specific tools or techniques could be used. There were additional potential benefits in providing an intermediate representation of the knowledge acquired for the Help-desk based on the experiences of other similar projects (Esprit:1987):-

i) If the form of representation is expert orientated, it would allow the knowledge of the Help-desk to be checked much more effectively than if it were closer to the final implementation phase of the system.

ii) The choice of representation approach is made on the basis of the expressed knowledge needs of the expert and is made machine and implementation independent. This again means that the representation can be discussed and validated by the expert, and where necessary the operator.

iii) Since the structured can be varied according to the forms of knowledge and expertise identified, the development of the intermediate representation can be planned and controlled more effectively,

iv) Problems of representation can be identified at an early stage, thereby avoiding the potential for problems to build up in a prototype only to be discovered when resources and time are 'sunk' in the project.

v) The intermediate representation itself provides a useful basis for the system's documentation.

vi) The imposition of a formal structure in the representation process allows for a methodical maintenance of the contained knowledge.

The latter point is particularly important in the case of the Help-desk since the system developer (i.e. the author) would be leaving the company shortly after its implementation.

7.6.4.2. Different Levels of Representation

It is evident from *Section 7.6.2.*, in defining the knowledge characteristics of the Help-desk that as there are different levels of knowledge, abstract/concrete, meta/base-level, symbolic/analogue etc, so there should be different levels of knowledge representation. Hickman (1989), defines four layers of expertise and knowledge :-

Strategic Layer:	Consisting of the knowledge required to determine the overall approach taken in a given problem domain.
Task Layer:	Consisting of the tasks which must be executed in given problem situations. This layer is more procedural in nature than the other layers.
Inference Layer:	Consisting of the relationships which can be applied, using domain layer facts/relationships as 'data'.
Domain Layer:	Consisting of all the basic facts and relationships used within the problem solving domain.

From this structure, a three layered approach to knowledge elicitation and representation was adopted in the organisation. *Figure 7.10* shows, these are contextual, structural and programming logic: each addresses different knowledge components of the Help-desk and as intermediate representations precede issues of tool selection and prototyping.

7.6.4.3: Knowledge Elicitation and Representation at the Contextual Level

Green et al (1989) argue that it is essential to define knowledge from an organisational context first in terms of the organisation of tasks and specification of top-level (meta) knowledge flows. For this they prescribe a 'transactional' systems based modelling technique. A similar way of capturing high level structures of tasks is though the IDEFo methodology described in detail in Chapter 5. All of the experts had participated in the IDEFo project and were familiar with the notation. As a functional modelling technique independent of both tools and specific problems, it has a number of qualities which could be used during knowledge acquisition, particularly at a contextual level.

a) High-level interviews: A first stage of elicitation was to define with the experts the scope of their knowledge, knowledge boundaries, major information inputs outputs processes and mechanisms and in this sense mirrored precisely the function of IDEFo but at a more detailed level. The IDEFo models of the organisation of the computer function (Appendix IVa) and computer mechanisms (Appendix IVb) were used to map out their expertise at a high level in terms of system boundaries, activities, primary knowledge flows and interactions with user functions. The model provided an inventory of all computer equipment which would be covered by the Help-desk plus all end-user functions which would be affected by its implementation.

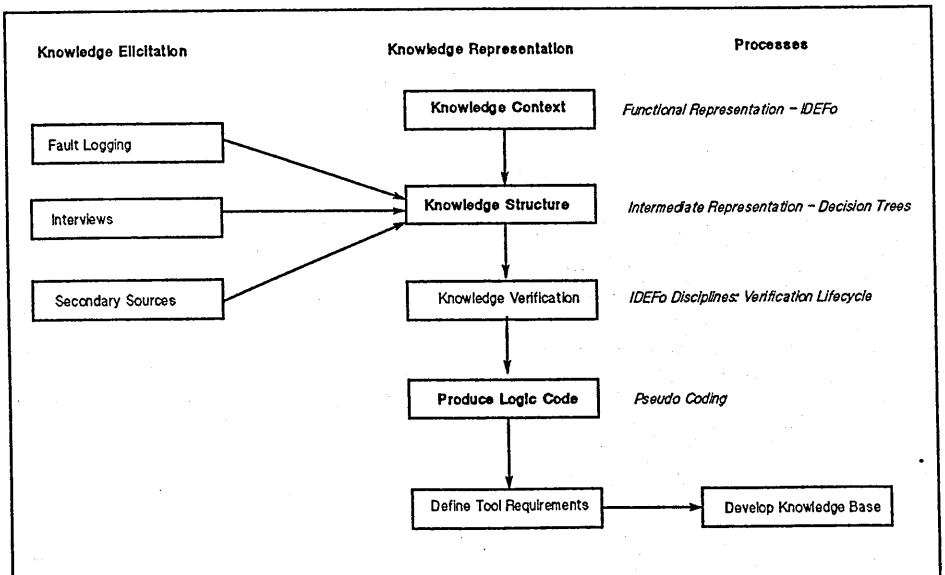


Figure 7.10: An Overview of the Knowledge Acquisition Approach in the Company

b) Communications: The IDEFo models were used during the interviews as a effective communication device which was commonly understood and therefore of much help to the knowledge engineer (interviewer) as well as to the expert (interviewee). Since the experts were familiar with IDEFo, they provided rough sketches and ideas using this notation.

c) Knowledge Context: It was important for the expert to appreciate the role and context of his knowledge in the organisation because of the necessity for the Help-desk, as a organisational service, to provide analogue knowledge to users (see *Section 7.6.2.* for the distinction between analogue and digital knowledge). IDEFo helped to establish an organisational context by showing the use of expertise at higher and lower levels in the company. Although expertise was generally from single sources corresponding to items of computer equipment, there were areas where the experts' domains crossed. For example, the IBM and VAX had similar networking operations and used the same printers and terminals. The IDEFo model was used therefore to help decide which of the experts would cover specific areas of expertise. Using more than one expert source for one item of equipment was not planned for the reason that knowledge acquisition and representation become more time consuming (Wolf:1989).

Using IDEFo as a complement to high-level interviewing in this way, equipped the author with a good broad understanding computer operations. This factor alone increased the effectiveness of knowledge acquisition and verification for, as Wolf(1989) notes, 'one of the real problems that can occur in the development of such systems (expert systems) is failure of the knowledge engineer through ignorance of the domain to ask the right questions'(p139.). The products from this stage of interviewing are the definition and description of the extent and context of domain knowledge to be contained by the system. It also provided secondary sources of knowledge such as codes of practice, computer manuals, and text-books which could be used to reinforce more detailed knowledge provided by the experts at later stages.

7.6.4.4. Knowledge Elicitation and Representation at the Structural Level

The above level defined the organisational and knowledge context in which trouble-shooting was undertaken, but IDEFo and other mechanisms at this level fail to map the decision-making process itself. For this a more explicit representation formalism was required and therefore classification trees Clancey (1985), also referred to as Fault Trees (Bradley *et al*: 1989) and Decision Trees (ITSC:1989), were used to provide a structured intermediate representation. The main purpose of the fault tree representation was to narrow down the area in which a fault is likely to have occurred and then to attempt an increasingly detailed explanation of its cause with possible remedial action or escalation instructions to the expert. Producing fault trees also provides a formal and systematic means of recording and verifying the knowledge.

A fault tree structure is shown in Figure 7.11. Because of the highly structured heuristic content of the Help-desk, nearly all knowledge could be represented in this way. *Figure 7.11* shows that the fault tree has a root node which describes a main symptom. Each node beneath the root then represents a decision that is based on the response to a diagnostic question. An example tree structure is shown in *Figure 7.12*. If the tree is traversed in a left-to-right, depth-first (*see Appendix I for a definition*), the logic of trouble-shooting from the tree is that symptom categories are at the top and specific symptom, failures and repair nodes, including expert escalation, decomposing down from these category nodes. This makes it possible to narrow the choices immediately by eliminating categories of problems at the higher nodes. The tree structure is replicated at different levels or *classes* so that one tree may become a branch of a higher-level tree structure. An important point that determines the efficiency of fault trees is that the branches from each node should represent roughly equal sub-divisions of the class represented by that node. Often, a node will just split into two branches, and so the test (or guard condition) that determines the split should attempt to knock out a major fault class.

The knowledge component upon which the fault trees were based came from a number of sources, with each source generating different levels and forms of knowledge:-

a) **In-depth Interviews:** The bulk of the Help-desk's knowledge came from regular weekly interviews with each of the experts. After each interview, fault trees were composed and presented the following week for verification, iteration and refinement. Such 'paper-based' verification was very similar to the reader/author cycle for IDEFo (see section 5.8.1) and in fact many of the disciplines of IDEFo such as issues and conflicts reporting; formal documentation; and implementation and validation procedures were adopted to the fault tree to make it more of a method of intermediate representation than a simply a tool. However, care was taken not to 'intellectualise' the process; the priority was to provide a simple though systematic means of expressing the knowledge of the expert in a way which the experts were comfortable with. The benefits associated with communicating in the experts' terms were that they could continue to express their knowledge in this way through-out the lifetime of the Help-desk and thereby define maintenance needs directly. Also, fault-trees were very easy to understand and the decision-logic, and therefore structured rules, could be distiled from the trees with some ease once they had been verified. In one instance, this manual representation of faults was used in itself to establish the cause of a computer fault. The problem concerned the tripping of a miniature circuit breaker in the computer operations room which consequently caused the system to shut-down automatically. Since the Vax operator was inundated with terminal users complaining that the system was 'down', he was under some duress to solve the problem and found it difficult to rationalise the source of the fault given that there were a large number of possibilities. The author was able to suggest the solution however, with very little understanding of the details of the system, on the basis of a fault tree produced some weeks earlier. This example suggests that some formalisation of expertise whether manual or automatic is a useful exercise .

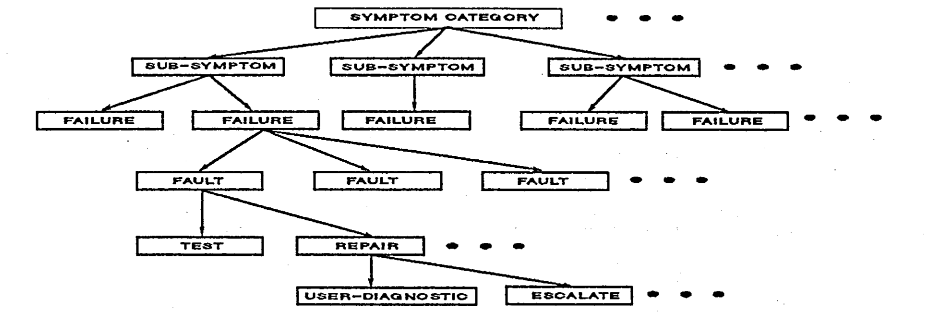


Figure 7.11: Fault Tree Structure for the Helpdesk
(Note: Three dots denote that the branch is a sub-set of a much larger branch)

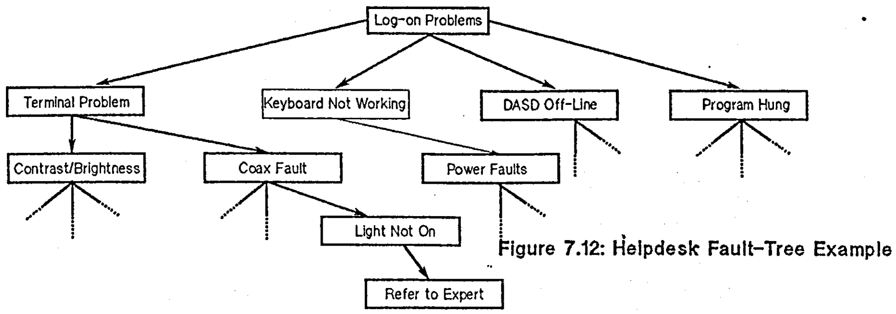


Figure 7.12: Helpdesk Fault-Tree Example

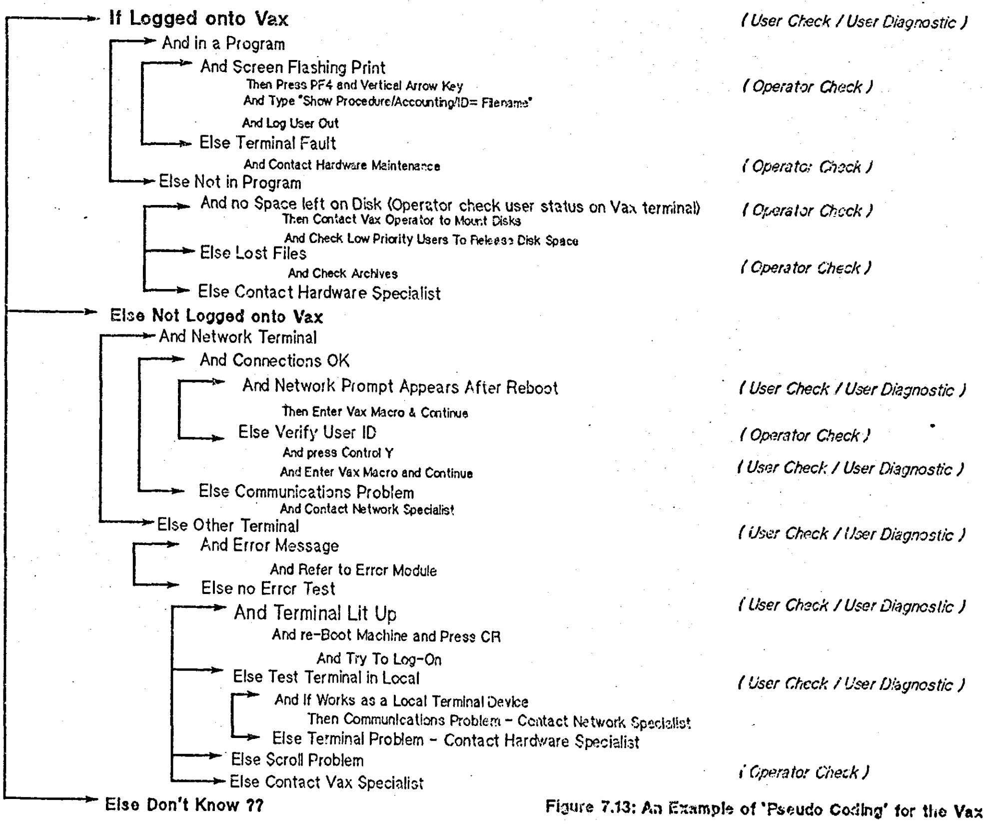
b) **Fault Logging:** Inevitably, the expert could not remember all faults and so it was useful for him to make a record of problems as they arose. The Incident Report Form used during the development of the evaluation prototype (*see Figure V, Appendix IXb.*) proved a useful format by which to record symptoms (as defined by the user), sub-symptoms (deduced by the expert), the root cause derived from these and the subsequent test and repair work. It also provided a convenient means of transferring fault records from the expert and service point at Preston to the author based in Manchester without having to produce fault trees.

c) Observation: There were small areas of expertise which the experts called borderline in that they only just satisfied the criteria of being 'routine' enough to be represented by the Help-desk. It was found that in these cases, the most effective and graphic form of elicitation was through observation and walk-throughs with the expert from which fault-trees could be drafted. Although this approach was successful in defining borderline deep/shallow based reasoning procedures, it was very time-consuming and demanding upon the expert.

d) Secondary Sources: Company codes of practice, vendor manuals, user manuals, training guides and other text-based information proved an important source of procedural knowledge for the Help-desk. This reflected the role of the Help-desk in providing a general service to the user community based upon advice, guide-lines and support for queries which need not be fault related.

7.6.4.5. Knowledge Elicitation and Representation at the Logic Programming Level

From contextual to structural, a third level of representation which was used on occasions was the production of pseudo programming code. This eased the transition from fault tree to knowledge-base programming, particularly when the fault tree was complex. In such cases, it was valuable to provide self-contained declarative statements or 'bundles of code' and work up the fault tree to show how each bundle was related to others. Therefore, this representation differed from the previous two in being bottom up. A sample of pseudo code is shown in Figure 7.13, taken from a medium-level fault tree for the Vax system. The arrows in *Figure 7.13* show how each bundle is nested within another higher level bundle. A consequence of adopting this bottom-up approach is that each bundle of code has no implied context, and therefore is guided and controlled only when a set of guard conditions, defined at a higher (meta) level, were satisfied. This is true only when there is one possible fault condition for any single node (i.e. no multiple causal faults). In the case where more than one fault exists, the new symptoms that it presents are dealt with as an entirely new fault category.



```
If Logged onto Vax                                                              ( User Check / User Diagnostic )
    And in a Program
        And Screen Flashing Print
                Then Press PF4 and Vertical Arrow Key                           ( Operator Check )
                And Type "Show Procedure/Accounting/ID= Filename"
                And Log User Out
        Else Terminal Fault
                And Contact Hardware Maintenance                                ( Operator Check )
    Else Not in Program
        And no Space left on Disk (Operator check user status on Vax terminal)  ( Operator Check )
                Then Contact Vax Operator to Mount Disks
                And Check Low Priority Users To Release Disk Space
        Else Lost Files
                And Check Archives                                              ( Operator Check )
        Else Contact Hardware Specialist
Else Not Logged onto Vax
    And Network Terminal
        And Connections OK
            And Network Prompt Appears After Reboot                             ( User Check / User Diagnostic )
                Then Enter Vax Macro & Continue
            Else Verify User ID                                                 ( Operator Check )
                And press Control Y
                And Enter Vax Macro and Continue                                ( User Check / User Diagnostic )
        Else Communications Problem
                And Contact Network Specialist
    Else Other Terminal                                                         ( User Check / User Diagnostic )
        And Error Message
                And Refer to Error Module
        Else no Error Test
            And Terminal Lit Up                                                 ( User Check / User Diagnostic )
                And re-Boot Machine and Press CR
                    And Try To Log-On
            Else Test Terminal in Local                                         ( User Check / User Diagnostic )
                And If Works as a Local Terminal Device
                    Then Communications Problem - Contact Network Specialist
                Else Terminal Problem - Contact Hardware Specialist
            Else Scroll Problem
            Else Contact Vax Specialist                                         ( Operator Check )
Else Don't Know ??
```

Figure 7.13: An Example of 'Pseudo Coding' for the Vax System

Figure 7.13 also highlights a pattern of decision-making in the help-desk. Structured rules define to the operator a sequence of well defined tests which are asked to the end-user in order to refine the users' understanding of the problem and thereby establish definite symptoms for the faults. In this process, end-users may participate directly by performing actions or repairs to their equipment (e.g. press PF2 and communicate the effect back to the operator); or passively in that the operator by-passes the end-user and uses a terminal console or profiles to find out more about the user's situation. If neither approach is forthcoming, then the Help-desk will instruct the operator, at a certain level, to refer the problem to the specialist directly. When problems are a high priority or just simply too complex, then they are escalated to the expert high up in the fault tree.

The discipline of pre-project analysis has provided a functional specification of the Help-desk outlining logical design needs (user/system interfaces for example); a functional specification defining knowledge context, boundaries and scope (through IDEFo and other processes); and a structural representation defining the types of knowledge used and their interactions (through fault trees and pseudo coding). This information is not only essential in constructing an expert system, but before this, in constraining and guiding the process of selecting an appropriate development tool.

7.7. Tool Evaluation and Selection

Tool selection is the process of matching total development features and requirements of an application against the total product and services capabilities of a tool. In addressing this topic, this section focuses upon four main issues:-

a) Which criteria are important in selecting a tool ?
b) Are there any suitable methodologies or techniques for tool evaluation and selection ?
c) What are the current capability and offerings from vendors in the expert systems market ?
d) How should the above be applied in selecting a tool for the Help-desk ?

The accent in the fist three issues is in providing an evaluative framework for tool selection so that future decisions concerning commitment towards ES development tools may be made by non-specialists in the company. As part of this process, it was necessary that knowledge about the ES market , structure, products and services was gathered and disseminated to key personnel in the company. The fourth issue relates specifically to how the evaluative framework was applied in the selection of an appropriate tool for the Help-desk. As with problem selection in *Chapter 6*, a significant amount of support analysis is necessary during tool selection; much of this is described in detail in Appendix XII.

7.7.1. Approaches towards Tool Selection

Appendix I has highlighted the variety of expert systems tools available to the developer. There are hierarchies of tools according to cost, complexity, functionality (Preece & Gregory: 1988); but essentially these range from artificial intelligence languages which require specialist skills and may need dedicated hardware to operate, to 'shells' which contain all aspects of a full expert system (such as inference engine, user interface, and explanation facilities), but without a knowledge-base. Associated with these tools is a new language and development approach, new vendor names and processes of selection. The purpose of providing an evaluative framework therefore was to allow project management, and other non-specialist functions, to apply an orientation and assessment framework for evaluating which tool, if any, should be used to build an expert system given that all previous stages of pre-project assessment have been undertaken. There have been a multitude of studies that have addressed the evaluation of expert system tools, and these may be classified into three main fields :-

i) Tool specifications: here commentators have provided a catalogue of individual tools and describe the technical settings which define on which occasions each should be used. For example, Harmon(1990) produces a newsletter which categorises tools with respect to their knowledge representation, inference and controlling strategies. This is of limited use in the client company since the framework should be dynamic and offer a strategy for selection rather than be affiliated to a particular product which is likely to become obsolete within a few years. Furthermore, this approach centres on tool capabilities only and adopts criteria such as method of representation for example, as a means of distinguishing between other tools rather than indicating precisely how this factor should influence the selection process.

ii) Classifications: A second approach is to identify problem or application categories from which tool attributes are inherited . For example, Gevarter (1987) provides a matrix which defines the suitability of backward and forward reasoning, choice of representation and uncertainty technique and many other features for each generic application category (diagnosis, planning design etc). This approach is useful if the level of understanding about expert systems is minimal but can be inaccurate and simplistic (Martin *et al*:1988). Moreover, as with tool specifications, the level of analysis focuses upon tool capabilities rather than problem requirements.

iii) Assessment Methods and Tools: This approach adopts a systematic comparison of tool features against problem characteristics through scaling and other assessment techniques. These methods allow features matching between specific and case sensitive problems and also help to define tool needs rather than be constrained to tools which are only identified as being feasible. Examples include the structured correspondence scheme of Markus *et al* (1987) and a contingent feature matching approach of van Koppen (1988). These methods therefore require a detailed specification of the application problem from which tool feature requirements are defined.

Although current assessments methods provide a useful basis for an evaluation framework, they are incomplete in the sense that they overlook other constraints besides the technical difficulties at the interface between the problem and the tool. For instance Martin *et al* defines a brief set of practical issues under which the selection process is constrained. These include size and complexity of the application; experience of the user in ES techniques; and budget and timescale limits. The theme of project and organisational factors influencing selection is extended by Leininger (1987) and Bryant(1988), albeit in informal terms, to consider 'external' factors such as the company's relationship with the vendor, and vendor service and support issues. Finally, Rothenberg *et al.*'s Selection Methodology (1987) is an attempt to combine objective techniques of problem features assessment with the discipline of defining project characteristics and tool capabilities. However, as a formal method it is highly prescriptive, complex and likely to be difficult to understand particularly by a non specialist.

7.7.2. Providing an Evaluative Framework for Tool Selection

A review of selection approaches above, reveals two main short-comings. First, a tendency to limit tool evaluation to a technical issue concentrating mainly on matching knowledge-base development needs with tool capabilities. Although this is important, and is covered well in the literature, it is not complete as an evaluation unless wider project and organisational constraints are incorporated into the formula. Second, a recent study of experiences in developing expert systems (Bramer:1990) highlighted a basic lack of support in applying selection techniques which suggests that more attention should be paid towards the processes of evaluation as well simply outlining requirements. In doing this, care should be taken to provide a framework which is simple to apply and flexible enough to adapt to organisational settings, rather than end up being a 'bureaucratic methodology' (Hirschheim *et al*:1987).

On the basis of these findings, a basic evaluative framework was developed for use in the company. This is shown in Figure 7.14. and is made up of four parallel assessments which take into account 'scaling factors' (Preece & Gregory: 1988), i.e. changes to the system such as future enhancements, modifications and maintenance, over 'n' years of use. These are:-

a) Problem Features (Box 3 in Figure 7.14): the framework assumes that all earlier analyses have been undertaken and that therefore the proposed design is specified in functional and logical terms with an understanding of knowledge representation and control requirements and interface features.

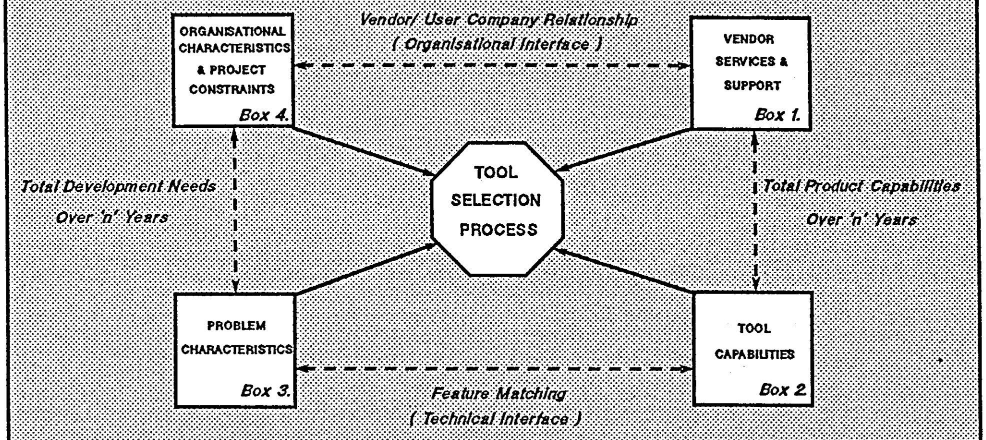


Figure 7.14: A Framework for Tool Evaluation & Selection

b) Project Constraints (Box 4): much of this information is available from the development suitability check-list and other previous assessments, which when brought together, provides a detailed specification of the problem. Together boxes 3 & 4 in *Figure 7.14* comprise total development needs for the proposed application.

c) Vendor Services and Support (Box 1): It is as necessary to evaluate the service and support provided by a vendor as much as the capabilities of the product itself. According to current and future organisational needs, it may be necessary that training, consultancy, maintenance and programming support and other services are provided by the vendor. The interface between *Boxes 1 & 4* highlights the importance of the relationship and mechanisms of transfer between ES vendor and user organisation in determining the quality of match between total development needs and total product capabilities. This is discussed in a wider context in the next chapter.

d) Tool Capabilities (Box 2): These may be defined in general terms such as ease of use and user-friendliness to highly specific measures of performance and capability in describing knowledge base, inferencing and interface functions. Not all features provided by the tool will be required by the problem however. The interaction between *Boxes 1 & 3* represents total product capabilities. A requirement of the framework is that there is knowledge of the market structure and range of products made available at an organisational level (rather than restricted to the authors own personal 'knowledge base'). To achieve this, it was necessary to undertake a survey of ES vendors as a formal and documented process of knowledge gathering.

7.7.3. Defining Total Product Capabilities Through a Vendor Survey

At the time of the study, little was known about the different types of expert systems, their operations and suitability to specific types of manufacturing problems. Studies that were available, besides being prohibitively expensive, were inaccurate and incomplete and suffered the blight of exaggerated capabilities and hype which characterised the market at that time (Ovum:1988a). Rather than simply provide a directory of names and addresses, a more rigourous and analytical investigation of the vendor market was necessary. For this and the above reasons, it was decided that a necessary task prior to using the selection framework should be to carry out a full survey of all expert systems producers and suppliers in the UK. Since the study was carried out, vendors have been contacted for correctness of information and therefore the findings remain valid for 1990.

7.7.3.1. Survey Objectives

The basic aim of the survey was to provide a comprehensive guide to ES software which was commercially available for use in the construction of expert systems, using a range of criteria consistent with the goals of the selection framework. More specifically, it was hoped that the survey would provide details on the vendor organisation; service and support levels; details of known applications and targeted users; the product's capabilities; and the product's development operating environments.

7.7.3.2. Survey Design and Methods

An exhibit of the vendor questionnaire and definitions and assumptions made in its design are given in Part A and Part B respectively of *Appendix XII*. A number of questions used in the questionnaire are based on other studies. Table 7.12 identifies the authors and their area of contribution.

Table 7.12: Reference Material User in the Vendor Questionnaire

Issue	**Markus** et. al (87)	**Holsapple** et al (86)	**van Koppen** (1988)	**Leininger** 1987	**Reichgelt** et al(86)	**Gervarter** (1887)
Problem/Task Characteristics	*				*	
Tool Inferencing, Control & Representation Features	*	*	*		*	
Interface & Interaction Tool Features		*	*			
Tool Hardware & Operating Characteristics	*		*			
Organisational Factors & Project Constraints				*		
Vendor Service & Support Capabilities				*		
Application Suitability						*

The questionnaire is designed to provide details on the total product capabilities of vendors and is made up of five sections:-

A: General Vendor Details: this section asks for background information about the vendor company (how long it has been in operation, its customer-base and whether it is a producer or supplier of tools) and its product (cost, description of tool). It also attempts to target the precise use for the product in terms of end-users (is the tool aimed at naive users or professional systems developers for example) and specialist application domains (for instance the tool may be designed specifically for planning applications) or sectors of industry (for example telecommunications).

B. Operating Environment: this section identifies the hardware requirements of the tool (whether it requires a Mainframe or dedicated workstation or whether it can operate on a personal computer for instance) and any size restraints such as minimum memory or disk requirements. It also defines the class of software tool and for shells and Artificial Intelligence(A.I.) languages establishes details about the source and language code used.

C. Development Characteristics: this section deals with the technical performance characteristics of the vendor tool in terms of how it represents knowledge (the more forms of representation the tool has the more flexible it is presumed to be); how it is able to control the direction of knowledge (through forward and backward chaining for example); how it deals with uncertainty; and the facilities it has to communicate with the systems developer, other hardware, and ultimately with the end-user.

D: Application Profile: a method of evaluating tool requirements described earlier was by classification according to generic application characteristics. Although this approach was has a number of shortfalls , it is useful in providing a first estimation as to the suitability of the tool in relation to the problem. This section looks at where a tool's customer base is concentrated: if a tool has a large number of applications in scheduling , estimating and stock control for example then it is likely to be suitable for most generic planning type applications.

E: Service and Support: this section attempts to identify the range of service and support activities undertaken by the vendor organisation from training and consultancy, to customerisation and maintenance. It also addresses the long term needs of the application in terms of scaling (is the system modular and does it have its own built-in programming for example), security and licence costs.

Over the period March-July, 1988, 72 questionnaires were issued to market producers and suppliers of expert systems in the UK. (This does not include academic and other research projects). A full listing of these organisations is given in Section F of Appendix XII. Of this number, 58 companies returned their questionnaires, indicating a response rate in excess of 80%. Following up on those companies which had failed to respond, it transpired that four companies had ceased to operate and three had amalgamated with larger organisations that had made the decision to diversify into supplying expert systems software. These movements reflect a general trend towards consolidation in the market (Ovum: 1988a).

The decision was taken to restrict the investigation to non-mainframe expert systems because the level of commitment towards these systems was beyond the scope of the study. Furthermore, a more cost effective route towards knowledge-based integration would be to use Personal Computers and Workstations which were able to be networked and communicate with company databases but be developed in a separate environment. On the basis of this premise, the sample was reduced to 48 companies and all proceeding calculations use this figure.

The questionnaire was aimed at the technical manager and/or support engineers in the company, as many of the questions required a detailed understanding of the product.

7.7.3.4. Survey Results

The survey results were used in two ways:-

a) At a market level: in order to define the state of the ES market in terms of size and shape, customer base, products and services. A detailed and full analysis of the survey results at a this level is given in Sections C,D & E of Appendix 12. However, in that these are structural issues, they are most relevant to discussions on how ES technology is presently transferred from the market place to user organisations, albeit from a vendor perspective, and will therefore be referred to frequently in the next chapter on technology transfer.

b) As a technical reference: for use in assessing the suitability of individual tools for specific expert system proposals. The survey was used in this way in selecting an appropriate tool for the Help-desk . It is also hoped that the criteria used in the survey will help to place future tools (and accompanying promotional literature) in a suitable perspective.

7.7.4. Tool Selection for the Help-desk

In applying the evaluation framework, the sequence of tasks was first to summarise problem characteristics; then apply project constraints from which a selection of 'feasible' tools may be made from the questionnaires. A short-list of tools is made on the basis of additional desirable features and the extent to which the vendor organisation satisfies service and support needs. The final selection is therefore made on the basis of meeting all four criteria defined by the boxes in Figure 7.14. There is also an additional factor which is the personal preference of the intended developer. This may be a particular penchant towards a tool with certain interface features or even design of the screens, which makes the programmer more comfortable with the product (Martin *et al*).

7.7.4.1. Problem Characteristics of the Help-desk

The following features were required in the Help-desk:-

i) Representation Structure: From an earlier analysis, it was established that the knowledge in the Help-desk is mainly heuristic and procedural both of which may be represented by a classification tree structure. Rule-based representations are therefore adequate with no need for model based structures. Rule induction may be applied since nearly all the knowledge is shallow and structured. However it is unlikely that induction alone would meet future Help-desk needs and therefore any tool would require a combination of representations.

ii) Uncertainty: There is no requirement to represent uncertainty in the knowledge base explicitly through certainty factors or fuzzy reasoning for example. The nature of the faults and queries in the Help-desk domain and the way that faults are collected are as well defined and structured statements. As well as being unnecessary, there are a number of misgivings about the validity of uncertainty techniques (Towriss:1988). However where there is implied uncertainty, 'the fault is *probably* due to human error' for example, this may be addressed through the ordering of rules and the actual wording of the textual output to the operator by which a sequence of checks (with the most likely first) is performed.

iii) Search Strategy : The Help-desk makes use of both forward and backward chaining, the use of which will depend at what stage a consultation has reached. At the beginning of a troubleshooting session, the operator will ask a number of general 'meta' questions which will determine the broad area in which the fault lies prior to more detailed reasoning. This is a forward-chaining function in which the system works from the basis of the facts it has learned from the user to derive conclusions. In subsequent stages, the Help-desk has to guide the operator quickly towards a detailed explanation, and will need to use the answers to questions already put to determine the next line of reasoning. This is a backward chaining function in which the system is trying to prove goals by matching the lines of reasoning from series of rules.

iv) Users' Interface: The effectiveness of interfaces will determine how effectively the operator is able to present digital knowledge in an analogue way. The interface between the operator and the user community is an organisational issue which is discussed in the evaluation to this chapter. The interface between the operator and the Help-desk requires a number of necessary and desirable features. Necessary features include context sensitive help and explanation features so that the operator upon requesting help from the system receives advice which is directly pertinent to the situation and not 'canned advice' (Morris:1987). To allow the operator to run tests and volunteer information, the Help-desk will also require a 'what-if' facility and menu driven features at all times. A mixed initiative system will not only speed up the heuristic reduction process but also allow the operator to develop with the system. A smooth interface is also required between the Help-desk and information sources such as the user and system profiles. In terms of desirable features, the Help-desk would benefit from word search facilities (for example if an end-user provides a key word this may be used to limit the area of the knowledge base); graphical displays and colours and browsing of the knowledge base. Finally, the ability to save, recall and change consultations would increase the flexibility of the operator to return to or defer consultations.

v) System Interfaces: Although the Help-desk may operate as a stand-alone system, its efficiency will be increased by linkages to other software. In the short-term all that is required is a database facility which can store and manipulate records of users and systems, log faults and store consultations. However future enhancements may require more substantial interfacing to provide an automatic fault logging system or Mainframe status monitoring, in which case a built-in programming language is desirable.

vi) Development Features: in constructing the knowledge base of the Help-desk, essential features include rule trace ,cross-referencing and consistency checking which allow the developer to fully understand the validity of a rule in terms of other rules. The programmer will also require variable and rule dictionaries, on screen help facilities and word searching to improve the productivity of rule construction. In terms of services and support, a full technical manual is required and technical advice such as a help-line or programming service.

vii) Maintenance and Future Needs: Future Help-desk maintenance needs will be addressed by the PC specialist. Since the project would be handed over to the expert when the author left, it was essential that the tool was easily understood. In order to minimise the complexity of maintenance itself , it is important that the Help-desk provides assistance by storing all procedural information such as user/system profiles, saved consultations and other import-export files, and the log of faults outside of the knowledge base in databases (DBASE, Clipper or Lotus for example).

As the Help-desk role increases, it is conceivable that more difficult problems will be approached in which case more complex (deep) reasoning capabilities will be required. A useful attribute for the Help-desk is that tool the chosen is 'modular' in that capabilities may be improved incrementally by adding-on new functions. An alternative is to provide an interfacing language, although more complex, so that the additional work and requirements may be achieved by using other software.

7.7.4.2.Project Constraints for the Help-desk

The choice of tool for the Help-desk is constrained by a number of 'first-project' criteria which were identified in the last chapter (*Section 6.8.1.*). These include that the system should be developed in less than a year; that the software is low cost (a ceiling of £2000 was later defined); that it could be learned and applied in a short period of time by a non ES programmer; and that the software could be developed and delivered on a standard 386 personal computer. Where problem features provided a technical filter by which to short-list feasible tools, these and other project constraints provided an organisational filter as the next section shows.

7.7.4.3. Short-Listing Tools

It was evident from the above analysis that there was no reason for choosing any more complicated representations than a rule-based one for the Help-desk. Moreover, in terms of technical features and project constraints, the only type of tool which satisfied both these were expert system shells. Seven shells were short-listed from the questionnaire and their relative strengths and weaknesses are summarised in *Table 7.13*. The criteria used in Table 7.13 look at both development needs and end-user (in this case the operator's) needs as well as general problem characteristics. Individual tool capabilities are measured in terms of their ability to satisfy these requirements using qualitative terms such as 'poor' or 'very good' for example. All the tools outlined in Table 7.13 broadly met problem characteristic requirements and offered acceptable levels of service and support. Therefore the determining factors in the selection process were mainly organisational and project constraints. On this basis, a final selection for the Help-desk was made with Crystal for the following reasons: -

i) Value: Crystal was one of the cheaper shells and yet had a number of facilities, as Table 7.13 shows, which were not available on the more expensive tools.

Factor \ Tool	CRYSTAL	XI-PLUS	SAVOIR	PERSONAL CONSULTANT	XPERTRULE[2]	TOP-ONE[1]	LEONARDO
Helpdesk Development Needs							
Meets Cost /Resource Constraints	YES	YES	PARTIALLY	NO	YES	PARTIALLY	PARTIALLY
Ease of Use	HIGH	AVERAGE	POOR	AVERAGE	AVERAGE	AVERAGE	GOOD
Short Learning Curve	YES	PARTIALLY	NO	NO	YES	NO	YES
Embedded Capabilities	YES	YES	YES	YES	YES	YES	YES
Satisfies Representation Needs	YES	YES	NO	YES	YES	YES	YES
Satisfy Inferencing Needs	YES	YES	YES	YES	YES	YES	YES
Satisfy Interfacing Needs	YES	YES	YES	YES	YES	YES	YES
Meets Helpdesk Vendor Support Needs	YES	YES	YES	YES	NO	PARTIALLY	YES
Meets Helpdesk Vendor Service Needs	YES	YES	YES	YES	YES	YES	YES
Meets future helpdesk needs	YES	YES	YES	YES	NO	YES	YES
Meets Future Helpdesk Maintenance Needs	NO	NO	NO	YES	NO	YES	YES
Helpdesk Operation Needs							
Provide What-if Analysis	YES	YES	YES	NO	NO	YES	YES
Word Search	YES	YES	NO	NO	NO	YES	YES
Menu Driven	YES	YES	YES	YES	YES	YES	YES
Context Sensitive Help/ Explanations	YES	YES	YES	YES	YES	YES	YES
Quality of Interface Features[3]	GOOD	AVERAGE	AVERAGE	GOOD	POOR	GOOD	HIGH
Short Operator Training	YES	NO	YES	YES	YES	NO	NO
Vendor Service & Support In Use	GOOD	GOOD	GOOD	AVERAGE	PARTIALLY	GOOD	GOOD
Record of Applications In Diagnosis	YES	YES	YES	YES	YES	NO	YES

Notes: [1] Top-one Operates on a Mainframe System

[2] Xpert-rule is a Rule Induction Shell

[3] Vendor Demonstration Software was Available to Assess User Interface Features

Table 7.13: A Summary of Total Shell Capabilities

ii) Ease of Use: The shell's support facilities (editing, debugging and built-in commands) and its simple logic allowed rules to be developed in a matter of days. This was important because development was restricted to less than a year, the learning curve had to be short. However, the skills of knowledge-base structuring and efficient programming took some months to actually master.

iii)Proven Record: Crystal has a record of over 10,000 users with many documented examples of use in trouble-shooting and diagnostic domains.

iv) Service: The vendors of Crystal provided good support facilities , training, consultancy and a user group. There were also more specialised services functions such as knowledge engineering and maintenance advice groups which could be drawn upon in future.

v) System Interfaces: Crystal provided in-built interfaces to Dbase and Lotus 123 and ASCII databases. It also had an in-built interface programming facility using 'C' which meant that it could be linked to other software systems.

vi) User Interfaces: The experts and end-users liked the interface on Crystal particularly.

A more detailed appreciation of Crystal is provided in Appendix XIIIa., with the emphasis being upon the tool's programming techniques and facilities as an preface to its use in the development of the Help-desk .

7.8. Help-desk Knowledge-Base Programming & Development

Since the Help-desk was intended to solve a large range of high level problems, the structure of the knowledge base was broad and shallow. It was clear that it would be both undesirable (especially in maintenance terms) and impractical to pull all rules together within a single knowledge base, and so a structure of modularised knowledge-bases was devised as shown in Figure 7.15. Each module or knowledge base is independent and deals with a specific area of the overall problem. However they may communicate with each other using ASCII files and Import/Export commands.

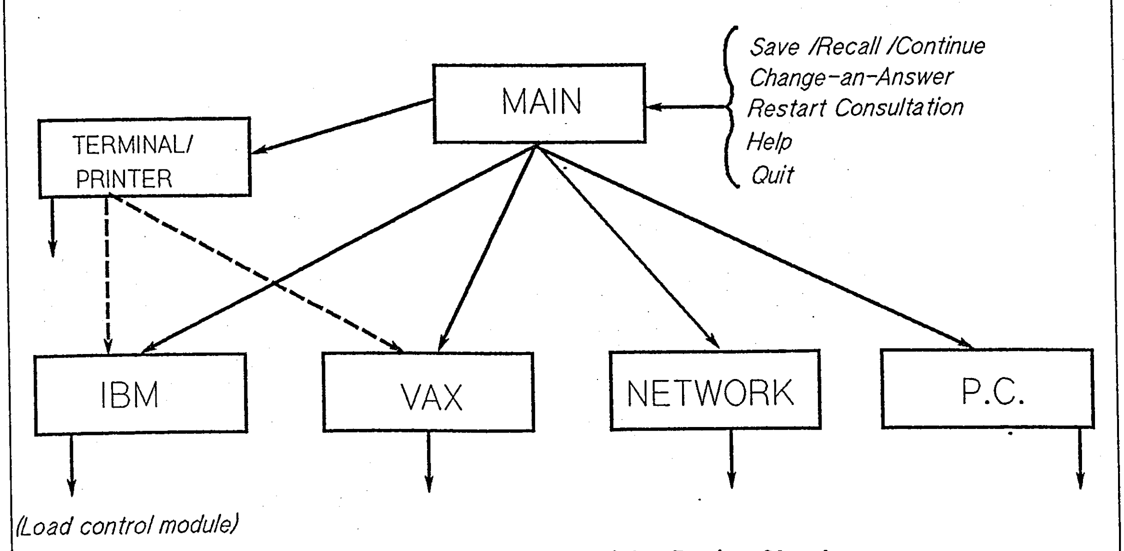


Figure 7.15 : Help-desk Modular Design Structure

Each knowledge base is also linked to a top-level control module (called 'Main' *in Figure 7.15*) which regulates the interaction between knowledge bases and provides general facilities such as saving and recalling consultations, changing answers, system information and help facilities. This control module contains a menu and each menu option loads the appropriate knowledge-base. Each knowledge-base then loads the control module at the end of a consultation in order to return the user to the main control menu. This structure lent itself well to the incremental Help-desk design in which decision-trees or pseudo codes were programed into Crystal bottom-up. This allowed single rules to be tested and verified before they were attached to higher level rule structures. Having verified the knowledge using decision-trees, programing itself using Crystal was straightforward. The build commands in Crystal allowed rules to be generated quickly (see Appendix XIIIa) and the in-built user functions such as Menu Screens, YES/No questions, Print and Screen Display Functions, Macros and View Forms provided a satisfactory interface within a short period of time. The use of these functions together with a number of other features which were developed specifically for the Help-desk are discussed in Appendix XIIIb. This appendix shows examples of Help-desk screens as well as describing sections of Crystal programing code.

One of the uncertainties in developing the Help-desk was in determining its eventual size. IDEFo was useful because it provided an indication of the scope of the Help-desk but was of little use in determining its depth. Similarly, decision-trees were developed incrementally (i.e. draft out a decision-tree, ask the expert to verify it and then program a small section of the knowledge-base) and so the total size of the Help-desk's knowledge-bases could not be predicted with accuracy. The 'size' of a knowledge base is frequently measured by the number of rules that it holds (Hayes-Roth *et al:* 1984). In the case of Crystal however, this was of little meaning because the software's notion of a rule was to count the number of rule-commands rather than separate and unique production rules. Therefore Crystal indicated the Help-desk to be 840 rules in size using 260 variables. This would be classed as a very large system using recent figures (MI:1989). However taking account of repeated rules and rationalising commands into single rules, this figure is reduced to about 240 rules (this constitutes a small-medium sized expert system using the same source.

As well as differences in measuring rules between shells, a second issue is the efficiency in which rules are programed. Rather than cluster rule commands within a long string of sub-rules associated with the Master Rule (see Appendix XIIIa for an account of Master Rules), it was considered easier for successors to the author to understand the program if rules were limited to one per build screen where possible. This was inefficient in machine terms (an expert system that can be logically divided into 20 sets of 50 rules is more efficient than a 1000 rule system) but proved to be more comprehensible to the expert who was to update the expert system in future. Furthermore, since the Help-desk would be operating on a high performance personal computer, the differences in efficiency in terms of time and memory requirements were of secondary importance.

7.9 Interim Evaluation

Almost one third through the allocated development period for the Help-desk, an interim evaluation took place with the purpose of comparing actual progress and performance against time and resource estimates and design criteria specified prior to development. From this it was hoped to identify basic design problems early on and measure the extent of 'drift' from planned estimates and thereby re-define or re-scope the project accordingly. Clearly such an approach would not have been possible if performance targets had not been set and a design specified beforehand.

For the Help-desk, this was essential since a number of problems did emerge during development:-

7.9.1. Calibration Problems in Scaling

Due to an absence of analytical tools by which to plan development requirements, as Section 7.5.1. has mentioned, time and resource estimates particularly were calibrated on the basis of experiences in developing the computer hardware evaluation prototype described in Appendix IXb. A central assumption behind this approach is that the scale-up of values would be proportional (e.g if the prototype is one-fifth the size and knowledge acquisition took one month, then knowledge acquisition for the full system will take five months). However experience in developing the Help-desk has shown this not to be the case because as size increases, so does the total complexity of the system. For instance, it became more difficult to link areas of knowledge as the knowledge-base increased in size despite the benefits of using decision-trees and modular knowledge base structures.

7.9.2. Knowledge Verification Demands

Figure 7.16 measures the deviation of actual durations with time estimates provided in Figure 7.7 (using the average figure of upper and lower case time estimates as a datum). A positive value of deviation suggests a time delay whilst a negative value indicates that the actual time taken was less than that planned. It can be seen clearly from Figure 17.6. that the most significant source of time delay was during knowledge representation. The formalisation of interview notes into decision-trees, and on occasions pseudo code, although a very effective means of representation, was a slow and laborious process especially in that each tree required verification prior to encoding.

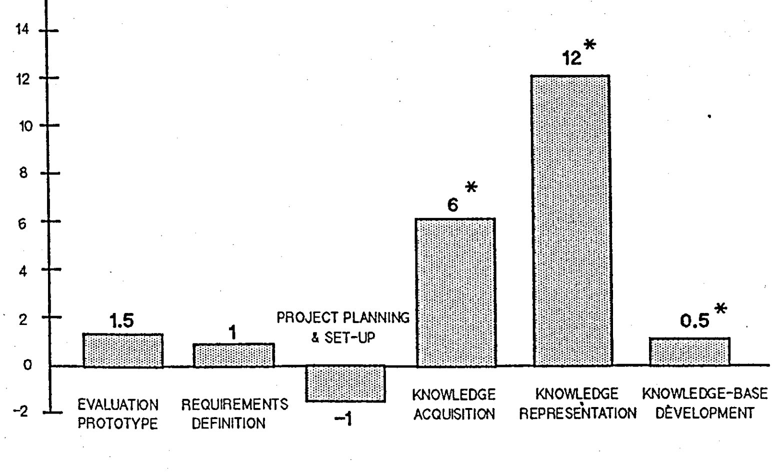


Figure 7.16: Discrepancy Between Estimated & Actual Duration for Phase 1

*Pro-rata time deviation for ALL tasks based upon actual deviations for 1/3rd of planned knowledge base

Associated with this misjudgement were a number of erroneous assumptions:-

a) Because the knowledge was 'shallow' and well structured, the expert would find it straightforward to articulate and represent it in an intermediate format. In fact the experts had a number of difficulties in expressing and formalising their knowledge

b) It was assumed that the domains of expertise were discrete and interacted in a well defined way. However, it became evident at the interface between VAX, IBM and Network domains particularly there were 'grey' areas of uncertainty and some conflicts in problem-solving approach between experts, each with their own view on the problem. Furthermore, problem-solving at the interfaces had causal and deep-level elements as well as shallow, making production rule representations less appropriate and more difficult to apply.

c) Because there was a substantial amount of secondary information, the process of knowledge acquisition would be reduced. Although this was true, the extent to which time was saved was less than expected because this information still required structuring in order to make it meaningful to the operator.

d) Although the boundaries of knowledge in the domain were known, some types of knowledge took longer to acquire than others with a cumulative effect upon the time taken to verify this knowledge. For example the Incident Report Form used by the experts to log faults as they arose, identified new faults which were relevant to the Help-desk almost a months after knowledge elicitation was planned to finish.

7.9.3. Failure to Define Operator Role

There were unforeseen difficulties in representing 'analogue' knowledge because the expert with the operator and end-users had to decide how information should be presented as well as what should be included. For the operator role particularly this required greater technical and interpretative skills than anticipated. The Help-desk attempted to resolve this problem by attempting to define the dialogue between the operator and end-user explicitly and incorporate as much of what the expert did at this level into the Help-desk as was possible. Thus for example, almost six weeks were spent in devising levels of user competence and programing these into the design of the interface.

Rather than attempt to define unambiguous and explicit forms of dialogue, from the complexity and time involved to produce even modest results it became apparent that effort would be better directed at improving the capabilities of the operator rather than the machine. Through improved training and tuition, the operator could be afforded more discretion in discerning the category of fault, competence of the user and nature of consultation required and would therefore utilise the Help-desk more effectively. This is not a sentimental notion of 'human-centredness' but a pragmatic response to an overly complex technology situation which could be simplified by investing more in the potential skills and flexibility of the operator. In practice though, for organisational and political reasons, it was necessary that the operator role was taken up by a semi-specialist and that the role of the Help-desk would subsequently change also.

7.9.4. Failure to Target End-Users

The Help-desk was expected to address a widely diverging set of user needs and capabilities which made a single system of operations difficult. In the end, despite profiling the user, the Help-desk provided a service which was directed at a mean level of competence and was therefore of little use with user with more complex problems. The mean complexity of problems differed greatly between domains. For instance VAX and IBM end-users varied significantly in competence making it difficult to focus the Help-desk upon any particular group. By contrast, the range of complexity for PC users was small and moreover, the bulk of end-user queries were frequently recurring and based upon shallow knowledge with a high procedural information content.

7.9.5. Organisational Problems

The Help-desk concept required more significant departmental, cross-site and organisational changes (cultural and physical) than expected. The changes required were especially acute in the VAX and IBM areas again where a traditional network of relationships had been established and would be disrupted by the Help-desk. Although possible conflicts of this sort had been identified during the selection process, the degree of dissonance could not be qualified until actual development.

These problems also have associated strengths and weaknesses; the necessity is that these experiences should be used positively to re-define the Help-desk. The direction of these changes is contingent upon the considered size and importance of each problem when measured against original objectives and performance criteria defined at the beginning of development. The next section outlines the main changes that were made to the Help-desk as a result of this evaluative process.

7.10 Redefining the Scope and Organisational Role for the Help-desk

On the basis of the above analysis, a second phase of development was defined in which a number of changes were made to the scope and size of the Help-desk. This 'phase two prototype' is defined in more detail in Appendix XIV and is also shown clearly from Figure 7.1. However, since much of the work for this revised Help-desk had already been completed and moreover, a great deal of the design information acquired during pre-project assessment phases remains valid, this section describes only the differences between the two in terms of scope, organisational role, size and operations.

7.10.1. Office Systems Help-desk

The PC domain proved to be the most simple to represent and verify; the most homogeneous in terms of range of user needs; and received the most favourable response from potential end-users and experts. For this reason, and given limited time available (5 months), it was decided to reduce the scope of the Help-desk to PC systems only. This provided the opportunity to enlarge upon the detail of troubleshooting in personal computing and also related fields such as dedicated word processors, typewriters, printers all of which are known collectively as 'Office Systems' (this term will be used henceforth). The source of expertise remained the same but was called upon to provide more knowledge about personal computer operations and hardware; printers and peripherals. There was also a more substantive role for secondary sources of information, particularly vendor manuals and documentation.

7.10.2. Help-desk Role and Layout

Instead of one Help-desk, the service relationship described in Section 7.3.4. was extended to three sources so that there were two personal computers operating the Help-desk at the Manchester site and one at the Preston site. This layout is outlined in Figure 7.17 and its operations are defined by Figure 1 of Appendix XIVa. Two of the Help-desks are used in a similar capacity to the Phase 1 Help-desk in that their purpose is to filter a broad-band of top-level problems from the Office Systems (OS) expert; the expert only being referred to when a problem is too complex or too difficult for the end-user to resolve the fault. The difference is that these Help-desks are manned by trainee OS specialists rather than non-experts. Thus the nature of the consultation between the Help-desk and operator became advisory rather than directive. It also meant that the Help-desk could assume, in terms of dialogue, terminology and general design of the interface, that the operator new much more about the domain than previously. The result was a more direct and simple design of Help-desk interface. An unexpected benefit from using the Help-desk in this way was that the trainees at both sides found it a useful training device since the expert's knowledge was systematically defined and explained in a logical process of decision-making; indeed it was actively supported by the expert in this role.

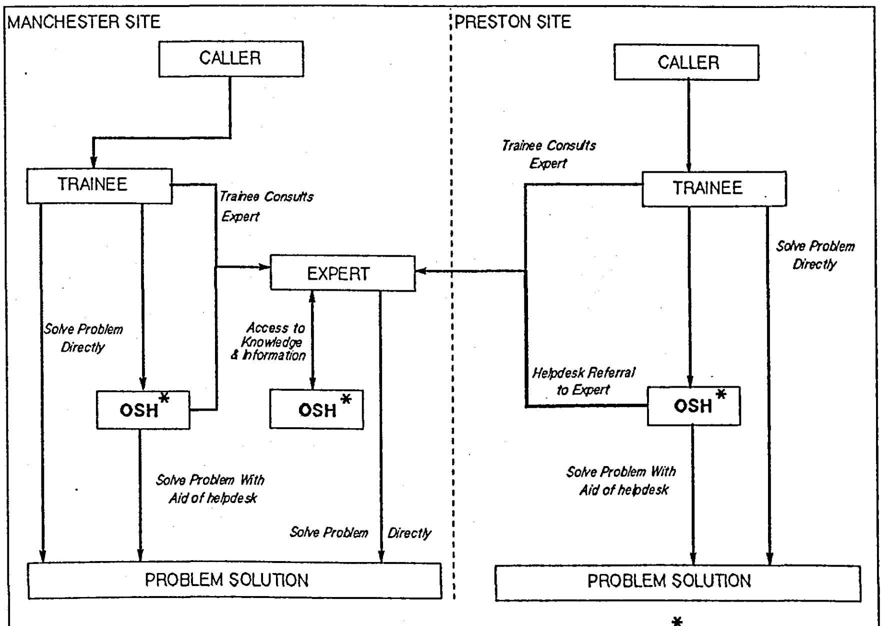
A second role for the Help-desk as Figure 1 in Appendix XIVa shows, is as a source of reference to the expert. Unlike the Vax and IBM modules, the OS expert required information on a vast number of possible machine configurations and their specific requirements, and the level of support afforded to each configuration, together with a good understanding of the technical specification of each machine. The Help-desk was useful therefore as a 'procedural animator' in providing central and rapid access to information. The expert also made use of the Help-desk to archive faults in equipment which although occasional could take the expert many hours to solve in future.

7.10.3. Help-desk Structure & Operations

The structure of the revised Help-desk is outlined in Figures 1-6 of Appendix XIVa, and the operation of the Help-desk is described in Appendix XIVb. Many of the features and techniques used in Phase 1 were applied to the OSH, albeit in a modified form.

7.10.4 Justification

The justification for the OSH was broadly similar to the phase one prototype and based on the opportunity costs of the expert undertaking more valuable work; but also, when called upon, to improve the productivity of expert troubleshooting by providing rapid access to information and supporting knowledge required to make a diagnosis. The cost structure was also broadly the same with the difference that the Help-desk would not be manned full-time; the costs of knowledge acquisition would be less; the hardware costs and software (licensing) costs would be increased; and with a diminished call rate, the operating costs would be less. The effects these changes and others have upon the cost justification of the OSH, taking account of the 'sunk' costs lost to the Phase 1 prototype, are discussed in the evaluation to this chapter. An important decision in these revised costings is the extent to which Phase 1 costs should be carried over into the revised Help-desk or whether this cost should be absorbed by the organisation as a learning experience or similar contingency. The effects on inclusion and omission upon payback are subsequently analysed.



*Office Systems Helpdesk

Figure 7.17: Layout of the Office Systems helpdesk

7.11. Design of User Interfaces

Many of the experiences gained during development of the Phase 1 prototype were used in correctly planning and refining estimates for the second phase with the effect that the system was built well within the five months allocated. One of the most important activities in this process was the design of the user interfaces. It was also one of the more problematic in that screens and menus could always be improved further according to operator preferences and therefore it was important to set a time limit in which to devise 'acceptable' rather than optimum interface designs. Nevertheless, it was found that by making even small improvements to syntax or the nature of the consultation through changes in screen presentation, the operator's response time (whether trainee or expert) to answer a query would increase also.

Following basic training, the operator very quickly began to explore the limits of Crystal's interface capabilities and within these limits define how best information could be presented. This was a dual process of correctness in which the expert defined the order of execution in information; and comprehension whereby the trainees expressed explanation and help facility, syntax and presentation needs. In the latter case, information had to be presented in a way which would communicate to the trainee *what* the fault was and also provide guide-lines which would allow the trainee to describe to end-users *how* to go about resolving the problem. At this design stage, the Help-desk was not run as a fully operational system for the reason highlighted by Milne (1990) that without program testing (see *7.12.1.* below), the operator would become disenchanted by bugs and system crashes with possible negative effects upon future involvement.

The re-definition of interface needs for the OSH was simple by comparison to other interface specifications (e.g. Morris 1987) but adequate for the task and acceptable for the operators. Most of the consultations centred on the use of simple YES/NO and Menu-driven functions which the operators preferred to more complex displays such as windows and input fields. Other preferences and requirements included:-

a) minimal operator input to perform a desired troubleshooting action. Menus offered a number of advantages over operator input such as reduced typing; less keystrokes required; and that the menu option formed part of the question text and therefore helped to explain the situation more fully.

b) a hierarchy of menus and forms. An initial problem experienced by all the operators was navigation through the system. This was greatly eased by pull-down menus and help screens indicating in which knowledge base the operator was located and at what level. Transparency could be improved further by the use of graphic decision-tree mapping of the knowledge-base (equivalent to file tree directories in personal computing) which are available from the vendors of Crystal.

c) terse and succinct answers with explanations available by pressing the F1 function key. Occasionally additional menu options were used such as 'Don't Know' or 'None of These' for where the operator was confused, or where it was possible for the problem situation not to correspond to any of the menu options provided. For YES/NO questions, the F1 key operated as a default 'don't know' option.

d) simple and uniform screen interfaces (e.g. no colours). On occasions graphics were used especially in the PC communications module for wiring and connector diagrams although without proper graphics facilities in Crystal, this was a laborious process.

The expert had an established approach and manner which was well received by end-users. Clearly, it was not possible to imbue the Help-desk with a 'persona'

(Ostberg: 1988), but it was possible to reflect the experts style of diagnosing faults and therefore retaining a consistency of approach with end-users whilst simultaneously addressing the trainees' specific interface needs.

7.12 Testing and Validating the Knowledge Base

Once the Help-desk was finished in terms of constructing the knowledge-base and interfaces, testing and validation sought to demonstrate that the system was complete, correct and has achieved its intended performance levels. This process marked a final stage of implementation in which the help-desk was relocated from a development machine under trial conditions onto a delivery machine and in a fully operational setting. There were many stages of testing but all may be subsumed within one of three broad categories: testing for programming errors which was undertaken by the system developer alone; testing for reasoning as a dual procedure between expert and developer; and testing for usability acceptability which was undertaken principally by the trainees and end-users representatives.

7.12.1 Testing for Machine Correctness

Early experience at testing the system during the development of the evaluation prototype proved that system crashes (in which the Help-desk operator was thrown out of the system), other programming bugs and screen problems discredited the system even during test phase with experts. Therefore the first stage of testing for the OSH was undertaken by the developer alone to attempt to remove bugs, refine programming and improve screen features. This identified holes in menu options; failure to take the user back to main menus; key commands not corresponding to actions specified; import and export problems when calling up profiles; and failure to save files. Clearly not all bugs could be identified and therefore a log of change requests was created which allowed the OSH operator to document problems (each screen had a reference number for this reason) which were later corrected by the developer.The method of checking for machine correctness was to systematically look at all the text displays and follow all lines of reasoning. A second test criteria in this category is the efficiency of the Help-desk in providing a solution. There were a number of occasions for example when an exhaustive sequence of yes/no consultations could have been replaced by a series of menu options; and where direct user input would have identified the fault source more effectively. There were also areas of 'untidy' programming which were corrected, thus improving the average run-time efficiency of consultations.

7.12.2. Testing for Reasoning

Although testing is defined as a postscript to development (Ross and Quinlin:1989), it was hoped that some aspects of testing the OSH could be .on-going as part of the process of development. This was certainly possible in two areas: testing reasoning; and testing acceptability through participation in interface design. In the former, the use of decision trees as an intermediate representation allowed testing and verification of reasoning with the expert prior to encoding. When a tree was tested, the expert verified each of the tree's branches to see if an accurate and efficient pathway had been used to reach a valid conclusion. The expert found this a more transparent and productive means of testing his reasoning that attempting to interpret the coded reasoning in Crystal. Although an intermediate representation improved reasoning correctness and consistency, it was still necessary to test the Help-desk system for other possible failures such as programming logic and information presentation. Hollnagel(1989) provided a number of criteria for this purpose:-

a) Reliability: Hollnagel defines reliability in relation to expert systems testing as 'the degree of unexplained variance in the results' (p385). If the results from the same system consultation vary a lot then the system has a low reliability. Hollnagel observes that for off-line, rule-based diagnostic systems, reliability is usually high. However, variations will arise according to the accuracy of the end-user input, the successful recognition and assessment of the fault by the trainee and the successful interpretation by the end-user of the required actions to take to resolve the problems. These human interface problems were of particular threat to the effectiveness of the Help-desk, especially since consultations were to take place over the telephone rather than at the location of the fault. Reliability was improved by making consultations as unambiguous as possible and providing context sensitive explanation facilities.

b) Testing Diagnostic Accuracy: In that the rules were highly structured and uncertainty reasoning was not required, the accuracy of problem solving was high. In the capacity of 'adviser' to the OSH however, accuracy was not critical since erroneous information to end-users would cause more of an inconvenience than a crisis. In measuring accuracy, it was important to ensure that the information provided to the end-user was as accurate as the information provided to the operator from the Help-desk.

c) Testing for Completeness: Completeness is defined as the extent to which reasoning is applied to work through a given problem. In the case of the OSH, completeness was low; it would advise the trainees down to a particular level of complexity defined by the expert after which the system would instruct the trainee to escalate the problem directly to the expert. This could also be used implicitly to filter high priority problems directly to the expert. Completeness was therefore defined by what was acceptable to the expert.

7.12.3. Testing for Usability and Acceptability

Eason et al. (1987) identifies usability and acceptability as the most important factors in determining the success of ES projects. Hollnagel defines usability as 'the ease with which the user can apply the system according to its purpose irrespective of the level of experience and proficiency of the user' (p389). The Phase 1 prototype attempted to achieve a high level of usability by incorporating complex interfacing facilities so that unskilled operators could use the Help-desk. In the OSH, it was accepted that a high level of usability was unattainable and the system was re-designed to address more knowledgeable users in an advisory capacity. This simplified the testing process significantly.

There are two levels of acceptability testing for the Help-desk. At a Help-desk operator level, it was ensured that the trainees were happy with the interface features; since they were involved directly in the development, testing of interface features was itself a part of the design process. Main changes at this level were syntax and description modifications, improvements on help and explanation facilities and improvements in the transparency of the system. As with the Phase 1 prototype, the OSH would be used primarily by the trainees as a filter for the expert and therefore the interface was biased towards their specific needs. It was convenient that the expert used the OSH in a different way to the trainees, often concentrating at different parts of the knowledge-base, so that his needs could be customised also without generating potential interface design conflicts.

The second level of acceptability was that the trouble-shooting service, in terms of both technical quality and delivery, was acceptable to the user community. This could not be determined until the Help-desk service function had established itself in the organisation. A performance standard for acceptability was set however in that end-users would not notice a change in the service level despite being attended to by a person other than the expert. Where end-users were referred to the expert, it was hoped that the speed of trouble-shooting would improve.

7.13. Help-desk Documentation & Training

There were two levels of training and documentation reflecting the needs of the Help-desk operators and the expert earmarked as future developer:-

7.13.1. Systems Use: It took only a day to for the expert and two days for the trainees to gain a solid appreciation of the operations of the Help-desk and this was reinforced by trial-runs during the testing phase. Both trainees and expert found the Help-desk straightforward to use and required no assistance or clarification from the developer after a few days of use with the system's help and explanation facilities proving adequate. A user manual was written documenting all key operations , machine delivery requirements, and systems mapping, similar in format to the operations description provided in Appendix XIVa. As well as essential procedures, the manual also provided 'useful hints' on making effective use of the system and getting out of trouble.

7.13.2. Systems Development: The expert was also chosen as the individual responsible for attending to future maintenance requirements and possible enhancements to the system. The expert had a full understanding of the problem situation, of the Help-desk structure and operations and therefore was ideally suited to the task. Regular sessions were held with the expert describing the rudiments of Crystal and the programming structure of the Help-desk. It was also agreed that in addition to the vendor Help-desk, the author would be available to offer advice for the proceeding six months should this be required. A full listing of the decision-tree structure was passed onto the expert as well as relevant parts of the IDEFo model and Crystal programming manual.

7. 14 Office Systems Help-desk Evaluation

An evaluation of the Help-desk is made with respect to four main factors: an evaluation of Help-desk performance and operating experiences; an evaluation of the human and organisational effects of the Help-desk; an evaluation of knowledge-base maintenance requirements; and a financial evaluation of the Help-desk using discounted cost/benefit analysis.

7.14.1. Performance Evaluation of the OSH

In terms of achieving the original objective, the OSH was implemented and is being used to solve between 60% and 70% of the expert's problems. However its role is different than that specified in the original Phase 1 plan and moreover, its function after one year of operations has also changed further. The Help-desk is necessarily evaluated from two points in time therefore: immediately after the Help-desk's implementation when fault logs and performance measures were taken for the first month; and one year later when general comments on its use are made.

a) Frequency of use:

The actual Help-desk loading was very similar to that predicted in the original estimates with an average of 36 calls at both the Preston and Manchester sites although with wild fluctuations: for instance one day there were only six calls in total whilst on another occasion there were over eighty calls. Furthermore, the call loading was not even through the day but peaked very strongly between 9.30 - 11.00am and 2.00-3.00pm. This unpredictability was originally compensated by incorporating a degree of redundancy in the expert's daily schedule. A consequence of the Help-desk is that the Office Systems (OS) support service could respond to these fluctuations much better, especially since at peak demand there would be two service lines(one at each site) and a third (the expert) available if necessary.

The duration of call estimates of 5½ minutes proved a little to high and settled down to about 5 minutes. This was to be expected since the OSH was manned by more skilled staff. However there was an unanticipated problem in this in that the trainees were perhaps too eager to resolve the problem themselves rather than escalate the problem directly to the expert. This explains why there is only a moderate reduction in the call duration period despite the skill differential between the original operator and the trainee. It also accounted for why the trainees requested more enhancements to the Help-desk than the expert.

The most frequently consulted and most effective part of the Help-desk was the printer problems section reflecting the greatest number of end-user queries in this areas. The least used section of the Help-desk was the System Profiles although this is because it was used almost exclusively by the expert rather than the trainees(although a trainee was responsible for updating the profile records). A second area of the Help-desk seldom used was the PC hardware troubleshooting section; again this was because it tended to be of use as a personal reference and information source for the expert than as part of the mainstream of end-user requests. In the sections of the Help-desk that were used occasionally, it was more critical that the interface features were of a high quality because the operator would require more assistance in using the system.

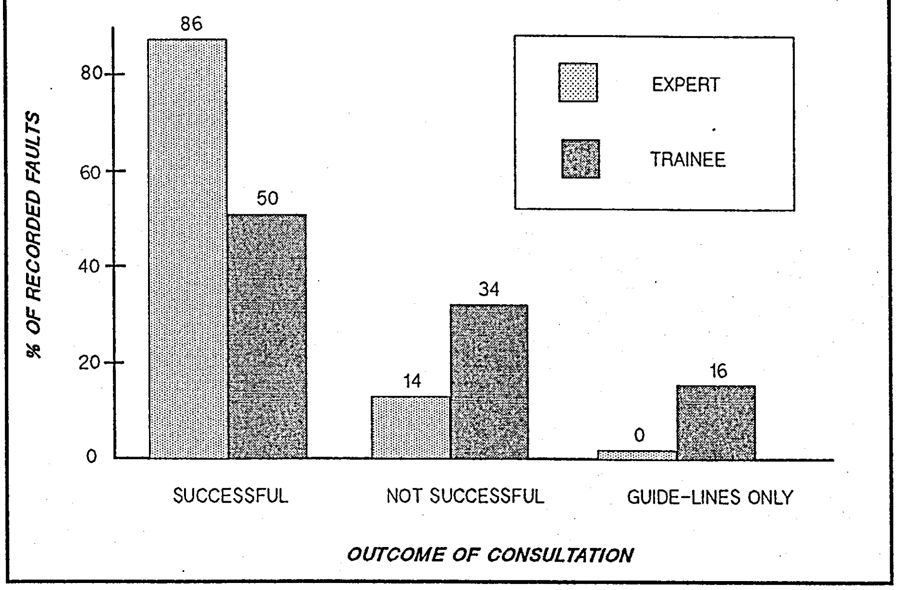
b) How successful was the Help-desk ?

A first question related to this issues is under what circumstances the Help-desk failed. A failure is defined from a *trainee perspective* as being unsuccessful in using the Help-desk to determine a fault cause and provide subsequent diagnostic advice. A consultation is a direct success if the trainee is able to resolve the end-user's query immediately or provide guide-lines to a solution which may take longer (e.g screen is faulty; contact Operations and order a new one). In both cases a query is resolved without intervention or reference to the expert. *Figure 7.18* provides a summary of faults logged for the first month of operation (not all faults were logged but it is unlikely that the results will change significantly). It shows that the trainee was able to resolve 66% of queries directly or as guide-lines. The success rate was probably greater than this since the trainee occasionally wished to verify a decision with the expert or seek limited assistance on how to 'process' the information provided by the Help-desk in terms of guide-lines to the end-user. Of those calls answered by the trainee that were not successful, 56% were referred to the expert (which from a computing service perspective is just as valuable as a direct success). The remaining 44% represents the true weaknesses of the Help-desk either because the problem was outside the intended domain or because a new problem was encountered. In both cases the trainee would often waste time using the Help-desk (or deciding himself whether he could solve the problem) before contacting the expert. This highlights the importance and difficulty of defining explicitly the boundaries and limitations of the domain covered by the Help-desk. It also reflects a natural desire by the trainee to solve the problem using the wrong mechanisms (i.e. to flog a dead expert system!).

Although the expert consulted the Help-desk much less frequently, Figure 7.18 shows that the expert was more successful in using the Help-desk to resolve a problem. However, the expert mainly used the Help-desk as an information base and therefore would use it more selectively. The expert did also ask that the occasional, very rare problem was encoded in the Help-desk. These problems were not especially difficult to program into the system but could save the expert great amounts of time in future. For instance the expert spent nearly two days (over a period of time) attempting to resolve a printer problem which upon finding the solution took just thirty seconds! This role for skills archiving became increasingly important for the Help-desk. When the Help-desk was of no use to the expert (14%

of the time), it usually meant that outside support (from OS vendors or maintenance organisations) was required.

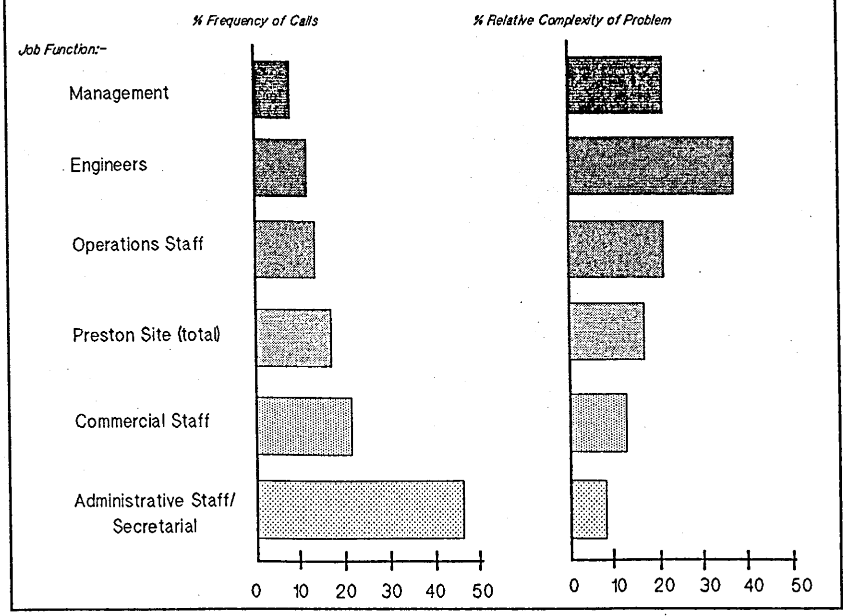
Figure 7.18: Helpdesk Fault Log Analysis (For the First Month Only)



c) End-user Profile

If the Help-desk was used almost exclusively by a certain class of end user, then there is good reason to customise it wholly around their needs. Figure 7.19 shows that the bulk of calls to the Help-desk were by administrative and secretarial staff; however not enough to justify a Help-desk dedicated to this level . Figure 7.19 also shows that the least frequent callers -managers and engineers - also placed the most demands upon the Help-desk in terms of the relative complexity of their queries. Indeed the frequency of calls was almost inversely proportional to call complexity with novice users such as administrative staff tending to ask more, simple questions and engineers who tended to be competent users asking relatively few but much more complex questions.

Figure 7.19: Helpdesk Evaluation: Frequency of Calls & Relative Complexity



d) The Organisational Value of the Help-desk

At the Preston site, the use of Office Systems and personal computers in particular was more recent than at Manchester and so potentially the Help-desk had a more important and valuable role. Prior to the use of the Help-desk, the computer department at Preston, with skills of only Mainframe related systems, would endeavour to respond to user queries but frequently escalate the problem to the expert at Manchester. Although this relationship was maintained, the dependency upon this support was diminished. Since the help-desk was developed principally at the Manchester site, there also was some bias towards the representation of computer equipment for this site. Although some modules were enlarged to accommodate computer equipment that was specific to the Preston site, the help-desk retained a strong Manchester identity. More changes are necessary in order to redress this organisational imbalance.

e) Evaluation from an Expert's Viewpoint

The expert found the help-desk straightforward to use but for the use of some screens and key functions which were subsequently modified. In terms of the structure and problem-solving logic of the system he was satisfied to the extent that he supported its use as a training aid. The expert expressed the benefits of the system in three ways: first, it reduced the level of disturbance and disruption to problems which justified his participation; second, it archived 'nasty' problems which were important but very occasional and therefore frequently forgotten; and third, it provided a quick reference to equipment information, configuration details and system records which improved the productivity of decision-making.

The extent to which the help-desk improved decision-making productivity was difficult to quantify. However on the basis of qualitative estimates by the expert a few conclusions could be drawn. Foremost is that the expert reported that 40% of faults took half the time to resolve because of the help-desk. From previous estimates, it was assumed that the average duration of a call by the expert took 4½ minutes. Thus if the expert answers 30% of total calls = **11 calls/day**. And 40% of these (which is 4.4 calls)calls take ½ of 4½ minutes = *2.25 minutes*. While 60% (which is 6.6 calls) take the normal *4½ minutes*. Then,

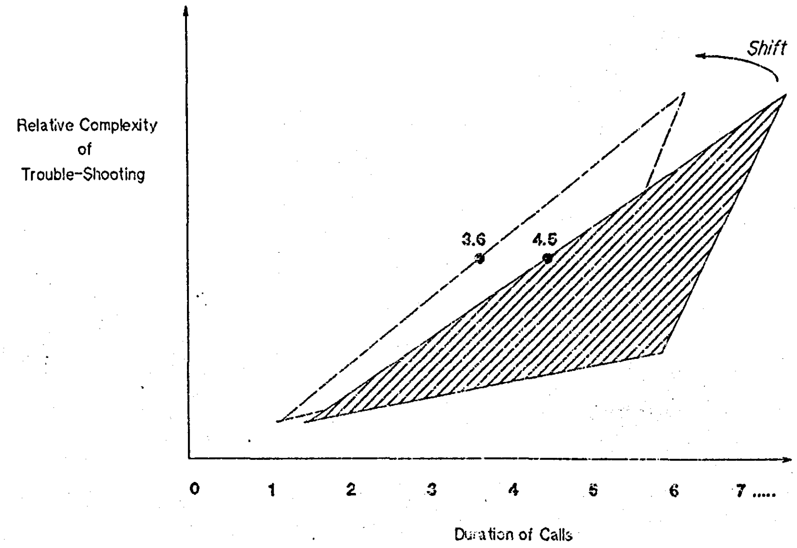
$$\text{New average duration} = \frac{(4.4 \times 2.25) + (6.6 \times 4.5)}{11} = \mathbf{3.6\ minutes}$$

Therefore the effect of the Help-desk was to reduce this duration to 3.6 minutes as Figure 7.20 shows. The help-desk however, did nothing to make trouble-shooting easier and, as Figure 7.20 highlights, the complexity of the task remains the same as do the expert skills required to resolve the problem.

f) Evaluation form a Trainees' Viewpoint

The trainees expressed the benefits of the help-desk in a number of ways: it allowed them to begin trouble-shooting earlier than had they followed the traditional induction path; it provided a systematic and well defined reference to the expert's knowledge with explanations of logic and was therefore valuable as a training aid; and it helped to establish a rapport with the User community. However, it took much longer for the trainees to use the help-desk effectively than anticipated. This was because in addition to understanding the technical operations of the help-desk and navigate through the system, they also had to become familiar with the organisational role that they had adopted (interacting with end-users on the phone and related skills).In this, it was essential to stress the fallibility of the help-desk to the trainees and importance of referring calls to the expert in fuzzy problem areas such as software-hardware interfacing.

Figure 7.20 The Effects of the Office Systems Helpdesk Upon Expert Response Times



g) Evaluation of the System One Year Later

It is seldom that operational experiences in using an expert system are evaluated (Mumford:1989), probably for the reason that many fail to reach the operations phase (Bramer:1990). However, close association with the organisation after the project provided the opportunity to evaluate the help-desk almost a year after implementation. The Help-desk is successful in that it remains in operational use: however the nature and extent of its use is different from that intended for the following reasons:-

i) the 'trainees' became more and more competent in Office Systems trouble-shooting with the consequence that the help-desk was used less to solve routine problems and more in the way used by the expert as an information reference. As the dependence upon the system reduced, the system was neglected and was not maintained. However this situation could reverse following the departure of one of the 'trainees' and subsequent replacement by a new trainee.

ii) the expert continues to use the help-desk as a reference point and places more value on the system as a mechanism for archiving nasty problems. However he has much less confidence in the system's performance generally because of a failure to regularly maintain the system. He is also disenchanted with the Systems Profile as a means of storing equipment details especially since low cost inventory systems are available on the market.

iii) Both the expert and original trainee are 'irritated' by the same consultation format. This suggests that it is not only the knowledge base that requires updating nor aspects of the interface, but the whole nature of the help-desk interaction, at an organisational level as well as a personal level, needs to be re-assessed. It also indicates a reluctance by the expert to perform maintenance work.

In its current use, the help-desk could in no way be justified as a viable project and continues to be used simply because it is available rather than because it is needed. This possibly downplays its usefulness as a 'training aid and knowledge-bank' but both these are unexpected benefits rather than pre-determined. The value of the help-desk diminished for a number of organisational and personal reasons, but mainly because of a failure to maintain the knowledge base and change the help-desk role as the structure of the problem changed. It is likely that the help-desk will continue to be used in this way for the next year hence unless major enhancements, described in Section 7.16. are implemented.

7.15. Evaluation of Human & Organisational Effects

Two conclusions may be drawn from previous discussions: firstly, that the human effects of the help-desk were largely positive; and secondly, the organisational effects were negligible. This section brings these observations together within a more analytical framework through a discussion of skills and organisational impacts, legal and ethical implications and other socio-technical issues.

7.15.1. The Skills Impact of the Help-desk

By introducing the help-desk in the organisation, the content of individuals' jobs will invariably change in some way. After all, the main reason for innovating is to improve, refine or alter the nature of the task. These changes may be adverse; for example Brodner (1990) refers to a 'neo-Taylorism' in which expert systems may be used to automate expertise and downgrade human cognitive skills. Conversely, Huxor(1988) is much more optimistic about the role and impact of expert systems and views their use as a medium for skill enhancement. These however represent extremes and the view taken in developing the help-desk was that it could have an indeterminate effect upon skills by potentially down-grading, upgrading or having no effect upon the skills of the individual depending upon the design and implementation of the system. As Clegg(1989) thus notes, the issue then is to ensure positive and planned skills impacts through attention to user needs (expert, operator, end-users) in advance and during design, implementation and evaluation phases.

For the trainees, the help-desk provided the opportunity to enhance their skills in a practical and non-destructive learning environment. This accelerated the process of integration into the organisation and allowed the trainee to operate independently of the expert other than for the most difficult problems. There was no suggestion of dependence upon the help-desk as a master-slave relationship; indeed, the help-desk was used more as adviser to the trainee. The value of the help-desk was thus to improve the skills *effectiveness* of the trainee.

For the expert, the help-desk improved skills *efficiency* by the provision of better decision-making support facilities. The help-desk was used as an information base and later as a knowledge-base for occasional problems. However, the skill content of decision-making remained the same.

7.15.2. Organisational Impact of the Help-desk

This section asks the question "to what extent did the help-desk change the work organisation? " The help-desk had potential for significant organisational redesign; not solely within the computer department but in the structure of relationships with the user community and the rest of the organisation. This approach is consistent with Klein et al.'s experiences in which an expert system designed for personnel selection was used as an opportunity to re-structure the organisation in a 'pro-active' way (1988). However as a first project using new technology such an aggressive approach was judged to be unworkable for reasons of political acceptability and other institutional constraints. Therefore the accent in developing the help-desk was that it would reinforce established communication channels and improve the existing service network with the user community. To this end, this is precisely what the help-desk succeeded in doing. The help-desk make trouble-shooting expertise more accessible and available more quickly than before and therefore encouraged end-users to make use of the service rather than seek alternative sources. A major problem in the company was in inexperienced end-users attempting to solve problems themselves or seeking advice from colleagues. Such 'dabbling' often had the effect of exacerbating the problem rather than solving it: the help-desk helped to make these informal networks redundant by proving that the formal channels were the most effective. In this sense, the old organisational patterns were reproduced and re-affirmed in the new technology.

7.15.3. Legal and Ethical Issues

Legal and ethical issues are used interdependently to refer to moral and judicial responsibilities associated in using the help-desk and the political, cultural and personal ramifications of not considering such issues as ownership and responsibility. Put very simply these issues question who will be to blame should the help-desk provide erroneous information. For critical system areas, e.g. process control and on-line trouble-shooting, mis-diagnosis is costly and possibly dangerous and therefore accountability for the system is essential. Although the effects of mis-diagnosis in the help-desk were only likely to cause inconvenience and delay, these issues nevertheless remained important.

Insofar as the help-desk is an internal organisational development rather than a commercial product, 'legality' relates to the allocation of responsibility for a fault rather than legal action, although if there is a genuine and provable fault in the vendor software then this may be the case. Clarke(1988) identified that accountability should not be made at the development level but at the management level. This is because users of the help-desk rightly so can invoke the 'piano-player's defence as Clarke calls it which states that 'it was my job to apply the tool not to understand it'. The developer may also adopted the same position by claiming that his role was just to capture the know-how of the expert and translate it into a machine processable format. Clarke even provides a pretext of non responsibility for the expert in that " ...because the form in which the knowledge was expressed only vaguely resembled their knowledge, and they could not be expected to understand and audit the particular formalism used by the knowledge engineer." (p15). For these reasons, although the development environment was much more constructive and co-operative than the formal image painted by Clarke, accountability for faults was attributed to the project manager. This ensured that the manager checked the training of help-desk users and that the help-desk itself was valid and complete prior to systems release. During operations however, accountability shifted to the expert who was responsible for maintaining the system during its planned lifetime and was the only person in the company trained to perform the duty. However if the company decided that the help-desk should be enlarged and the expert was reluctant to participate, Clarke suggests that the company would have no intellectual property right in the trouble-shooting knowledge of the expert. Although this was not an issue for the help-desk it is useful to point out that expert co-operation is not only necessary for practical reasons, it is also important for legal reasons.

7.16. An Evaluation of Maintenance Requirements

Earlier analysis has suggested that maintenance proved to be a major constraint in the operational effectiveness of the help-desk. This section looks in closer detail at maintenance requirements over the first month of operations to evaluate if maintenance problems encountered later in the systems life could have been predicted. It also assesses the long-term changes required to restore the help-desk to its original value to the company and also make maintenance itself more straightforward.

7.16.1. Analysis of Maintenance Needs over the 1st Month of Use

Over the first month, maintenance requests from the trainees and expert were collated by the author and analysed. The analysis provided some indication of the direction of change and maintenance requirements of the help-desk and the results are shown in Figure 21. Maintenance work was divided into five broad categories as follows:-

a) Missing Logic: this was where the expert identified inconsistencies in the reasoning process such as incorrect assumptions and misinterpretation of the significance of information inputs. This was by far the most complex form of maintenance and often required the complete re-writing of rules from the decision-tree rather than amendments to the existing rule base. Figure 21a. shows the distribution of maintenance work in terms of effort and number of requests. It can be seen that although missing logic made up less than ten percent of the maintenance requests, the actual maintenance effort (in terms of time spent doing the work) was 20 percent of the total. Surprisingly a number of missing logic requests were supplied by the trainees; however these usually were requests to enlarge on the reasoning of the expert who tended to make 'leaps' between, what appeared to the trainee as, unrelated rules-of-thumb. Figure 7.21b. .shows the distribution of missing logic problems according to a classification of problem domains. It can be seen from this that over half of missing logic problems occur in the PC hardware section. This section was the least well understood and the most difficult to bound because of the difficulties in separating hardware problems from the causal effects of the software on which its operations were dependent. For example, a frequently recurring hard disk problem was not because of hardware failure but because of a bug in the operating system software. In encoding trouble-shooting routines, there were a number of instances where the effects of the software interface, as a determinant of the problem, were not taken into consideration. The maintenance decision was therefore either to simplify trouble-shooting further by referring the trainee to the expert straight away or attempt to insert the missing logic more for the benefit of the expert than end-users.

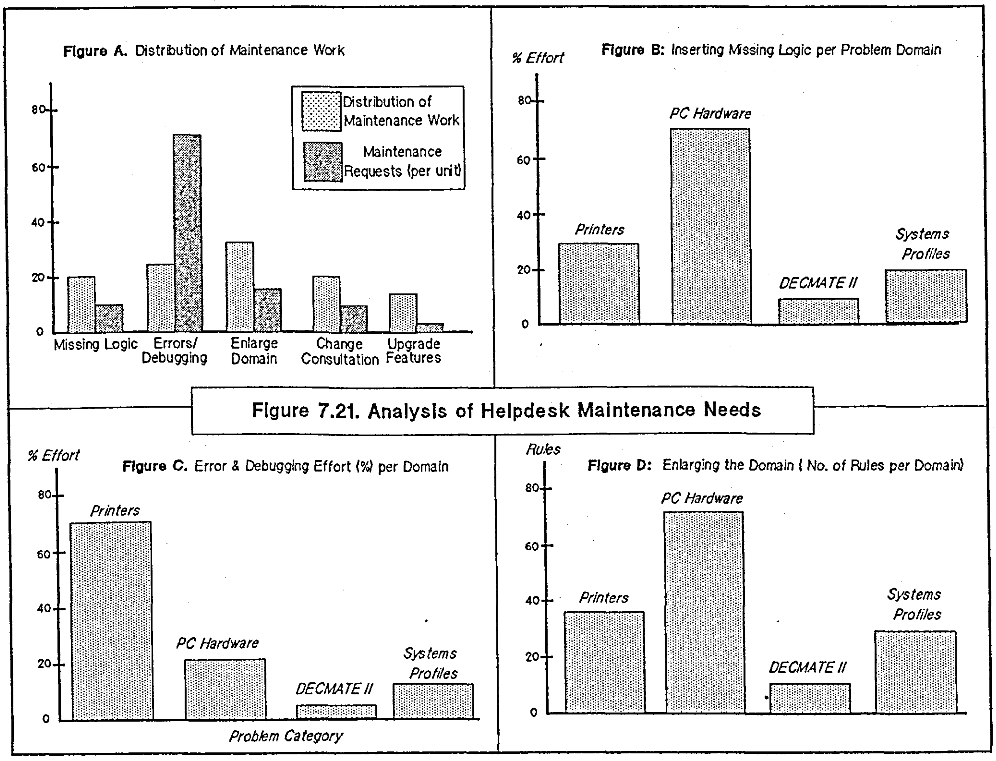


Figure 7.21. Analysis of Helpdesk Maintenance Needs

b) Errors/De-bugging: these are mainly programming faults which the developer failed to identify during testing. The main types of error were system crashes, failure to import records and recall files, incorrect use of menu fields and syntax errors. These problems were the most commonly reported and yet the most simple to rectify, as Figure 7.21a. shows. A breakdown of error and debugging effort in Figure 7.21c., indicates that the majority of faults occur in the printer domain. This however reflects the size of the knowledge-base rather than the peculiarities of the domain itself.

c) Enlarging the Domain: as well as making improvements to current knowledge-bases, some maintenance work concentrated on enlarging them. In the printer sections, new printers were added and the detail of the System Profiles was increased. The greatest maintenance work in this category however is shown by Figure 7.21d. to be in the PC hardware sections. In the first month of operations, a new PC communications module was added as well as enhancements to keyboard, monitor and operating system diagnostics. A significant amount of this knowledge however was procedural and could be gathered from secondary sources thus accelerating the process of enlargement.

d) Change aspects of the Consultation: there were a number of areas of the help-desk interface which help-desk operators had difficulty in understanding. Requests were made to simplify save and recall procedures; improve screen presentation in some areas and re-order menu options so that the most frequently selected option was closer to the cursor mark. The trainees also requested further explanation facilities in the printer section and greater access to other levels of the knowledge base. As knowledge of the domain crystallized, what was previously considered uncertain changed to become certain and routine by the application of rules-of-thumb. This required that not only new knowledge was added to the help-desk, but that the nature of the consultation was modified - as a problem becomes more familiar to a person, less information and fewer steps of reasoning are required in order to recognise the problem and take subsequent actions. In practice this meant that fewer menus were required with a greater role for direct input, and questions were made more concise.

e) Upgrade Features: there were only three upgrade requests concerned with improving the performance of the help-desk by enhancing the system profile facility; breaking the hardware knowledge-base into smaller modules; and re-structuring the printer and interface modules. Figure 7.21a. shows that this form of maintenance is highly demanding of re-development .

7.16.2. Future Enhancements to the Help-desk

The future role of the help-desk was contingent upon organisational developments. An important criterion during application selection therefore was continuity: that the project would appeal to a large section of the organisation who would appreciate the value of the system and, furthermore, that the application could grow and be improved as understanding in the company increased. For the help-desk to retain its value in the organisation, there were a number of possible long-term enhancements :-

a) PC Inventory System: The help-desk may be linked to an on-line user and systems database which would allow browsing and sorting of records and the automatic configuration of knowledge bases according to the hardware set-up of the end-user. There are a number of Office System inventory systems available which could be linked to Crystal using 'C' software hooks. An alternative is to develop an in-house system on *DBaseIII plus*. If the latter is chosen, then it could be incorporated with the Helpdesk's profile facility directly.

b) Fault-Logging System: It is possible to save fault consultations on the Help-desk for later referral and analysis. This is achieved by assigning a date and time-coded filename to a particular consultation. This facility is rather cumbersome and an automated fault-logging system is more desirable. This could be a possible future enhancement to the system: the fault-logging system could be written as an embedded database linked to the help-desk. The implication with this proposal and the inventory system identified above, is towards a shared or fully integrated database. It is expected that this database would require three man-months of development work using DBASEIII, and one man-month linking the database to the Help-desk.

c) PC Support Substitute: Although the Help-desk is presently used as an expert/trainee aid, it could also be adopted as a means of disseminating support knowledge to end-users of Office Systems. To manage these changes would require reorganisation of screens and the facility to use higher-level (more simplistic) menus and explanation functions. It would also necessitate a higher level of problem-solving: where experts presently decide upon the problem category intuitively, non-expert users would require a 'meta-support' level which would systematically identify the problem category. This work would take many months to complete however and would require major cultural and organisational changes which are unlikely to prove acceptable. On a more practical footing, end-users may not have access to a second machine on which to run the Help-desk; or may be unable to use the help-desk effectively, in which case it may be quicker for the user to consult the expert directly rather than attempt self-diagnosis or repair.

d) The Addition of Software Troubleshooting: The help-desk is presently biased towards the diagnostics of hardware faults. However a useful complement would be to include advisory and troubleshooting procedures for application software such as Displaywrite 4 and Lotus 123 which were used in the company. The knowledge of software is more uncertain and complex than for hardware and would require more sophisticated programming techniques and extensive consultation with experts. To encode knowledge-bases for high demand software would probably take three-man months of work. No additional resources would be required.

e) Enlarge the Scope of the Help-desk: The objective of the Phase 1 project was to develop a system which would encapsulate the knowledge of all areas of computing, including Mainframe systems and network communications. This proved too ambitious given the time restrictions. However, although incomplete, a substantial amount of work was carried out and many of the knowledge bases could be made useful with some re-development. This would involve six months work on three main areas: Vax Hardware (2 man-months); IBM Mainframe (3 man-months); and Networks (1 man-month).

7.16.3. Methods of Facilitating Maintenance

In order to provide predictable changes to an expert system, Compton *et al.* stress the importance of establishing a maintenance discipline (1989). To prevent *ad-hoc* and unstructured changes to the help-desk, the trainees' systems were made run-time so that if modifications were required, they were logged formally and passed on to the expert at Manchester. The expert retained the development copy of the help-desk. Moreover, procedures were set-up which ensured that the trainees always made use of the latest version of the help-desk.

Although maintenance was eased by the modular structure of the help-desk, the very nature of rule-based logic was not suited to the process (Kennet *et al*:1989). Walters & Nielsen (1989) refer to the problems of 'procedural fever' in rule-based programming where volumes of rules are required to express small amounts of knowledge. This was compounded in the Help-desk by a very broad and shallow structure rather than one which was narrow and well defined. Crystal itself had poor maintenance facilities and it was very difficult to trace rules and define their interactions with other rules other than by using the Rule and Variable Dictionaries (*see Appendix XIVb*) with the Rule Break Facility. This was adequate in editing rules but not in defining their effects and relationships. Because of these shortfalls, there was a greater reliance upon the intermediate representation (i.e. decision trees) in order to map out these relationships and make the knowledge base transparent. However, as a manual technique the process was often slow and there were drawing inconsistencies. Better still would be to make use of automated graphics tools which map out the knowledge-base as a rule decision-tree and improve the quality and speed of maintenance (see for example ICAT:1986). However given the cost restrictions this option was unattainable.

7.16.4. Observations on Maintenance

The findings from the help-desk provide scope for wider discussions on maintenance requirements in three broad areas:

a). Levels of Maintenance: The help-desk underwent three distinct phases of maintenance in the first year. The first, *stabilisation*, was the period up to two months after operations when unidentified bugs identified and minor changes to syntax and presentation were made often according to the preferences of the individual. This process was effectively a 'settling-down' period when the problems of live use were resolved and interfaces were customised more closely about individual needs. The effects of not performing such maintenance would cause irritation to the system's operators but would not render the help-desk inoperable. A second phase arises after 3-5 months of operations when *restoration* of the knowledge base is necessary in order to retain the validity and correctness of the knowledge-base and thus retain both value to the operator and credibility to the end-user. Failure to perform this second level of maintenance in the help-desk resulted in a lack of confidence in its use and a gradual diminishing of its role in trouble-shooting. Finally, the third phase of maintenance, *enhancement*, was necessary for the help-desk after only 6 months and indicated that major structural changes were necessary as well as integrating features if the help-desk was to retain systems effectiveness and shadow organisational changes.

b) Maintenance Planning: In devising a strategy for maintenance, the developer should take account of the returns on effort expended against the costs of not performing the maintenance work (in terms of erroneous information for example). Figure 7.21a shows for instance that the return on effort is high for programming errors and debugging and maintenance effort should be directed at this category in the first months of operations at part of the process of 'stabilisation'. By contrast enlarging the size of the knowledge base is effort (cost) intensive and should only be undertaken when the costs of not performing this maintenance work are greater much later in the lifetime of the help-desk at the 'enhancement' phase.

c) Depreciation as a Maintenance Cost; To retain the value of the knowledge base and the value of the system generally, the maintenance costs were much more than the assumed figure of 10%. This is because maintenance costs should cover not simply the resource costs of re-development, but also a depreciation cost which reflects the loss of value of the knowledge held in the help-desk over its lifetime. To predict a level of depreciation, more has to be known about the stability of the domain and if possible to quantify this. Table 7.14 maps out the changes in lifetime of computer equipment over the planned lifetime of the help-desk in the company. Two assumptions are made. First that the effective lifetime of computer equipment (and therefore its guaranteed stability in the knowledge base) is the assumed accounting lifetime which is set in the company at *five years*. There are notable exceptions: for instance, plotter hardware has very high maintenance costs and may be scrapped after only two years: by contrast, some personal computers continue to be used in the company after nine years of use. However for these purposes five years is an acceptable average. Thus during the development of the knowledge-base in Year 0, there will be recently purchased computer equipment in its first year of use, equipment in its second year of use etc., up to equipment which is in it's final year of accounting use before it is scrapped as an asset. The design of the knowledge base reflects the equipment in use rather than anticipating future equipment needs. Therefore, after one year of operating the help-desk, Table 7.14 shows that one fifth of the total equipment is scrapped and replaced by new equipment which is different enough from its predecessor that the help-desk no longer remains useful. Thus the help-desk has effectively depreciated in value by one fifth or 20%. With similar logic, at the end of Year 2 the help-desk has depreciated by 40% and by 60% at the end of Year 3. At the end of Year5, the help-desk has no accounting value and indeed may be a liability to the company. For this reason, the effective life of the help-desk has been set at 3 Years with a scrap-value of 40% of the original development costs.

It may be argued that the help-desk should not bother to encode equipment which is in its last year of use because it will be scrapped after only one year of help-desk operations. However there is a second assumption which weakens this argument in that one quarter of all future purchases are repeat orders and therefore by implication one quarter of the trouble-shooting routines remain valid for new computer equipment. Thus, the depreciation of the knowledge base after the first year of introducing new equipment is reduced by a factor of 0.25. The effect is to change the rate of depreciation to the figures shown in parentheses in Table 7.14. These figures are mentioned in the forthcoming cost/benefit analysis in *Section 7.17*. They are sensitive to two assumptions: firstly, that there is a constant purchasing policy for new computer equipment; and secondly, that computer model upgrades are mostly similar to their predecessors.

Table 7.14 Defining Depreciation Rates for the Helpdesk

	Operational Lifetime of the Helpdesk						
	Year 0	Year 1.	Year 2.	Year 3.	Year 4.	Year 5.	
Lifetime status of Equipment	1n[1]	2n	3n	4n	5n	1 (0.75)	Equivalent Depreciation Factor
	2n	3n	4n	5n	1 (0.75)	2 (1.75)	
	3n	4n	5n	1 (0.75)	2 (1.75)	3 (2.75)	
	4n	5n	1 (0.75)	2 (1.75)	3 (2.75)	4 (3.75)	
	5n	1 (0.75)	2 (1.75)	3 (2.75)	4 (3.75)	5 (4.75)	
% Depreciation	0 (0)	20 (15)[2]	40 (35)	60 (55)	80 (75)	100 (95)	

Notes: – 1 *The range of computer equipment lifetimes is between 1 and 5 years 'n' units of computer equipment have been in use for one year, '2n' for two etc.*

2 *Figures in parentheses indicate the equivalent depreciation rate as a proportion of the helpdesk knowledge base. E.g., at Year 1.end, the proportion of knowledge base requiring maintenance is 1/5=20% straight-line or 0.75/5 =15% equivalent depreciation*

These figures provide a useful basis for deriving maintenance and depreciation cost functions and ultimately a cost model of maintenance. This in turn opens up opportunities to investigate the minimum average annual equivalent value (after Black's analysis of transportation maintenance requirements (1987) from which the optimum replacement time for the help-desk may be calculated. Such cost analyses however is beyond the scope of this study and is therefore suggested as future research.

7.17. A Cost/Benefit Analysis of the Help-desk

The goals of the help-desk were near term success, tangible benefits and a relatively short development period. As a first expert systems project with accepted technical uncertainty, it was appropriate at the specification stage to define the economic feasibility of the project in terms of risk using undiscounted payback as the tool. This showed that a return was expected within the three year lifetime of the help-desk although precise costs and benefits were not measured, in part, because of the imperfections in some of the assumptions and estimates made but also because a relative measure of return on capital was acceptable. Even here though, the sensitivity to changes in assumptions was pronounced. However with the opportunity to make use of actual costs and performance measures it is possible in this section to perform a *post-hoc* evaluation of the system using cost-benefit analyses. As before, costs are divided into operating and development costs. However there are a number of additions to the formula to take account of:-

a) the effects of two months extra development,
b) the costs sunk in the Phase 1 project
c) the costs of having three delivery helpdesks
d) the effects of discounting costs and benefits
f) the effects of changing the organisational role of the help-desk
g) and the effects of increased maintenance costs

7.17.1. Help-desk Development Costs

The total help-desk development costs are summarised in Table 7.15 and are made up of the following cost components:-

a) Sunk Costs

If it is necessary to gain an appreciation of tool capabilities; understand the problem domain more fully and be in a position to estimate timescales and resource requirements how should this work be costed? The Phase 1 prototype represented a steep organisational and personal learning curve which improved skills and was instrumental to the success of the OSH. Large companies may have research or training budgets to cover such costs or they may be carried as an overhead. Alternatively, companies with a programme of developments may amortise the total learning curve costs over the lifetime of future projects. For the help-desk the decision was made that these costs should be construed as part of development itself. The main costs from Phase 1 were the costs of expert commitment during knowledge acquisition through to knowledge verification. Figure 7.16 shows that the total delay estimate would be 18 weeks for these phases, had the prototype not been revised. Infact there was only a third of this delay, 6 weeks. Therefore total expert commitment to the prototype is planned duration plus delays. Thus, from Figure 7.7. using the worst case scenario as being true, the planned duration is 17 weeks plus delays of 6 weeks equals **23 weeks.** If cost of expert is £19.61 per hour then this equates to a total cost for four experts at three hours a week of(3 x 4 x 23 x 19.61) = £5412.36

b) New Development Costs

These are borne from two extra months development in enhancing the office systems module and twenty one hour sessions with the expert. This amounts to:

2 extra months @ £600 / month £1200
20 weeks @ £19.61/hour........................£392.20

Table 7.15: An Evaluation of Costs

1. Helpdesk Development Costs			
Cost Factor	**Year 0 Cost (£)**		
Sunk Development Costs	5412.36		
2 Months Extra Development Work	1200.00		
20 Hours Expert Consultation	392.20		
Hardware Development Costs	644.00		
Crystal Shell Purchase	1500.00		
System Developer Costs	6000.00		
Total Helpdesk Development Costs	15148.6		
Discounted at 5% at Year 1 End	**14,391.13**		
2. Helpdesk Operating Costs			
Cost Factor	**Year1**	**Year2**	**Year3**
Helpdesk Operations Costs	4433.33	4211.67	4001.08
Expert Referral Costs	2897.26	2752.40	2614.78
Call Costs	2872.8	2729.16	2592.70
Knowledge-Base Maintenance	2158.67	2050.74	1948.20
Helpdesk Licensing Costs	228.00	216.60	205.77
Helpdesk Hardware Maintenance	611.80	581.21	552.15
Total Operating Costs	13201.9	12541.80	11914.7
3. AS-IS Costs			
Cost Factor	**Year1**	**Year2**	**Year3**
Expert Costs	12071.92	11468.32	10894.90
Call Costs	2872.80	2729.16	2592.70
Total Costs	14944.7	14197.50	13487.60
4. Cost / Benefits			
	Year 1	**Year2**	**Year3**
Net Present Value - Cost Saving (Payback = 8.26 years)	-12648.33	-10992.63	-9419.73

c) Constant Development Costs

The following development costs remain unchanged:-

Hardware development costs	£460
Hardware maintenance costs	£184
Software costs	£1500
Systems Developer Costs	£6000

Thus at Year 0, total development costs add up to **£15148.56**, Table 7.15 shows.

d) Discount rate

If it were assumed that costs varied over the three year life of the help-desk then a discount rate of 10% would be used. However for evaluation purposes cost are assumed to remain constant and therefore a reduced **discount rate of 5%** is used to reflect the loss of value of monies over the three years. Thus, at the end of Year 1, development costs are 5% less at £14391.13. Similarly, Table 7.15. shows that development costs are £13671.58 and **£12988.00** at the end of Year 1 and Year 2 respectively.

7.17.2. Help-desk Operating Costs

The total help-desk operating costs are summarised in Table 7.15. These costs make use of the following assumptions:-

i) Total daily call rate for the OSH is *36 calls per day*
ii) The call performance parameters remain the same, i.e. the 2 trainees handling 70% of all calls and the expert takes the remaining 30% .
iii) Of the 30% of calls received , the average call duration for the expert is 3.6 minutes (from Figure 7.20)
iv) Of the 70% of calls received, the average call duration for the trainees (over the first six months) was 5 minutes. This also takes account of call administration. This figure does not take account of trainee learning beyond the six months however.
v) The Net working day is taken to be 383 minutes as before.
vi) Hourly costs of an expert remain at £19.61 per hour
vii) The costs of the trainee to the company is £14,000. Since the trainees are new total costs approximate actual value to the company.

a) Loading of Trainees: If the trainees handle 25 calls a day (70% of 36) and the personal call level is net working day/average call duration = 383/5 = 77 calls a day, then theoretical staff requirements are 25/77 = **0.33 staff** In other words, one staff need only spend a third of the day on the help-desk attending to end-user calls. Therefore if there are two trainees, then this loading is halved to one sixth of a working day per trainee or 0.167 of the day. Given that the yearly cost of a trainee is assumed to be £14,000. Thus the total trainee costs of attending to the help-desk are 2 x (14,000 x 0.167)= £4666.67 per annum or £4433.33 in Year 1, £4211.67 in Year 2 and £4001.08 in Year 3.

b) Costs of Expert Referral: If the expert is required to attend to 11 calls a day (circa 30% of 36) and the average duration per call is 3.6 minutes, then the total commitment to troubleshooting is (3.6 x 11) = **40 minutes** a day or 0.648 hours a day. Given that the cost of expertise is £19.61 per hour then the daily cost of expert referral is (19.61 x 0.648) = £12.71 per day or £3049.75 per annum which discounted is £2897.26 for Year 1, £2752.40 in Year 2 and £2614.78 in Year 3.

c) Call Costs: Given the cost per call is set at £0.35 and daily number of calls is 36 then daily costs amount to £ 12.60 per day or (12.60 x 5 x 48) = £3024 per annum or £2872.80 at Year 1, £2729.16 at Year 2 and £2592.70 in Year 3.

d) Help-desk Maintenance: Despite the relative neglect of the help-desk knowledge bases and interfaces, maintenance costs were in excess of the 10% of development costs assumed in the original estimates. Indeed for the help-desk to retain its value in the organisation, given the maintenance requirements stipulated in Section 7.16., it is estimated that this figure would rise to above 30%. However since these requirements were not followed, a figure of 15 % will be used for this evaluation.

Thus help-desk maintenance costs are equal to (15% of £14391.13) £2158.67 in Year 1, (15% of £13671.58) £2050.737 in Year 2. and (15% of £12988.00) £1948.20 in Year 3.

On the basis of the analysis carried out in section 7.16 and summarised in table 7.14.,the depreciation rates assumed on the value of the help-desk are Year 1 at 15%; Year 2 at 35%; and Year 3 at 55% of development costs . Taking these as yearly depreciation rates rather than cumulative values, we get:-

At Year 1, depreciation costs are 15 % of £14391.13 = £ 2158.67
At Year 2 ....20 % of £13671.58 = £ 2734.32
At Year 3 ....20 % of £12988.00 = £ 2597.60

It is also instructive from Table 7.14 that the value of the help-desk would be zero after 5.26 years with 100% depreciation. However depreciation is useful as a measure of the value of the help-desk as an **financial asset**, not as an indication of economic cost and therefore it is inappropriate to include these figures as operating costs.

e) **Additional Software Licensing Costs:** The costs of running two run-time versions of the help-desk @ £120 per licensed copy was £240 per year: discounting this equals £228 in Year 1, £216.60 in Year 2 and £205.77 in Year 3.

f) Hardware Delivery Costs: Although there were three machines used to deliver the help-desk, it would be inaccurate to cost each machine individually since the utilisation is much less than previously. On the basis of actual use in relation to the help-desk, the equivalent of one machine is used and therefore total hardware costs (maintenance and depreciation) are for one machine and amount to (£460 + £184) = £644 per annum as before or discounting, £611.80 in Year1, £581.21 in Year2, and £552.15 in Year3.

7.17.3. Costing the 'As-is Situation'

This option is to do nothing and allow the expert to continue to perform office systems trouble-shooting manually. Assuming an average call duration of 4½ minutes, a loading of 36 calls a day at a cost of £19.61 per hour, the yearly costs are £12,707.28 on the basis that:

Daily loading = 36 x 4½ = 162 minutes or 2.7 hours a day
Daily cost of expertise = 2.7 x 19.61 = £52.95
Annual costs = 52.95 x 5 x 48 = £12,707.28

Discounting at 5 percent, costs at Year 1 are £12071.92, £11468.32 at Year 2, and £10894.90 at Year 3 end. The call costs remain the same as Section 7.17.2 , and are discounted to £2158.67 in Year 1, £2729.16 in Year 2, and £2592.70 in Year 3. The total 'AS-IS' costs over the three years are summarised in Table 7.15.

7.17.4. Cost/ Benefit Analysis

The principal justification for using a help-desk was that it would make better utilisation of the expert. There are two direct benefits associated with this: first, the cost savings by replacing the expert by a substitute for 70% of all queries; and second, the cost savings by improving the effectiveness of the expert during trouble-shooting consultations. Two measures are value are used to evaluate the costs/benefits of the help-desk; these are Net Present Value and Discounted Payback:

a) Net Present Value (NPV): This represents the discounted value of the help-desk according to the difference between costs over the three year lifetime of the project. Using the calculated values from Table 7.15.,

NPV =(Development Costs) - (Operating Cost Difference)

Thus, at Year 1: NPV = £14,944.70 - (£14944.7 - £13201.9)= (£12,648.33)

Similarly, at Year 2: NPV = £12,648.33 - (£14197.5 - £12541.8) = (£10,992.63)

and at Year 3: NPV = £10,992.63 - (£13,487.6 - 11914.70)= (£9419.73)

b) Discounted Payback: This is for the benefit of comparing payback based on actual values to those of the estimates made at the beginning of development and discussed in Section 7.5.3.4.

$$\text{Payback} = \frac{\text{Total Development Costs}}{\text{(Total As-is Costs) - Help-desk Operating Costs}}$$

$$\text{Therefore Payback using Figures for Year 1} = \frac{14{,}391.13}{(\ 14944.72 - 13201.86)} = \textbf{8.26 years}$$

This compares badly with the estimated payback of **1.81 years** calculated in Section 7.5.3.4. The discrepancy reflects the fact that the value of money was discounted over the three years; but was also affected by the increased costs of maintenance, the costs of phase 1 development (which came to around £5000) and additional development work. Clearly, given the lifetime of the project as being three years the help-desk cannot be justified in economic terms.

Both cost measures underline that the use of the help-desk can only be justified in cost terms if it becomes a centralised, dedicated and full-time service operated by a non-expert and thus exploiting the utilisation of a low cost alternative to the expert. This requires that the call loading increases significantly which means in practice that further work is necessary on the unfinished IBM, VAX and Network modules to incorporate them into the schema of the office systems help-desk, as originally planned at the Phase 1 stage. At present the cost savings by displacing the expert at a high level of problem solving are diluted between three help-desk functions. Moreover, the cost differential between the expert and trainees is not sufficiently high to compensate for the poor call loading (trainees spend only one sixth of their time at the help-desk) and the additional expense of Phase 1 development.

So far, the cost justification for the help-desk has been made in terms that cost savings would accrue by using the help-desk instead of exclusively the expert. As well as saving costs however, expert systems can add value to the organisation (Harbridge: 1989). These often provide benefits at an organisational level rather than the project level and therefore are much more difficult to define. There were five outstanding added-value benefits of the help-desk:-

i) that the help-desk could provide a quicker and better trouble-shooting service and therefore discouraged end-user 'experimentation',

ii) that the downtime on end-user computer equipment would decrease,

iii) that by making current equipment more effective, less spare capacity is required,

iv) that the help-desk would preserve computing knowledge

v) that the help-desk improved training capabilities.

It would be very difficult to quantify these benefits and there would likely be a high margin of error. However the potential returns to the organisation might far exceed the cost savings. A possible solution is to attempt to quantify all added-value benefits and 'intangibles', as Primrose and Leonard argue (1988). However this in itself can be demanding in time and effort and may prove unacceptable in an organisation based on established accounting methods.

7.18. Conclusions to Chapter 7

This chapter began by defining a model of user participation for the help-desk, where 'user' was taken to mean all those people directly involved in the project and a representation of those who might be affected by the system. The intention was to apply a four tier model of user consultation and participation at contextual, conceptual, communications and physical levels of development. Generally this worked well and allowed a schema to be developed which ensured that the right individuals were involved (for personal, political and organisational reasons as well as technical) at various stages of development. Although the most important, participation at a contextual level, where personal roles and functioning of the system were defined, was the most difficult to achieve in practice, for cultural reasons mainly, with the subsequent effect that the design of the help-desk was strongly orientated about the expert's own definition of the trouble-shooting problem rather than that of end-users and end-user management.

It is very difficult to estimate the knowledge content of a problem and therefore define the amount of knowledge acquisition and knowledge-base verification work required. Moreover, there appears to be no scientific or analytical substitute to experience in defining the size and time and effort required to construct it, although even then there may be calibration problems in scaling up, as the progression from evaluation prototype to Phase 1 prototype demonstrated. The problems encountered during the development of the help-desk however are not untypical and there are a plentiful supply of reported situations where time and effort estimates were irretrievably wrong. Despite the pitfalls, a help-desk was developed and implemented and remains in operation albeit in a diminished role to that originally intended.

The fact that Phase 1 planning efforts were inaccurate does not diminish the value of pre-project analysis, although it does draw a boundary around what may be predicted and specified prior to development and what must evolve through the process of evolutionary or incremental development. Pre-project analysis was essential in understanding the organisational requirements of the system, defining design and interface needs and also development process requirements. It was also central to defining performance measures and completion targets, both of which help to identify development drifting (Veryard:1987) early on in the project and thereby allowed the phase 1 project to be re-defined.

There were two outstanding problems in developing the help-desk: the first was the difficult and time consuming process of verifying knowledge which has been discussed; and the second relates to the complexities involved in defining an appropriate mode of interaction between the organisation (represented by the user community), the help-desk operator and the technology itself, such that knowledge is presented in an analogue rather than digital form to the operator, and ultimately to the end-users. Considerable time was spent in designing 'clever' interfaces which would operate at different levels of interaction according to the competence of the end-user. However there were technical and practical difficulties in achieving this objective in that the programmer had to define explicitly different modes of consultation, with variations on screen presentation and depth of diagnosis for each. The limitations of Crystal as a stand-alone, backward chaining shell were most acute in this area. There were also organisational limitations in the notion of levels of interface. Foremost was that the range of user needs and abilities was much too great to be accommodated by the help-desk and therefore expectations of what the system could do ought to have been lessened. A second limitation in defining levels was that this simplified the skills of the expert as an interpreter in matching knowledge of the domain with the personal needs of the end-user. The transition to the Office Systems Help-desk (OSH) was an appreciation of this and reflected both a personal and organisational learning process which acknowledged the perplexities of knowledge sharing as an organisational objective and the more realistic expectations of what an individual new to the company can achieve in twelve months.

An unresolved issue in expert systems development, which is punctuated throughout this chapter, is the uncertainty over maintenance requirements. Different methods of undertaking this responsibility have been mentioned (e.g. in Guida & Tasso :1989) but there is no suggestion as to how to predict the amount of maintenance required for a domain nor how this might be costed in present value terms over the planned lifetime of the project. Two concepts were introduced in the case of the help-desk; firstly, that the physical resource costs of maintenance are for reasonably stable domains about fifteen to twenty percent of initial development costs discounted over each year. This figure is consistent with experiences with the help-desk: however analysis at the planning stage has shown that the economic viability of the project is highly sensitive to changes in this figure. A second concept which deserves further research, was to make use of depreciation theory, and particularly replacement theory (Black :1987) to determine the effective lifetime of an expert system in the organisation. This works on the principle that the knowledge held within an expert system is a financial asset whose value like a car will reduce over the years due to the ephemeral nature of knowledge itself. A depreciation model was developed for the help-desk based on the accounting lifetime of the computer equipment it was designed to trouble-shoot.

Perhaps the most significant conclusion that has emerged from this chapter is that the help-desk cannot be justified using cost/benefit analysis. A question from this then is, 'if these analyses showed an unfavourable result, does this mean the help-desk was of no value? ' Using an economic model of evaluation, yes: however economic analysis is limited because it cannot measure qualitative benefits other than by their end effect. For instance, the help-desk provided an effective training device for the trainees which enhanced learning and eased their transition into a live trouble-shooting environment. In the end state the tangible cost benefit from this is that the help-desk would relieve the expert of some training duties . This though does not comprehend the organisational and educational value of the system to the trainee.

There is also a second set of benefits which should be quantified but which are very difficult to do so. These include the significant savings in down-time by providing an improved equipment trouble-shooting service to end-users:. and a reduction in the necessary spare capacity of PCs printers and peripherals to compensate for this downtime. These are valid organisational cost savings which by their very nature are difficult to establish. Furthermore, and again at an organisational level, the help-desk as a first project, had an important role as an agent and facilitator of 'knowledge transfer and' was also used itself as part of the technology assessment process. These examples indicate that although economic models of evaluation are essential, they are not complete when processes of change as well as the end state are evaluated. Indeed if one were to evaluate the great deal of time spend on problem identification, application selection and in defining an organisational framework for development on the basis of the payback of the help-desk then it might judged to be a waste of resources.

The limitations of economic measures call for a creditable alternative which provides a wider framework of evaluation. One which looks at the whole process of introducing expert systems technology into the organisation and its effects; together with an assessment of the mechanisms of innovation, assessment and development. These issues are brought together in the next chapter within a framework, not of Technology Assessment necessarily, but of technology transfer.

Chapter 8.

Enlarging the Context of Expert Systems Evaluation

8.1. Introduction

The aim of this chapter is to place the assessment and development experiences of applying expert systems technology in the client organisation into a wider context of evaluation in two ways, as Figure 8.1., an outline to this chapter, shows: firstly, by making comparisons with other manufacturing organisations' experiences; and secondly, by placing experiences within a model of technology transfer in order to explain the total innovative process.

The first aim was made possible through a survey of 135 manufacturing companies, all of which were at some stage in the lifecycle of expert systems development. The survey was undertaken in collaboration with a London based expert systems consultancy and an international journal, although despite the scope of circulation, the sample size was restricted to UK manufacturers only. The analysis of survey results was broken down into seven main sections according to an analysis of:-

a) organisations using expert systems,
b) areas of business in which ES are being used,
c) technology, and the form it takes when applied,
d) approach and processes of development,
e) perceived and actual benefits and constraints of implementation,
f) attitudes towards maintenance and the methods used.

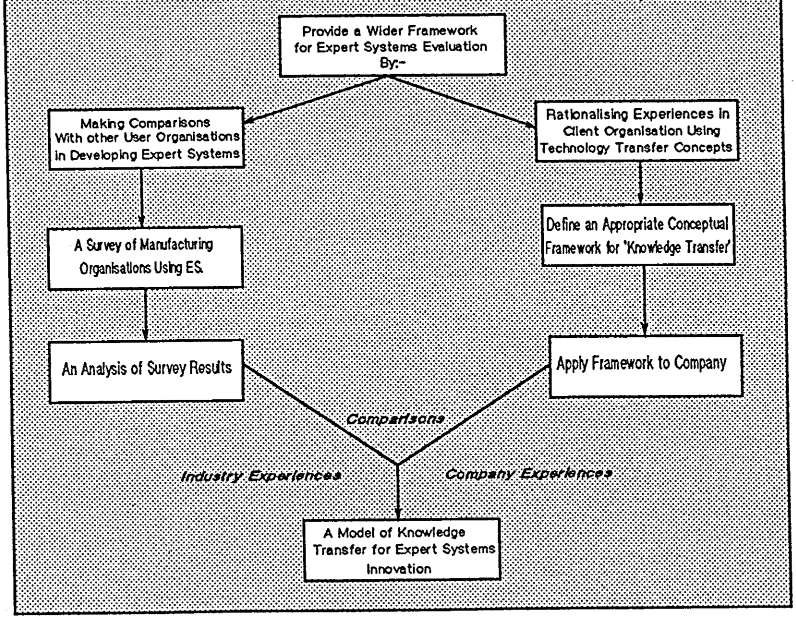
The results of the survey provided a useful comparison, particularly between the experiences of respondents who had developed *diagnostic* expert systems with those of the author in developing the Office Systems Help-desk. Although there were a number of commonalities in approach, shared problems and barriers to implementation, it is argued that there are also fundamental differences in the way that technology assessment was undertaken by these organisations in contrast to the approach fostered in the client company. At the end of this stage of evaluation, the survey results are reinforced with available literature to make general observations on the state of use of expert systems in manufacturing from a Users' perspective.

The second approach to evaluation used in this chapter makes use of technology transfer, and more specifically 'knowledge transfer', concepts in order to rationalise the author's own role and that of the client organisations in the assessment and development of expert systems. Having defined these terms, a review of current technology transfer model shows that the bulk of technology transfer models are inappropriate because of their implicit focus upon innovation from a supplier's rather than a user's point of view. Thus, in terms of mechanisms, they draw attention to enhancing the availability and dissemination of ES concepts, information and technology in a typically deterministic mode without drawing attention to user needs and delivery. Consequently, 'failure' is often described in technical or financial terms when the real source of failure may be through the inappropriate choice of transfer mechanisms and processes

The shortcomings of current models led to the adoption of a simple evaluative framework made up of three essential components of technology transfer; accessibility, mobility and receptivity. This framework provided the opportunity to assess approaches towards transfer and diffusion at different levels in the organisation and with respect to discrete transfer processes and mechanisms. Four levels of transfer were identified: market level (which includes vendors and external suppliers of information or support, e.g. academia, government etc); organisation level (specifically the client organisation in this case); group level (such as the ES project team, computer department, end-user community etc); and individual level (e.g. manager, end-user, expert, developer etc.). An analysis of the client organisation focused upon mobility and receptivity issues at the organisational and individual levels particularly. This made it possible to assess the author's personal effectiveness as 'transfer agent', and the company's attitudes and response towards technology change. It also provided an opportunity to evaluate the effectiveness of the help-desk itself as a transfer mechanism in diffusing ES ideas and concepts .

Although the User survey is treated as a distinct process of evaluation from the analysis of knowledge transfer processes, in fact it provided a cogent argument for the need to address mobility and receptivity issues during ES innovation. Both set of experiences are therefore combined in the final section of this chapter to define an 'emergent' model of the knowledge transfer process. The various attributes of the model are described and examples of its use are provided.

Figure 8.1: A Mapping of Chapter Eight.



8.2. Evaluation by Comparison with Other Manufacturing Companies' Experiences

An inevitable problem through being immersed in a single organisation is that many of the developmental experiences and the viability of the development approach itself may only be tenable for the client company alone. In order to identify how universally relevant the development approach used in the client organisation was, and also to see if there are common problems in the transfer of expert systems, it was necessary to compare personal experiences with those of other companies. However, then and now, there is very little feedback on precisely how expert systems are introduced into manufacturing organisations (Bramer:1990); how they are justified and developed (Harbridge:1989); and how they are actually used, if at all, in an operational context (Jamieson & Szeto: 1989). The analysis is subsequently limited in scope to discussions on the market and uses for the technology (see Frost & Sullivan:1990 for example). Moreover, where research does look at the types of user organisations that make use of expert systems and their choice of methodology and development approach (e.g. O'Neill & Morris: 1989), it does so again from a vendor viewpoint. Recent studies though, have begun to evaluate expert systems from a user-organisation perspective and provide useful insights into the processes and mechanisms of knowledge transfer together with possible barriers and organisational impacts. However it is difficult to make generalisations for the manufacturing sector from these studies because the sample size was often too small (e.g Bramer: 1990); or because they address all sectors of industry and education such as finance, commerce and government rather than concentrate on manufacturing alone (e.g. Harris et al :1990). Furthermore, although studies are available which meet all these criteria (e.g NIEVR:1989), as foreign studies they lose their significance in the unique cultural and economic settings of the UK manufacturing sector.

8.2.1. Questionnaire Design & Methodology

For these reasons, the author collaborated with a London based consultancy, 'es (Connect)', in performing a national survey of expert system users. The survey had four primary objectives:-

a) Who use expert systems ?
b) How are expert systems are being used
c) How well do expert systems match users' needs ?
d) How integrated are expert systems into companies' larger activities ?

How well the survey accomplished these objectives is unclear. However, the survey was seen as a valuable vehicle for the author to investigate the approaches to expert systems innovation adopted by other manufacturing organisations. The emphasis was therefore upon making the best use of the survey results despite a number of questions and over two thirds of survey responses not being directly relevant to the thesis study.

8.2.1.1. Questionnaire Design

The questionnaire consists of eight sections as Appendix XVa. shows. The first section provides factual information about the respondents which was of little use other than for follow-up purposes. The second section provides an aggregate measure of the overall status of expert systems in the User Organisation (UO) and then a more detailed breakdown of the total number and status of specific expert system projects being undertaken. The status of projects varies from nothing more than an idea to a fully operational system and is intended to provide a measure of commitment by the UO towards expert systems.

Section 3 of the questionnaire provides a description of the industry in which the UO is based; these were later coded according to standard SIC company classifications. It also requested details about the job function of the end-user of the system and how it was intended to be used. Beyond Section 3., the focus of questions shifts from an organisational level and concentrates on specific application projects. Section 4 begins by determining the present status of the application, the hardware and software being used to develop the system and the job function of the individual held responsible for its development.

Since the application may be used for a number of reasons Section 5 asks the respondents to rank those task functions which are most important, e.g design, configuration, planning etc. This also indicates functions which may be supported by the application. From systems classification, Section 6 looks more closely at how the system is used. Questions distinguish between the original systems design (e.g. replace an expert or provide advice only) and the actual design in practice; and identify the level of integration necessary for the application to function - whether it is sufficient as a stand-alone system or requires interfacing with a database for instance. Section 6 concludes by establishing how, if at all, maintenance of the knowledge base is carried out and by whom.

Section 7 looks at how user organisations go about developing their systems: whether for example they use formal methodologies or informal guide-lines, and if they use software engineering and project planning tools or software as an aid to development. It also determines the sources of information on which the knowledge base is founded. Assuming that a system of some form is developed, the final section attempts to qualify its organisational impact and implementation benefits together with constraints which might have prevented or impinged upon its successful implementation. In terms of benefits and constraints the questionnaire makes use of both ranking and rating methods of data input. Ranking was intended to provide a measure of importance for each constraint or benefit in relation to all others. By contrast, the ratings indicated the significance of individual benefits or constraints as perceived by the questionnaire respondents.

There were a number of weaknesses in the design of the questionnaire, most notably the poor presentation and wording of some questions and the respondents confusion over rating and ranking in Sections 5 and 8. There are also many more questions which the author would have wished to be included, especially on the use of the expert system and learning more about the development and operational settings. However, given the limited time available to perform such a study, the es (Connect) survey provided access to over 130 end-users and, given that the quality of responses were very high, provided a great deal of information which was unattainable by other means.

8.2.1.2. Questionnaire Methodology

During the design phase of the questionnaire, meetings were held between es (Connect) and the author who had designed a questionnaire with a number of common questions. Although the questionnaires differed in style and depth, there was great potential for collaboration. It was agreed therefore that a summary of the findings would be published in the July 1989 edition of the Systems International journal and, following this, a more detailed and rigourous analysis would be undertaken by the author for inclusion in the thesis although the results would remain the property of es (Connect) and could be used accordingly.

The questionnaire (exhibited in Appendix XVa.) appeared in the March 1989 edition of Systems International, a journal aimed at computer department management and information technology specialists. It was assumed therefore that the audience for the survey would generally be receptive to the survey and that respondents would consist solely of ‘interested organisations’. In total, over 460 questionnaires were returned complete. From this number, the author selected 135 for detailed analysis based on three limiting criteria:-

a) The application must be in the UK : the journal had a wide European circulation with almost 15% of respondents from France or Germany.

b) The application must be in the manufacturing sector. Over 50 responses were from financial services, 60 from the defence sector and the rest from local authorities, education, construction, computer services and medicine.

c) All respondents selected must be ‘end-users’ rather than producers or suppliers of expert systems software or specialist IT consultants. A special case was made of user organisations who championed the project but may have used consultancies to perform some specialist task of development. This is distinct from collaborative arrangements where user sites agree to test vendors’ new expert systems software before it is launched commercially (twelve respondents proved to be Alpha and Beta sites for vendor products and were not included).

8.2.2. Analysis of Results

The results are presented in seven sections. These do not correspond directly to those of the questionnaire design. All Figures referred to in this section are listed in Appendix 15b

8.2.2.1. An Analysis of Organisations

This section provides information on the backgrounds of respondents. Figure A1 in Appendix 15b shows the distribution of the sample according to manufacturing sector. These have been divided into seven main categories: transportation manufacture (which includes aeronautic and automotive manufacture and ship-building); computer and related electronic component manufacture; consultancy and consultant engineering; plant and process control (this includes chemical companies, paper and adhesives, materials fabrication such as rubber and plastics); supply and light engineering (such as cable manufacturing, electrical equipment, telecommunications and precision instruments); and heavy engineering and manufacture (such as steel manufacture, metal products and machinery fabrication and power engineering). Figure A1 also shows that there is a fairly even distribution of the sample across sectors, although computing and electronic component companies make up over a quarter of the total sample. This reflects a general trend that most ES activity is concentrated in high technology sectors since the diffusion of ES ideas and concepts is the most concentrated (Ovum:1988a).

Figure A2 in Appendix 15b shows the overall status of applications in respondent organisations. Since the questionnaire was naturally bias towards interested organisations, it is of no surprise that respondents are involved in some stage of development. For a quarter of the sample, there was very little commitment towards development per se , but rather an interest in the technology and an obligation to 'maintain a watching brief' and find out more about the technology, its strengths and weaknesses. More generally, D'Agapeyeff and Hawkins (1988) note that one reason why a large proportion of manufacturing are still at this level is '..the contradictory explanations they receive as to what these systems are, or should be...In these circumstances, it is understandable for management to pause until the matter us resolved by the emergence of a consensus view' (p187). The authors add moreover, that many companies will fail to develop beyond this information gathering level because of the lack of management commitment; because of business secrecy, which prevents proven expert systems in companies from being demonstrated (thus inhibiting the transfer process between organisations); and by a fear of the nature and the costs of this technology. The latter point is again borne from the lack of demonstrable systems and benefits. Indeed it is perhaps for this reason that 28% of respondents have built a demonstration expert system in their organisation prior to further development. Many respondents spoke of the 'immaturity of the technology' and the purpose of the demonstrator was as a fail-safe which allowed some experimentation without risk or significant costs. Moreover, respondents clearly separated this from a next stage of development, production of the full-scale prototype in which 29% of the sample were involved.

Other surveys have shown the difficulty that organisations have in making the transition from working or full-scale prototype to operational system . O'Neill & Morris found that of a sample of 38 software houses representing a total of 600 applications, only 25% had reached operational status. A more recent survey by Harris et al (1990) shows that when user organisations were interviewed directly this figure was only 18%. Figure A2 shows that only 14% of respondents had operational systems. This, of course, does not suggest that the application is commercially successful necessarily but that it has achieved 'live' status by being implemented and used in the company.

It would be incautious to conclude from Figure A3 that some manufacturing sectors were more successful in producing operational systems than others, although it is interesting to note that light and heavy manufacturing sectors had no operational systems. All sectors were involved in earlier development stages: however the computing and electronic component sector as well as having the greatest number of operational systems also had the most problems in terms of abandoned projects and lack of interest. Of those companies that expressed no interest in expert systems, Figure A4 shows that the main reason was lack of awareness : as one respondents put it 'nobody knows what they are or how they are used '. The other main reasons suggest that a preliminary evaluation of the technology had taken place and it had been concluded that there was no use for the technology (30%), that it was too costly (20%), or that the intended problem domain was judged to be too complex for the technology (10%).

8.2.2.2. An Analysis of Applications

This section looks at individual applications. Most respondents provided information on their latest project and for this reason most were at the ideas and early stages of development and only 6% were at the operational stage as Figure B1 shows. For this reason, the questionnaire tended more to mirrored respondent's expectations rather than actual systems performance.

Figure B2 shows the most important application function for respondents. By far the most significant role for applications was in diagnosing faults with nearly a fifth of all systems designed for this purpose. Furthermore, over 60% of all applications were concentrated in only five application functions,: diagnosis, information interpretation, alert/warning, design and selection. This suggests a preference to apply expert systems to a core of application domains. It may also indicate as D'Agapeyeff and Hawkins (1987) suggest a preference towards 'simpler expert systems' in proven application domains. This is endorsed by the survey which shows that of those applications which were of operational status most were shell-based and limited to diagnosis, selection and information interpretation (text animation for example) domains. As well as defining a principal application function, most expert systems were also capable of supporting other functions. Figure B3 shows that the most supported application functions were in the interpretation, presentation and acquisition of information, these being an essential component of most generic application functions. Respondents also judged prediction to be a necessary component of their systems.

8.2.2.3. Analysis of End-Users

This section looks at the actual and intended end-users for the application. Seven broad categories of end-user were identified from the survey ranging from management, and technical design specialists to shop-floor workers and administrative staff. Figure C1 shows a fairly even distribution among these job functions, with the most common end-user being design/project engineers and manufacturing(shop-floor) operators (both at 22%) and the least common being the IT/computer specialist(at 7%) in the user organisation. This suggests that expert systems are being used across a spectrum of skills and job descriptions. It also dispels the notion that expert systems are being used at particular levels in the organisation. Indeed, Figure C2 shows that more than half of respondents planned or had systems whose most important purpose was to support a knowledgeable end-user (this may be a design expert or a skilled machine operator for instance) in a support capacity. This has two implications: firstly, that expert systems were being used to improve or enhance the capabilities of the expert (by improving decision-making productivity, speed of recall, presentation of results etc); and secondly, that where the end-user was an expert substitute, the person was still required to have a good understanding of the domain and moreover, the dependency upon the expert or specialist was seldom removed completely. Indeed only 9% of respondents intended that the most important role for the application was to replace the expert or remove human participation entirely from a work task (for example by creating a front-end interface to a database or complex computer system).

A second important use of applications was in providing advice to 'lay' people so that, for example, they may undertake their own trouble-shooting or perform selection and configuration procedures without continually referring to specialists. This category of use was especially valauble to management who required a functional understanding of all areas of business. A third category of use applied to monitoring, real-time and process control systems whose function was to interpret feed-back information from machinery and other on-line information and take subsequent control, planning, diagnostic or scheduling action.

Figure C3 looks to see if there is a predominance of certain types of end-users according to manufacturing sector. Some of the results are to be expected: for instance in computer and electronic based companies, the end-user is likely to be an IT/computer specialist; and in process control and chemical companies, the applications are primarily shop based and therefore operated by shop-floor operators (Milne:1988). Similarly, in consultancy and consultant engineering organisations, there is are many cases where the end-users are engineers using expert systems as decision-support aides. However, other results are less answerable: for instance in light engineering and supply companies, the end-user is likely to be the shop-floor manager.

8.2.2.4. Analysis of Technology

Figure D1.1 shows that over a third of all respondents made use of shell-based ES software; a quarter made use of A.I. tools and the remainder used A.I. languages and conventional programming techniques. This corresponds roughly to the market situation for these products (Ovum:1988b). No application specific shells were used; nor were there any ES environments. In terms of hardware, Figures D1.2 and D1.3 show that the most common development and delivery medium is the personal computer (at about 60% in both cases) whilst the second most common hardware is the organisation's existing Mainframe system (at 21% and 23% respectively). Again the results reflect a national trend away from dedicated A.I. machinery towards equipment that is available in the user organisation. It might also reflect a reluctance to commit more substantial resources towards expert systems development until the technology is considered proven.

A further measure of commitment and complexity is the extent to which the application is integrated in the organisation. Figure D2 shows that although the single most common level was stand-alone, most systems were either embedded, linked or integrated in some way to other company systems or software. The most frequent form of integration was as an interface to databases with DbaseIII, Clipper and Oracle proving the most popular. Twelve percent of the sample also interfaced their ES applications to other computer systems such as Computer Aided Design (CAD) and Material Requirements Planning(MRP) systems. A further 12% also designed ES which were embedded and formed part of a wider information system- for example intelligent scheduling and planning systems. The remaining 15% were linked systems which were downloaded to or loaded from other systems or formed a front-end to other systems.

A possible hypothesis which deserved testing was that the level of integration of the expert system would alter the nature of the end users' role. A cross-tabulation of integration level against system use however , shown in Figure D3, reveals no indication that stand-alone systems are likely to encourage a 'replacement focus' or that interfacing will necessarily enhance the quality of the end-user function. This is more dependent upon the organisational integration of the system in the workplace rather than its technical interfacing features (Dawson:1988).

8.2.2.5. Analysis of Development Approach and Processes

Other work has shown the absence of a development approach to be a key feature in the development of expert systems in user organisations (Harris *et al*). Jamieson & Szeto moreover argue that a reason for this is because current methodologies are difficult to apply and, as Chapters 4 and 5 point out, lack consideration of project selection, justification and organisational issues in development. The user survey mirrors these findings and more generally identifies a great deal of confusion over how best to develop expert systems. Figure E1.1 shows that only 65% of respondents endeavoured to use a methodology or published guide-line. Furthermore, of these, there appeared to be no coherent pattern of development with companies placing different emphasis upon different phases of development. Figure E1.2 shows that of that used guide-lines, there was little or no attention to pre-project specification and analysis; maintenance and upgrading; and evaluation and justification. Indeed, in a survey carried out by Jamieson & Szeto most respondents assumed that the application was viable without the need for evaluation and most justifications were carried out post-hoc or were accounted for within an 'experimentation' or R & D budgets.

Figure E1.2 also shows that the primary component of development was prototyping with over 78% of those that followed a methodology or guide-line adopting this technique. However in many cases, this was seen as a substitute for requirements and feasibility studies and project planning activities. Of the respondents that did attempt to incorporate pre-implementation design and analysis activities into the development methodology, Figure E1.3 shows that the approach taken was to make use of currently used software engineering techniques and planning systems in an attempt to structure and control the development process. Methodologies such as Mascot, Demarco, Yourdon and company specific planning and control techniques were used for this purpose. It is significant that from the sample that considered design and analysis activities, only two organisations made used of ES specific development lifecycles, described in Chapter 4., and both had problems applying them because of their 'complexity'. Equally discouraging is that only three and five companies respectively had attempted to perform business and functional analyses to systematically identify and define possible ES applications. Despite this, all were enthusiastic that their particular approach had been successful in choosing a viable and useful application. One organisation for instance had made use of Checkland's Soft Systems Methodology in combination with IBM's Business Planning Approach for this purpose.

Following analysis and design comes knowledge acquisition. Morris and O'Neill explore the usefulness of the increasing number of methodologies produced by the research community for eliciting and organising expert knowledge. They argue on the basis of survey results that in practice only the most simple interviewing techniques are being used. This however might reflect user organisations' lack of appreciation of the importance and value of knowledge acquisition methodologies as much as a misalignment of research needs. Results from the user survey show that the main source of information for systems was through structured and unstructured interviews. Figure E2 shows that although this technique was used by nearly 37% of respondents, equal use was made of secondary sources of information such as technical literature and manuals. It is also evident from Figure E2 that user observation techniques and the more analytical deep reasoning based methods such as Repertory Grid Analysis, described in the last chapter, were used infrequently. Both observations reflects again a preference towards simple shallow reasoning systems whose purpose is more to present information and high-level knowledge in an effective manner than perform complex decision-making actions.

The choice of development methodology may be as much a function of who is responsible for development and personal preferences than any rationale based on what is required. Responses broadly fell into five main categories of development responsibility: systems or computer department; end-user; specialist knowledge engineer; consultant; and other function. Figure E3 shows that 28% of projects were under the responsibility of the specialist knowledge engineer. This person was either trained or employed especially to do manage the project. Other users preferred to retain responsibility within the computer department (23%) or to the person considered technically the most competent (if not the knowledge engineer then it might be an external or internal consultant (20%)). In the remainder of the cases, responsibility was attributed to senior figures in the organisation who had championed the project (most frequently the technical director or equivalent).

In cross-tabulating intended design use against development responsibility, Figure E4 seeks to establish whether certain project viewpoints invoke particular design uses. The results show significantly that in all cases where the intended design use is to replace an individual, the person responsible for development is the end-user. By contrast, where the expert system was designed to give advice to a lay person, in most cases the person help responsible for development was the knowledge engineer. This dichotomy might lead to a number of conclusions: foremost, it suggests the personal and political ends of the end-user may be divergent from those of the organisation; and secondly, that although end-user participation is essential towards the success of the project, actual control and responsibility by the end-user may have negative organisational effects. Figure E4 also shows that in all cases where an expert system was designed for real-time feedback control, the person held responsible for the project was an outside consultant. This highlights the complexity of the task and the need for user organisations to liaise with specialist expert system based consultancy organisations.

8.2.2.6. Implementation and Evaluation

This section looks at the performance of systems which have achieved operational status or are close to it. A first question is to what extent are systems used as originally designed ? Only a quarter of respondents stated that their designs had changed through development and Figure F1. identifies the main reasons for this. Of those respondents that had made changes to the design, three categories of change emerged all of which echo the problems encountered in the development of the Help-desk. These are: changes to the size in the domain (45%) - many respondents described how implementation constraints such as shortage of time and the difficulties of knowledge verification had forced a reduction in the scope of the project or design features; secondly, changes to the user interface (33%) -companies were generally over-ambitious in the interfaces they hoped to achieve given the limitations of the software tools they used. Finally, Respondents described how their systems were used in a different way to that intended: only one company saw this as a positive attribute.

The benefits were analysed using both ranking and rating methods. Ranking allowed those factors to be identified which were considered essential to success of the application. Figure F2 show that the most important expressed key benefit (ranked first) was that the expert system would increase the accuracy of decisions, followed next by an increase in problem solving ability and an increase in the quality of work. Figure F3 takes a slightly different approach by ordering benefits according to those ranked 1, 2 or 3 in importance. In this case an increase in the quality of work was considered the most important followed by increases in problem solving ability, accuracy of work and accuracy of decision-making. Both results show a tendency to express benefits in added value terms rather than cost reduction terms such as reduction in staff numbers or using cheaper staff with lower skills. This also reflects the findings shown in Figure F3 that for over half of respondents, the intended use of the system was to support a knowledgeable user and in this segment, these qualitative benefits were consistently rated highest. By contrast, where the intended use was to replace people, key benefits were to reduce skill levels , increase work load and enable staff reductions. Where the system was designed to support a lay person, respondents tended to provide their own benefits not supplied by the questionnaire. These included such factors as 'improving training capabilities, allowing operators to work independently, allow faults to be recorded effectively and so on. However, 'increase work load' and 'increase output' were more important benefits in this segment.

A second measure of benefits is rating the strength of feeling that a factor is important. This was calibrated on a 0-4 scale in the questionnaire and the results, shown in Figure F4., are expressed relative to the sample mid-point or mean for all factors. A positive factor suggests that the respective benefit is more important than the average, and negative factors less important. The results from Figure F4 are broadly similar to those of the rankings but for one exception; in expressing the *perceived importance* of benefits, respondents considered cost effectiveness the most important factor whilst ranking it only moderately important relative to the other factors. A possible explanation is that most respondents were in the process of development rather than at the stage of operations and they were therefore more preoccupied about cost effectiveness and cost control than the end benefits. It is significant from Figure F4 that the benefits considered to be the least important relate to downgrading individual roles through a reduction in skilled personnel required or a general reduction in skill level.

Constraints were also analysed by ranking and rating data in order to assess the *strength of feeling* that a particular factor contributed towards implementation problems (Figure F5); and the factors that *actually* contributed to preventing implementation (Figure F6.). The most important constraint, that there was a lack of budget provision is due to the fact that management could not be convinced of the technology's benefits. Indeed, a lack of management support, awareness and understanding of expert system benefits were all rated highly. This does not suggest that management are an unjust encumbrance upon the process of expert systems transfer, but more of a 'regulator' ensuring that ES projects are developed because they have a business and organisational value rather than because they are purely of technical interest. It is apparent from the survey that many of the systems are developed by 'enthusiasts' (whether users, experts or appointed knowledge engineers) and as a result are separated from company's mainstream information technology and computing functions with the effect that they may not follow company strategy or assessment procedures. Moreover, the fact that awareness and support was a rarity and there was a lack of confidence in systems suggests a bottom-up technology driven approach towards development by such enthusiasts rather than an organisationally driven top-down approach . This may be reaffirmed by earlier results which show the almost exclusive use of design tools in development and the near complete absence of business planning and functional methodologies.

The least important constraints are those related to the personnel resources such as lack of suitable experts, lack of identified users and technical resources, although a lack of knowledge engineers was cited as the third most important implementation constraint. The label knowledge engineer was not helpful since many of the projects were small scale and the expert was often responsible for development, as well as being the 'knowledge engineer' and eventual end-user, again highlighting a 'hobbyist' approach rather than a process within mainstream systems development.

Some of the most strongly felt constraints were specific to organisations. These included too high expectations of what the project team and expert systems could achieve, a lack of time , no access to experts, no access to end-users, changes in the organisation, and a lack of interest by end-users. Moreover, and without exception, organisations which expressed these and similar constraints had not used a methodology and instead had adopted guide-lines which looked at technical design issues only.

8.2.2.7. Analysis of Maintenance Requirements

This section looks at respondents' attitudes towards maintenance and the allocation of resources in performing this function. The first interesting observation is that 40% of respondents did not have a view on maintenance (and this does not include those that didn't know what to do). Of those that did comment, six clear categorise of maintenance were identified as Figure G1. shows. Of these, the largest group responsible for maintenance was the systems developer (33.5%). Respondents in this group viewed maintenance as anything from changes to screen layout and bug fixtures to major enhancements to the knowledge-base, and believed that formal documentation and change control procedures ought to be managed and undertaken by the most qualified person available, although most were unsure precisely how this should be carried out. In contrast 25% of maintenance work was carried out by the end-user, reflecting the high proportion of experienced users developing their own systems. Respondents in this category argued that the end-user (and in some cases the end-user was the expert) had the greatest understanding of the domain and would remain committed to using the system if the individual had an active role in future maintenance and development work. Nearly 10% of respondents intended to use automatic methods of maintenance by providing automatic input through feed-back control real-time data updates, and automatic induction techniques. However few of these companies had actually implemented a system and were in the process of development suggesting, possibly, that they were not fully aware of the technical complexity of achieving this in practice.

Perhaps most significant though, is that more than one fifth of those that commented on maintenance indicated that they were unsure how to go about performing maintenance work or would not bother to do any and therefore leave the knowledge base static. Very few companies from the total sample discussed maintenance as a design issue and instead considered it as something to consider during implementation. On this issue, Figure G2 shows how attitudes towards maintenance change over phases of the development cycle. At the ideas stage, well over half of respondents are undecided on how to maintain the system or assume than it may be left static. As development progressed to demonstration and prototyping phases, possibly because of the necessity to involve the end-user and expert in these activities, almost half of respondents believed that either the expert or end-user should undertake maintenance work. Further development showed an increased preference towards the systems developer until at the operational phase over 50% of respondents believed that the maintenance should be undertaken by this person.

Figure G3 checks to see if attitudes towards maintenance differ according to who is responsible for development. There are two significant observations: first, than when the user (who may be expert or end-user or both) is responsible for development then responsibility for maintenance is also passed on to the end-user or expert. Secondly, when development is under the responsibility of the specialist knowledge engineer, maintenance is clearly viewed as a systems development function. It is reasonable to assume that maintenance responsibilities might change if the software tools adopted were easier to use. However Figure G4. indicates that although systems using A.I. languages, being more complex, required maintenance by systems developers as expected, there was no evidence that shells, as the least difficult software to use, promoted maintenance by end-users or experts. Infact, over 30% of shell based systems were either left static or else it could not be decided how maintenance should be carried out.

It is also interesting to see how the level of integration affects attitudes towards maintenance. Figure G5. provides a comparison between stand-alone and embedded expert systems. In the latter, the ES forms only part of a wider integrated system. In that most embedded systems share information with other systems, it is essential that the knowledge-base is continually up-dated. For this reason, no embedded systems are left static and relatively few compared to stand-alone systems do not have a maintenance plan. One of the greatest benefits of embedded systems is that maintenance is made easier by direct information entry into database files or relational databases held in other computer systems and then converted into a knowledge format for inclusion into the knowledge base of the expert system (Herrmann: 1990). By contrast, the maintenance of shells requires encoding knowledge into production rules using the specific syntax of the programming tool. This accounts for why in Figure G5 there are twice the number of user and expert maintained embedded systems than stand-alone systems and, conversely, double the number of stand-alone systems to embedded systems maintained by systems developers.

8.2.3. Comparisons with the Office Systems Help-desk

The Office Systems Help-desk (OSH) was developed in a particular way for specific reasons. During the process of developing the system numerous problems were encountered, some of which could be resolved, whilst others required changes in the design and scope of the system. Although many of these experiences were specific to the organisation and its settings, it was plausible that some were characteristic of the technology per se, and in particular of diagnostic expert systems. The purpose of this section therefore is to compare the experiences of respondents from the User Survey who developed diagnostic expert systems with those of the author in developing the OSH. Diagnostic systems (fault trouble-shooting, error correction, debugging, diagnostic adviser etc) are the most mature and commercial manifestation of expert systems in manufacturing (MI:89) and this view is endorsed by the survey results which shows that the bulk of operational systems are of this form.

Diagnostic systems had two clear roles as Figure H1.2 shows. By far the most important though was in using these systems to provide advice to lay people. Figure H1.4 shows that in most cases, the people in question were either manufacturing operators (such as machine controllers, computer operators or automated-cell based-manufacturing controllers) or design engineering staff (for purposes of printed circuit diagnostics, electronic design analysis and rectifier debugging for example). In some sectors of manufacturing, such people were considered as 'knowledgeable users' rather than lay people; however generally, knowledgeable users was a distinct category made up of experts, specialists or practitioners who used the expert system for information support and suggestions. No diagnostic systems were designed to replace an individual because of the importance attached to this function in companies and, by the nature of the domain, the technical complexity of endeavouring to do so. Also by virtue of the technical complexities amongst other reasons, there were no diagnostic systems which made use of real-time feedback mechanisms. This also mirrors the fact that 55% of diagnostic systems were 'stand-alone' with the most important method of integration being the provision of software linkages to databases (14%) as Figure H1.3 shows. As well as being Stand-alone, Figure H2.1 indicates that two thirds of diagnostic systems use rule-based shells. Underlying this figure is a preference towards developing and delivering expert systems on existing company hardware: in most cases, this means either personal computers (53% of all development hardware and 46% of all delivery hardware - Figure H2.2) or company Mainframe systems (33% and 40% respectively).

In terms of development responsibility, Figure H1.1 shows a high proportion of diagnostic systems made use of a knowledge engineer (or systems developer) rather than the end-user or expert despite a high proportion of systems being developed on shell-based systems . Moreover, nearly 30% of knowledge engineers were people brought into the company to perform this task specifically. The knowledge engineering task was considered to encompass systems design and programming and testing as well as knowledge acquisition for most of the projects. More than two thirds of developers made use of a methodology of some form with a higher proportion adopting structured engineering methods than in other domains. However, there were none in this sample that made use of business planning or lifecycle methodologies.

Figure H2.4 shows that the most important method of acquiring knowledge was through interviewing the expert or specialist (57%). This was followed by the use of literature in order to become familiar with the domain, and supplemented by reviews of company documents and procedure manuals. Observation, case studies and even intuition were also mentioned as secondary knowledge acquisition techniques. Many of the respondents experienced problems in acquiring knowledge: these included availability of the expert; difficulties in getting experts to express their knowledge; and the complications of verifying knowledge without using a representation technique. For those companies that did use such techniques, the preference was towards the use of informal methods to suit the expert rather than the developer: these included pictorial representations such as graphs, spider diagrams and rich pictures. Again there was an emphasis upon making use of existing company tools and capabilities rather than of using dedicated ES tools. In the case of the Help-desk, a similar approach was adopted although there were areas where specific methods, such as fault-tree analysis, had to be used.

The analysis identified a number of similarities between other diagnostic ESs and the OSH. There was a clear tendency, for instance, to make use of simple and existing hardware to construct shell based systems and most systems were stand-alone. There was generally more thought given to implementation than in other domains; maintenance in particular was acknowledged as being an important development issue with half of users with diagnostic systems indicating that this function should be undertaken by professional systems developers in preference to end-users or experts. Furthermore, development was more likely to be undertaken by a knowledge engineer who in turn was more likely to adopt a methodology, albeit simple guide-lines in most cases.

There were also a number of similarities between diagnostic systems in the survey and the OSH in terms of technical problems experienced. These centred on design and development issues such as the difficulties of knowledge representation and verification and the length of time taken to perform these tasks; the absence of formal tools to undertake knowledge acquisition; the complexities of interface design and of defining end-users' needs; and the convolution of structuring rule-based programming for testing and maintenance purposes. Most user organisations were aware that the development process ought to be structured and formalised in some way, but had limited information on those methodologies available or had decided that it would be too complex or costly to use them and made use of simple guide-lines instead (e.g. Waterman: 1986).

As well as similarities, there were also a number of important differences between the OSH and diagnostic systems analysed in the survey. These differences are fundamental and derive from the contrasting approaches to technology assessment as well as development. Indeed, many of the implementation constraints expressed by respondents such as: uncertainty over the choice of approach , tool, and even problem; a lack of support and organisational understanding particularly from management; and a failure to justify the benefits of the system, were a direct result of adopting a bottom-up (technology- push) approach towards ES technology transfer. Many of the respondents received local (group, departmental, individual) support and failed to be integrated within the organisation: as a consequence they functioned in isolation to the established company practices and policy on computer systems specification, design and strategic development.

By contrast, the OSH was one of a number of end products arising from a top-down (problem-driven) programme of organisational needs evaluation, business analysis, and technology assessment and subsequently received full organisational support and commitment. Furthermore, it established a continuity of evaluation and development beyond the lifetime of particular projects or individuals. In short, the approach adopted in the client organisation was not exclusively project based as in the case of many respondents to the survey. Instead, the client company was provided with an evaluative framework and development infrastructure which was problem based and therefore placed a greater emphasis upon identification, selection, specification, analysis and other pre-development issues . Although this approach demands significant organisational commitment prior to tangible demonstrations of a 'product', it is argued that it has more lasting benefits in the longer term and ensures that expert systems are developed in an organisation because they are needed rather than because they are feasible.

8.2.4. General Observations at an Industry level

The findings from both Vendor and User questionnaires lead to a number of observations and recommendations on the assessment and development of expert systems in manufacturing organisations. These experiences are valuable also in defining a set of 'user characteristics' for use in models of technology transfer described in Section 8.3.

a) Customer Base: Given that the survey was naturally skewed towards interested organisations, the response to expert systems was only moderate. The survey showed that although most companies were receptive to expert system ideas and concepts, 60% of the sample were no further in development than providing a demonstration system. This reflects the relatively immature and uncertain state of the customer base and suggests that market penetration is not as deep as vendors might suggest and, moreover, is strongly product orientated in that there is a reliance upon user experimentation of the technology as a means of understanding ideas and concepts and defining application potential. D'Agepeyeff and Hawkins (1988) note moreover, a reluctance on the part of management and organisations to commit their companies beyond this experimentation phase.

Because of this market driven structure, managers receive contradictory explanations as to what these systems are or should be. This is compounded by a lack of factual information on the process of implementing applications and the benefits which they can provide. The main reasons for this, D'Agepeyeff and Hawkins argue, is because of the 'unnecessary secrecy', commercial confidentiality and the dispersion of development throughout industry rather than a focus upon collaboration. Furthermore, Niwa (1988) argues that with the absence of established methods and best practice guide-lines, users may think · that a system's good performance is due to the technology that the vendors apply rather than because of the way in which it is used. This is made worse by the fact that the vendor-user point of contact is very often between technical staff in both organisations who are concerned with the technological aspects of development only.

b) Applications: Despite the potential for great diversity and creativity in the choice of applications, the survey shows a convergence not just upon generic applications such as diagnosis and selection, but upon specific applications such as equipment diagnostics which suggests a preference towards applications with a proven record of success in other organisations rather than risk developing innovative systems.

The reality of expert systems in manufacturing differs significantly from the ambitious claims of the Artificial Intelligence fraternity (e.g. Feigenbaum:1988). Of those projects that achieved operations in the User survey, most were simple in form and scope. Most did not make use of uncertainty reasoning or anything more complex than explicit rule-based representations. Their simplicity has benefits in banishing the widespread misconceptions about expert systems being 'intelligent learning systems.' Indeed, part of the problem in defining the business and organisational role for expert systems has been in understanding what they actually are. Lawrence (1989) argues that terms such as AI and Expert Systems are 'anthropomorphic' in that they ascribe human characteristics to non-human things, 'thereby stimulating a response that is often more emotional than objective'. Hewett (1986) even suggests that ES jargon may actually distract manufacturing companies and prevent potentially useful applications from reaching fruition. Leonard-Barton *et al* (1987) moreover, claim that as well as the 'mysticism' over technology, a further barrier to technology transfer is users' overestimation of precisely what commercial ES products are able to achieve in an organisational setting. As these authors comment, ' Overenthusiasm can sometimes lead to unfounded assumptions and unrealistic expectations. A typical claim is that expert systems help companies "clone experts"; this is overblown. '

c) **Organisation:** Niwa points out the double-hurdle faced by user organisations in first learning about the technology and understanding its capabilities and then attempting to assess their potential within the organisation against actual company problems. How a company undertakes such activities varies according to its structure and internal organisation. Chapter 4 identified a number of organisational strategies which are located between two extremes:-

i) 'End-user computing'; or what Markus(1984) calls a 'DIY strategy' in which individuals or groups within a company avoid contact with the internal computing department and rely instead on external information services and information and personal knowledge in order to develop systems.

ii) 'In-house development'; assumes that the organisation already has the infrastructure to develop and support expert systems projects in a centralised and controlled way. The role of the vendor is therefore minimal. The sensitivity to specific organisational requirements which characterise in-house developments is off-set by the high costs of maintaining a development infrastructure.

Niwa argues that the bulk of sales of shell-based ES products is dependent upon the concept of end-using computing and particularly the notion of user experimentation. Despite the clear association between end-user computing and shells, Markus argues that end-user computing as a development strategy suffers from an introspective view of technology, with needs and benefits defined in technical rather than organisational terms; furthermore, in the longer term the technology becomes outdated or under-utilized because of the failure to integrate the system in the organisation. Many of the organisational problems identified from the User survey, such as a lack of commitment, uncoordinated and informal developments and a failure to operationalise are themselves characteristic shortcomings of end-user computing. It is for these reasons that a centralised but participatory approach to development was forcibly argued in Chapter 4.

An 'end-user' philosophy might also explain a distinct 'hobbyist' approach towards the organisation of projects, although larger organisations tended to be more co-ordinated since they were more able to establish centres of excellence within the company, and manage to cover the overheads. Changes in the marketing of ES products and services in future may have a beneficial effect in defining this technology as 'just another computing tool'. This will facilitate the integration of ES developments with present computer department activities and thereby remove the tendency to view expert systems as a 'laymans' programming tool and alternative to the IT function. The survey shows that the initiative for ES innovation is taken by technologists rather than management. This generates a tension between the technologists understanding of the technology and its capabilities and functional and senior managements' understanding of the organisation and its needs and priorities within an overall business and IT strategy. Experiences in the client organisation, described in Chapter 6., showed that the only way to reconcile these two perspectives was to combine them both within a formal programme of development; but again, not many of the companies were committed or able to support this infrastructure.

d) Integration: Despite users' perception of the importance of information interpretation, presentation and acquisition, the level of integration was only moderate with the most important category of application being stand-alone systems. This underlines the constraints of shell-based ES products in that few offered friendly interfaces and links to software or built-in programming facilities. It also reflects a change in expectations since many of the applications earmarked by users at an 'ideas' stage assume that integration of some form is necessary. However during specification and development phases, the practical technical and organisational difficulties of achieving this dictate that fewer integrations are actually achieved. This situation is likely to change though with the takeup of 'second wave' expert systems defined by Ovum (1988a) which makes integration more attainable through the user of higher-level interfaces and languages.

e) Development Methods and Approach: There appears to be major differences between the methods available at a research level (either within academia, government or market sectors) and those which are required and are possible within a manufacturing setting. The survey shows that the use of specific ES methods was negligible; which although not downgrading the value of the methods themselves, does suggest that the methods and mechanisms of application are inappropriate. This view is endorsed by the findings of Harris *et al* which show that attendance to training courses, seminars and conferences on ES development methodologies is very low. D'agepeyeff and Hawkins also note that perceptions of what expert systems are and how they may be used is 'entirely different' from those in research.

The development methodologies used by respondents to the User Survey were a derivative and customisation of the simple guide-lines expressed in the ES literature, especially the approach offered by Hayes-Roth et al (1984) and described in Chapter 4. This approach provides a basic structure for companies when developing their own 'eclectic' approach. However such guide-lines indicate which tasks are to be undertaken rather than how they should be accomplished. It also says something of the rate of take-up of ES ideas and concepts that industry understanding has progressed little beyond this level of abstraction.

The literature indicates that knowledge acquisition is a major bottle-neck for expert systems development. This view is supported from the survey results which show this to be a consistent problem which, while not preventing implementation, frequently hampers development progress. Indeed many of the problems of knowledge elicitation, representation and verification experienced in developing the Help-desk are also mentioned by these companies, indicating this to be a universally difficult task. Despite this, the take-up of knowledge acquisition tools by respondents was minimal.

f) 'Take-Up': Despite the predicted growths in the ES market (SI:1990; Ovum 1988b) and a strong 'technology-push' culture generated by ES software vendors, the actual take-up of expert systems in manufacturing sectors is low. For instance Gabriel (1989) reports findings that only 22% of UK manufacturers have invested in expert systems. Furthermore, and as the user survey vindicates, the fact that a company has invested in the technology far from suggests that it is being used operationally. Ovum (1988a) attribute this low take up upon a lack of understanding , awareness and confidence in the technology. Where ES applications have been implemented, Teschler (1988) describes a change in attitudes in the end-use of expert systems from being perceived as a 'gee-whiz' technology to becoming 'just another computing tool' to be used when appropriate. Ovum too report that there is a distinct trend towards operational systems which are proven and less ambitious than previously, thus explaining the growth in popularity of application specific software.

Although a funereal outlook may have been inferred from this analysis, there is some comfort from the user survey and other research which shows that expert systems can be used successfully in manufacturing companies for competitive advantage (Bramer). This leads to the question of whether some companies are more likely to be successful in applying ES technology because of structural or cultural factors for example ? Unfortunately there is no research on expert systems use to answer this question: however, it is possible to define which sectors of manufacturing are the most active in developing ES. Ovum for instance found that awareness of the technology was greatest in high technology companies such as computing and electronics, and moreover, that the market-user relationship was more 'mature' than in other sectors in the sense that there was more collaboration and a greater use of integrated applications.

8.3. Evaluating Expert Systems Experiences Using Models of Technology Transfer

The previous section has placed evaluation within a wider context by comparing the specific assessment and development experiences within the client organisation with those of other manufacturing companies. The empirical message arising from this comparison was that although a number of common steps emerged, and there appears to be a set of universal problems which the ES developer must face, the user survey provided little insight in defining a 'best practice' method for ES innovation. Indeed, the user survey was more of a testament to the confusion which abounds industry over the technology's role and means of development, and rather than providing an exemplar or performance standard for the help-desk, it demonstrated how **not** to develop expert systems.

In view of these difficulties, this section broadens evaluation still further by looking not solely at the success with which development is undertaken, but also at the total lifecycle of change from initial knowledge transfer through to the diffusion of information and eventual technology transfer in the organisation. Adopting such a framework of evaluation makes it possible to assess the transfer processes and mechanisms utilized in the client organisation at all stages, from the market down to the ultimate end-user. At a more general level too, it is hoped that in applying these concepts, a model of technology transfer will emerge which may help future developers to appreciate and improve upon the deficiencies in current approaches to ES innovation. Before either are discussed however, it is first necessary to understand what is meant by technology transfer and how it may be applied to expert systems.

8.3.1. From Technology Transfer to Knowledge Transfer

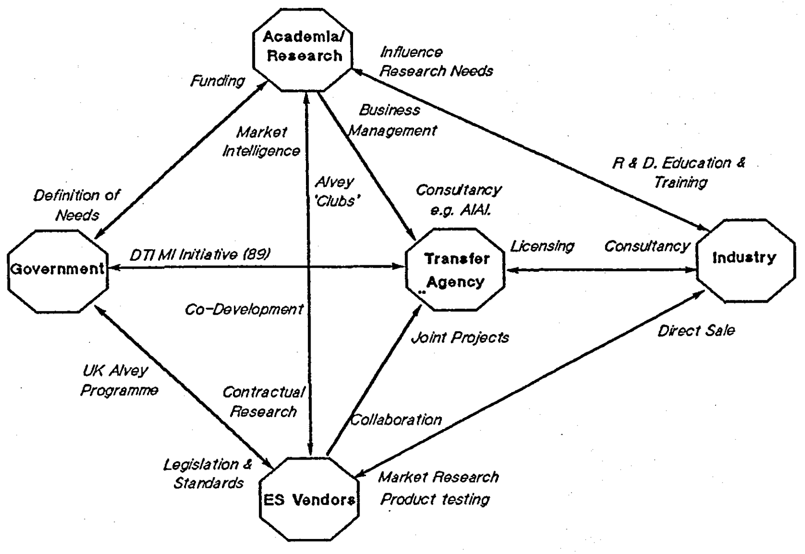
Rothwell (1978) defines technology transfer as the 'flow of new technologies from research to production or research to practical application'. This definition is enlarged upon by Roman *et al* (1983) who argue that technology transfer is essentially a process of diffusion of information: they add, 'technology transfer is the process of collection, documentation and successful dissemination of scientific and technical information to a receiver through a number of mechanisms, both formal and informal, passive and active.' (p159.). Both definitions do not to distinguish between technologies and see no reason to do so. Monk (1987) however, argues that the transfer process for IT (Information Technology) innovations, like expert systems, is unique and shaped by the characteristics of information processes within these systems rather than by the characteristics of the hardware or software alone. As this author comments, 'I.T. comprises both existing products, systems, services and information sets (i.e. techniques) and the principles from which those techniques have been developed (i.e. technical knowledge)'. Although these definitions are useful in defining the transfer process in terms of ideas, concepts, information and services flows and recognise that these are as important as merits of the physical technology, Niwa argues that current definitions are bias towards a supplier/doner viewpoint rather than the recipient. Niwa suggests a 'knowledge transfer' perspective instead which looks at 'what kinds of knowledge are really needed' rather than a technology transfer focus upon 'how can expert system technologies be applied ?'

8.3.2. Expert Systems Transfer Structure

The basic structure of ES transfer at an inter-organisational level is shown in Figure 8.2. There are a number of technology transfer models evident from this structure:-

i) Information Dissemination Model: as the name implies this model concentrates on transfer by improving the accessibility to knowledge of expert systems. An example is the Government's Manufacturing Intelligence programme (MI:89) of providing state of the art reports on ES technology applications, tool/product reviews and vendor co-sponsored competitions for the most 'innovative' applications. There are also a number of library and information services (particularly those associated with universities) that specialise in such information gathering and distribution activities.

Figure 8.2 Expert Systems Transfer: Structure & Components



ii) Technology Transfer Agencies: These are companies set up to facilitate the transfer of ES technology and expertise from one sector to the industrial sector, as Figure 8.2 shows. Transfer agencies may be a front-end consultancy and training service to an academic institution, for example, the Artificial Intelligence Application Institute at Strathclyde University; as an intermediary function between Government and Industry (such as ES technology shops and advisory services and government sponsored projects); as a specialist source of expertise serving vendors; and also as a marketing service to industry for the commercialisation of in-house developments (e.g. Wheat Counsellor by ICI).

iii) Co-operative/Joint Ventures: The accent of this transfer mechanism is on improving company awareness and accelerating the commercialisation of ES technology by involving established companies in the research process. Perhaps the most well known of these is the Government sponsored Alvey programme in which manufacturing companies, consultancies and universities undertook joint research and formed groups or 'clubs' in specific application areas such Banking, Insurance and Finance, and Manufacturing and Design. However, although such approaches have been successful in improving general awareness, research as shown that programmes such as Alvey have consistently failed to transfer knowledge and expertise from supplier to recipient (Whybrow: 1988) so that it is usable.

iv) Licensing: In this approach, knowledge of ES tools and applications developed in a university or company is licensed to a firm for further development and commercialisation. Examples include IKONMAN, a generic process planning expert system developed by British Alcan; and LPA Prolog, an AI programming tool developed originally at Imperial College London. A number of first generation shells were also developed in this way with leading edge companies such as ICL identifying market potential.

v) Direct Sale: the final and most obvious transfer route is direct sale of products and information or services from supplier to recipient. However, this process is manifest through other processes; for instance a recent report identified that most ES consultancies support a specific product range and some have their own product lines (Frost & Sullivan:1990) thus dispelling the notion of the 'independent consultant'.

It is difficult to state which transfer route is the most successful since this clearly is contingent upon the application and needs of participating organisations. However, Ovum(1988) found that the most established and successful route for technology transfer after direct purchase was through collaboration with universities and other academic centres. Even here though, research tended to be technology rather than needs based; as the report states, 'There was a lack of direct and commercial application of academic research...it seems that false expectations of the benefits which academic links are likely to provide is causing unnecessary disappointment with the results.' Indeed, there are a number of inherent problems and weaknesses in these transfer relationships which have been identified from other studies (see Niwa:1988 for example). Common to all these studies, is that such problems derive from the processes of transfer rather than the technology itself, and may therefore be appreciated by first understanding the characteristics of transfer components.

8.3.2.1. Transfer Characteristics and Problems- A Users' Perspective

Some of the problems and characteristics of user organisations are discussed at length in previous sections and are founded on the results of the User Survey and recent research literature in this field (see Bramer:1990, d'Agapeyeff & Hawkins(1987,1988) and O'Neill and Morris:1989 for instance). The main conclusions which are drawn from this analysis may be summarised thus:-

i) User organisations have little understanding of ES development approaches for ES and this is compounded by the lack of 'best practice guidelines' (Born:1988).

ii) ES Lifecycle methodologies such as KADS and GEMINI, described in Chapter 4., are beyond the reach of most (moderately sized systems) users.

iii) The more insightful user organisations have reacted to ES development by applying existing capabilities and known techniques in order to structure and control the development process.

iv) There is a clear neglect of pre-implementation issues such as selection, requirements analysis and justification; and post-implementation factors such as evaluation and maintenance.

v) There is little feedback back to manufacturing industries of those mechanisms which have proved successful for user organisations; other users have therefore relied on suppliers and vendors for guide-lines.

vi) The focus of help for user organisations by the research community is misaligned and suggests a lack of appreciation of the application context.

8.3.2.2. *Transfer Characteristics and Problems- A Suppliers' Perspective*

As above, use may be made of survey work, in this case the Vendor Survey used in Chapter 7 for tool selection purposes, and combined with available literature in this field to appraise the characteristics and problems of ES donors or suppliers.

From the literature firstly, Niwa (1988) argues that the focus of ES suppliers is upon the potential capabilities of expert systems technology rather than on how to use them effectively in user organisations. Furthermore, this author argues the take-up of expert systems is measured from a vendor perspective to denote how many companies are introducing ES tools into their organisation; again, not whether such tools are actually being used in an operational setting. As the author comments, ' In many cases, the purpose of buying such a system is not to use it, but to have it as a status symbol. Even if no such systems are at the demonstration level (and not workable), this fact is seldom understood by users because these systems are not used for practical purposes. Instead they often remain "toys" of technical persons' (pg 148). Although this view may be slightly cynical, it does uncover a tendency to purchase tools because the competition is doing so, without prior assessment of need or value. This view is echoed by other commentators, 'knowledge engineering products, whether in the form of expert systems or AI software engineering tools have been, and continue to be, technology driven rather than market driven products; that is, they emerge into the commercial arena as a consequence of applied or pure research as opposed to being crafted in response to demand of an emerging customer base that desires a specific solution to a perceived need' (Wess: 1988). Wess attributes this problem to two factors: the over-engineering or excessive development of a software tool prior to sufficient customer feedback and experience with the product; and a concentration of effort upon selling the product - Wess observed that some vendors spent on 15-20% of sales on the initial research and development whilst marketing and sales expenses were in excess of 50% (p172.)

Ovum(1988a) provided a more sanguine picture of expert systems earlier, by referring to a 'second wave' in the sale and marketing of systems which are more functional, offer better support and improved facilities for integration. Furthermore, there is a greater emphasis upon proven applications and domains. This is in contrast to what Ovum call the 'first wave' which is characterised by: many small vendors dependent upon single shell products; a tendency to market the technology as being new and different from other computing products; and, with little evidence of operational systems, a reliance instead upon unrealistic claims about capabilities. However, it is noteworthy that this report concludes by stating that the 'hype and oversell continues' (p8), together with the implication that rather than attempt to regulate the transfer process at a market level, the onus should be upon the end-user in affording due care to evaluation and assessment.

The Vendor Survey tends to endorse the notion of a 'second wave' of expert systems, although since most end users have not progressed beyond experience of the first wave, it is difficult to assess the positive impacts in this change of transfer approach. A full analysis of the survey results is given in Appendix XIIc. However, the more pertinent findings are given below:-

i) Consolidation: a number of vendors reported that they were in the process of merging with or acquiring other companies, many of which were established computing organisations. This reflects a double trend in that expert systems are gradually entering the mainstream of IT applications; and moreover, their value is recognised as such by these companies.

ii) Integration: second wave expert system tools make integration to databases and other information systems a much easier and more cost effective option.

iii) Application Specific Software: there is a clear trend in the marketing of ES products away from the 'general problem solver' towards generic or application specific software, such as 'Violet' designed specifically for on-line diagnostic applications (Milne: 1990).

iv) Support: most vendors offer good support facilities. Usually, the more expensive the software the more comprehensive the range of support services available. Environments for example include feasibility assessment, requirements analysis, training and development consultancy within the price of the software. By contrast, suppliers of shells and most AI tools favour 'arms length' support through help-lines and user groups. Although many shell suppliers offer assistance in development, the fee charged is often more than the initial cost of the software. Very few vendors provide pre-implementation support services for problem identification and application selection for example.

v) Delivery: vendors have responded to end-user calls for simple hardware delivery and nearly all shells and AI Tool-kits now operate on standard personal computers or Mainframe systems rather than dedicated machinery. In terms of software, the market is dominated by the sale of low cost rule-based expert system shells. This is also a measure of the level of commitment towards this technology in user sectors.

vi) Development: a presumed benefit of expert system shells is that they allow users and experts to develop applications themselves. However this view is not supported by the vendors of these products. In fact most respondents state that development should be undertaken by either the knowledge engineer or a professional systems developer. As with conventional computing projects, vendors stressed the need to structure knowledge and information and apply project management disciplines.

vii) Applications: although expert systems support uncertainty functions, and are distinguished from conventional systems by this , the most frequently cited application domains such as diagnosis did not require this function. Indeed the most common application domains were also in terms of knowledge representation and control strategy the most straightforward. Milne argues that complex domains such as planning, scheduling and process control are unsuitable for expert systems because of reliability problems.

From the analysis above, there is a clear sign that vendors have become much more realistic about what their product are able to do. The so called 'second wave' of expert systems have a much greater user input in their design and this is reflected by improved integration, interface and delivery capabilities. However, as before, the market remains very strongly product orientated with support and services based solely around the use of the technology. Vendors continue to emphasise the potential capabilities of expert systems without indicating to the mass of end-users how this can be achieved in practice. Furthermore, by offering a standardised product, it is assumed that end-user needs are homogeneous when as the next section shows, each organisation has a unique set of problems and requirements.

8.3.3. An Evaluation of Technology Transfer Models

An analysis of both vendor and user characteristics in previous sections reveals the underlying importance of technology transfer processes and mechanisms. It also suggests the potential value of knowledge transfer models in understanding these processes. Despite this, little research work has undertaken on this subject and that which is available adopts a 'supplier perspective'. Drummond et al (1988) for instance, offer a model of technology transfer for expert systems transfer which is exactly this; it looks at the introduction of the technology without defining a process of evaluation or assessment of needs. It is therefore limited in focus to a uni-directional process of information dissemination and product and technical transfer and certainly not knowledge transfer in the sense of a two-way process of evaluation. In turning to general models of technology transfer however, it is evident that the same technology push focus predominates. Work on transfer models currently being carried out at Cranfield (see Cordey-Hayes:1990 and Lefever:1991) suggests that models of technology transfer fail in two respects:-

a) A disregard for the processes and mechanisms by which a technology is transferred from a donor (vendor, academia, intermediary etc) to the recipient (in this case user organisations in the manufacturing sectors)

b) A failure to recognise the significance of recipient's needs and address service delivery aspects in the knowledge transfer process.

In response to these shortcomings, an evaluative framework was devised made up of three components: accessibility, mobility and receptivity. These are defined below from an end-use perspective and used to evaluate the knowledge transfer processes.

a) Accessibility

In the context of knowledge transfer, accessibility refers to the source, availability and physical proximity to information, products, services which facilitate the effective transfer of explicit knowledge (e.g. manifest through physical systems and formal processes) and tacit knowledge (manifest through ideas, concepts, influences, culture and other informal processes). As well as defining an opportunity, accessibility also defines the means by which knowledge is presented and made available to a defined or intended user. Accessibility may be active, such as the marketing of a product in which case there is a commercial and economic value associated with optimising the channels, processes and mechanisms of accessibility. It may also be passive, in the sense that a person may chance to read a report on expert systems in a computing journal (this was the next most important source of accessibility in the client company after 'aggressive' vendor marketing). Accessibility is principally a presentation issue (e.g. the four P's of marketing : product/technology & capabilities; place/distribution; price/costs; and packaging/marketing) and provides no measure of how or why this knowledge is used. While accessibility is necessary in understanding knowledge transfer, it is not sufficient in describing the totality of the process up to and within the boundaries of the user organisation.

b) Mobility

A second phase of evaluation which redresses the imbalance between technology-push and demand-pull is an analysis of mobility, i.e. the processes and mechanisms which define the movement of knowledge between supplier and recipient at any given level. Mobility may therefore refer to the transfer of knowledge between organisations, from transfer agencies and vendors to user organisation for instance, or within organisations between groups or functions and individuals, e.g. the migration of employees from one sector to another each taking knowledge and ideas with them.. The latter is particularly relevant in the case of the author's role in the study as transfer agent. Where accessibility is concerned with the *structure* of knowledge transfer, mobility looks more at the formal and informal *processes* of transfer between and within organisations with preeminence attached to how knowledge should be transferred and delivered in an appropriate way rather than simply addressing what opportunities might arise from using the technology.

c) Receptivity

Receptivity is the critical issue in ES assessment and development and refers to the ability and readiness of user organisations to act upon knowledge (tools, information, ideas methods, techniques etc) and accept the subsequent changes which are likely to arise as a result. As with mobility, receptivity operates at different levels. At an organisational level for instance, earlier work has shown that high technology companies are more receptive to ES ideas than other sectors, but there is no evidence that they are any more successful in exploiting these ideas in order to produce operational systems. Receptivity may also be a function of individual characteristics such as leadership skills, political influence and so on. For example, the ES literature continually reminds readers of the need for a product champion, a willing and articulate expert and co-operative users. Moreover individual qualities may extend to influence group behaviour and even shape corporate culture and technology policy (see Peters & Waterman: 1982). In that individuals are also essentially political and have unique viewpoints, receptivity cannot be understood in single dimensions alone. In this sense, Linstone's Multiple Perspective concepts may be applied in order to understand receptivity at different viewpoints and organisational settings.

Like Linstone too, the three components of technology transfer Accessibility, Mobility and Receptivity (henceforth referred to as AMR) are not mutually exclusive but integrated. Furthermore, although there is a suggested sequence, it is critical to evaluation that transfer processes are viewed from all three dimensions interdependently acknowledging that each component should influence the way in which the other operates. A problem in expert systems knowledge transfer for example is that issues of receptivity are not considered when defining methods of transfer and mobility within the organisation.

In using the AMR framework it is important to define a perspective and direction of transfer. This is to prevent confusion over definitions; for instance 'receptivity' from a vendor perspective may be 'accessibility' from a user perspective. This framework is used from an *end-user perspective* in evaluating the effectiveness of knowledge transfer processes and mechanisms at both industry and company levels. Despite this rejoinder however, the AMR structure should be seen as being self-replicating between levels. For example, when evaluating organisational receptivity it is a valuable exercise to evaluate AMR processes at group and individual levels.

8.3.4. An Evaluation of Knowledge Transfer Processes Within the Client Organisation

This section looks at issues of accessibility, mobility and receptivity within the client organisation and from three broad fronts: first, from an evaluation of the author's role as transfer agent; second, from an evaluation of transfer processes and mechanisms ; and finally from an evaluation of organisational receptivity. The latter looks at two issues in particular, the effectiveness of the help-desk in knowledge transfer and the impact of organisational factors on diffusion effectiveness.

8.3.4.1. Evaluation of the Transfer Agent Role

If 'technology does not move of its own accord, people move it' as Bradley (1988) states, then individuals clearly have an important role in the transfer process. A related question therefore, is in what way people affect knowledge transfer ? The author's role, both formally and informally was as facilitator or transfer agent, although the focus shifted more to that of developer at later stages. As a formally recognised transfer agent, a major function was to promote expert systems technology by improving the mechanisms of accessibility and providing a framework which would allow for its effective use in the company. This role was intended to supersede informal processes of knowledge transfer, through vendor promotions and trade literature for example, and also more formal processes such as technical enquiries by company personnel. The latter often identified a potential for the technology at their functional level; however, by their nature these applications tended to be unsupported by the company and 'technology driven' . It was therefore intended to formalise information accessibility in the company but within a framework of organisational evaluation.

As transfer agent, the author was expected to liaise with external sources, principally suppliers (vendors, consultants, academia) and Cranfield as the supporting institution; and also operate within a well defined structure and hierarchy of the client company. The dual focus upon external and internal mobility is shown in Figure 8.3. This figure is important because it helps to distinguish between the knowledge and skills required by the transfer agent in achieving knowledge transfer as an objective, and the knowledge and information required by the user organisation as recipient in order to satisfy organisational and business objectives. The agent's objectives referred more to the processes of transfer i.e. the design of a programme of technology transfer, evaluation and development; while those of the organisation were more specific and could be expressed as deliverables - hold awareness presentations, identify appropriate ES applications and develop an expert system for example.

For the transfer agent, there was a double learning curve; firstly in gaining an understanding of the organisation, its structure and culture, working practices and operations, and problems. This process benefited significantly from the use of IDEFo as a legitimate means of gaining access to company wide information. A second personal learning- curve was in acquiring knowledge and skills of ES technology. As there was no expertise within the company, the author was dependent upon external sources (mainly through the ES school at Cranfield, vendors, and research literature as 'knowledge suppliers').

Figure 8.3 shows the layered structure in which the author operated from the core. Chapter 6 stressed the value of mobility within the computer department in order to exploit the existing service network and communications structure it had with the rest of the organisation. Figure 8.3. also shows though, how from an external viewpoint, the transfer agent directly interfaced with knowledge suppliers at an organisational level in an attempt to filter and channel information on ES before determining how it should be used within the organisation. Clearly informal processes of transfer could not be prevented, but it was hoped that by offering a formal structure of technology assessment and source of knowledge operating within a known entity, the computer department, informal processes would have less significance and therefore be used less frequently.

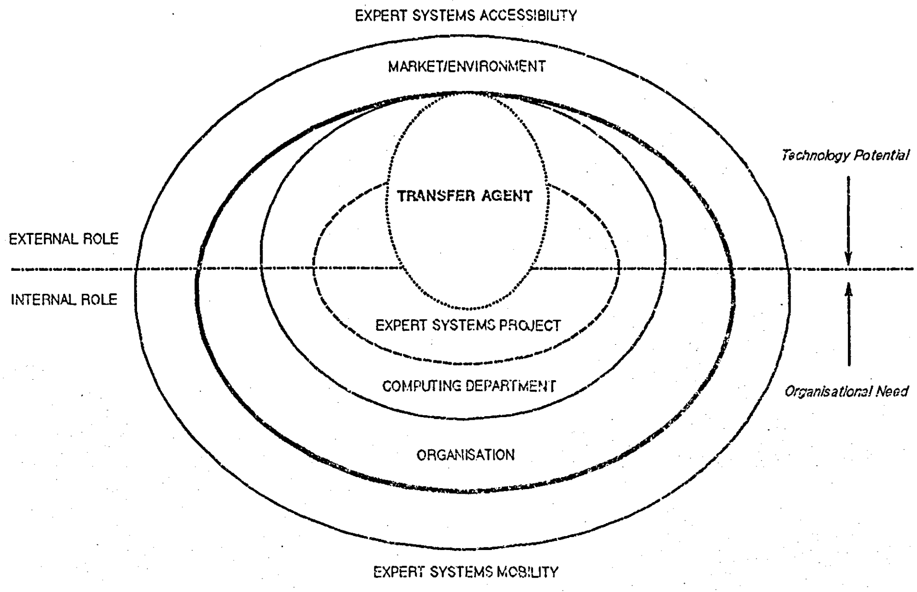


Figure 8.3 : Accessibility, Mobility and the Role of the Transfer Agent

It can be seen from Figure 8.3. that knowledge transfer operates at different levels from issues of accessibility at a market/environment level to mobility and receptivity issues at the organisation and group (e.g. computer department and ES project team) levels and ultimately at an individual level (e.g. transfer agent and end-user of information). At each level, different transfer processes and mechanisms are required. For instance, at a personal level informal processes are effective whilst more structured and formal methods are demanded at an organisational level. Distinct support functions are also necessary: for example at an organisational level, active senior management support (i.e. project champion) was essential in validating the author's role. At a group level however, this support becomes passive and is absorbed within the authority of the computer department.

Before mechanisms of transfer can be defined, it is useful that the existing organisational processes are understood. MPC as an 'enquiry system' proved useful in understanding not just components of a situation in terms of settings (i.e. personal, techno-personal, technical etc), but also a sensitivity towards *how* a problem was being regarded. Having defined a context for knowledge transfer, it is easier to define appropriate formal and informal support mechanisms. As a new technology, the author adopted a major T (Technical) perspective effort at an accessibility stage in order to understand the concepts behind the technology and its potential. However in evaluating its use within the organisation, O (Organisational) and P (Personal) perspectives should determine and shape the evaluation and development approach taken.

8.3.4.2. Evaluation of Transfer Processes and Mechanisms

The User Survey and other studies mentioned previously have shown how current ES methodologies have not been adopted in manufacturing sectors. Chapter 4 describes the weaknesses of these approaches in detail, but in short, they tend to require significant resources to use; are orientated about the design and development of systems only; and provide no support during pre-implementation stages. Of those respondents that actually followed a coherent approach, there was a tendency to adopt tools and methods which were familiar to their organisations rather than adopt dedicated ES tools. A similar approach was adopted in the client organisation although there was the additional requirements that the framework was used for evaluation and assessment purposes as well as purely development.

The design of the framework used in the client organisation was based on the work carried out in Chapter 4 and is summarised in Figures 4.5. In its 'development approach', it distinguishes between the tasks necessary for ES assessment and development (called the activity framework) and the actual processes(formal/informal) and mechanisms(hard/soft) which allow such tasks to be performed according to the properties of the organisation. Although current ES methodologies take cognisance of the first requirement, they are too prescriptive and complex to allow for the second. By contrast, many of the respondents in the user survey adopted development approaches which mirrored the organisational situation but were deficient in that they overlooked the task requirements of ES evaluation and development. In order to achieve both sets of requirements, technical and organisational, whilst recognising what was physically possible for one person to achieve, and also what was politically acceptable to the company, the framework that emerged from these constraints is shown complete in Figure 8.4 . The hard activities, denoted by boxes, indicate formal activities undertaken within the company and provide formal and tangible deliverables to the knowledge transfer process. Examples of hard activities from Figure 8.4. include the composition of the IDEFo model and use of the Managers' check-list as organisational functions, and development of the help-desk and associated tasks as technical functions. Soft activities denoted by the circles in Figure 8.4. indicate informal processes which usually derived from formal tasks and provided the author with the opportunity to carry out and contribute additional information to the transfer process. Perhaps the most graphic example of this is the way in which the *process* of modelling the organisation using IDEFO provided knowledge of the organisation, its problems and business needs and access to company resources and people which would be difficult to acquire and justify through formal means.

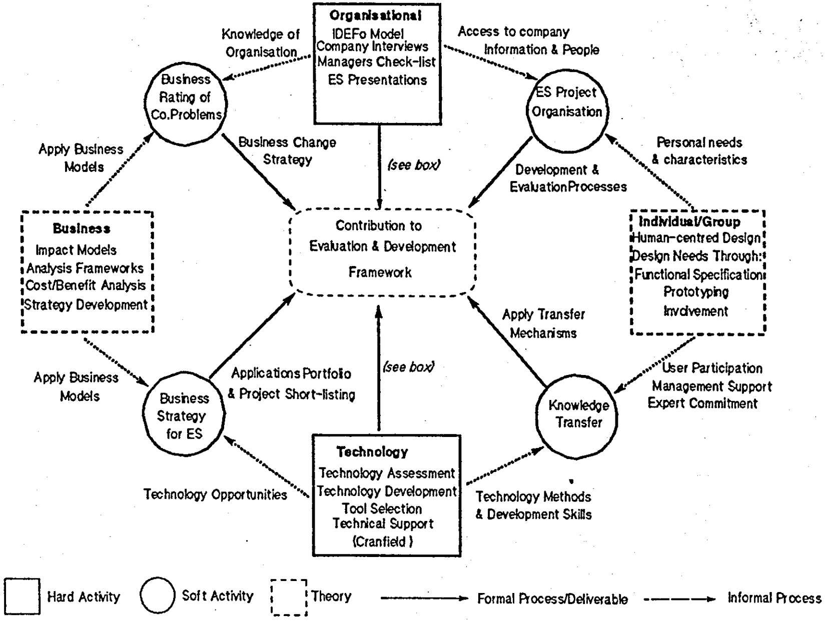


Figure 8.4 : Framework Processes (Hard/Soft) & Mechanisms (Formal/Informal)

In general, soft activities were of direct benefit to the transfer agent while hard activities were of specific use to the organisation. Formal deliverables could emerge from soft processes however, an example being the Applications Portfolio shown in Appendix VI. The dotted boxes denote external knowledge acquired and made available to the transfer agent (such as business and human factors theory and concepts acquired whilst at Cranfield) which are applied indirectly to achieve certain goals within the company For example, business impact models were used informally to prioritise company related problems identified through the IDEFo study.

In some cases, soft processes failed, and required formal organisational support and recognition to be effective, thus transforming them to hard activities. An example is the design of the development check-list which, from a collection of ideas and informal rules for use by the author, was transformed to state explicitly those criteria which should be used in the assessment of potential ES project suitability. This was necessary to ensure evaluation knowledge was retained within the organisation after the departure of the transfer agent.

8.3.4.3. Evaluation of Organisational Receptivity

Receptivity is a function of the perceived appropriateness of ES knowledge. At an organisational level, receptivity to ES may depend on the way it was perceived to fit existing product and service capabilities and critical business activities as well as more immediate measures of financial return. At an individual level by contrast, receptivity may be a function of personal acceptability. As well as levels, receptivity is also a function of the nature of transfer. In the client organisation , receptivity was evaluated therefore from two basic viewpoints: with respect to the use of the help-desk as essentially a 'product' transfer; and use of the development framework as an 'information transfer'.

a) Help-desk Receptivity

A justification for developing the help-desk was as a mechanism for improving awareness and understanding of ES, but with a proviso that the problem selected was worthwhile and would provide a tangible cost saving. Although the help-desk succeeded in communicating the concepts and potential of the technology, its impact was localised in the company, and alone had a limited effect. Figure 8.5.shows the value of the Help-desk as a mechanism for knowledge transfer at different organisational levels. The darker the shading the greater the receptivity and diffusion of knowledge. Figure 8.5 shows that diffusion for the help-desk was contingent upon association with the system. Thus for people directly involved in the project knowledge transfer was concentrated: moreover, the processes of transfer were generally informal and interactive.

At the level of the Computer Department receptivity was again quite high since most members of the department would be affected in some way by the help-desk and therefore their interest and support in the project was essential. Accessibility to the help-desk was also high and during operations most members of the department formally or informally made use of the system. In this respect, the help-desk was an effective transfer mechanism of knowledge of the technical domain as well as ES concepts. At an operations stage, the help-desk raised the level of interest to the extent that it provoked new ideas and applications by computer department staff at both sites. Furthermore, the basic help-desk shell was used to support a smaller second application at the Preston site developed by a member of the computer department and was therefore used as a basis for skills transfer also.

Beyond the level of the computer department, a distinct group within the organisation was future users of the help-desk, collectively known as the user community in Figure 8.5. For this group, receptivity was expressed in terms of the demonstrable use of the help-desk as a diagnostic aid. In terms of knowledge transfer this was limiting since this group associated expert systems with trouble-shooting rather than an approach with diverse capabilities. Furthermore, a large section of the user community were not aware that an expert system was being used and only noticed a small change in the nature of the trouble-shooting *service* being offered to them.

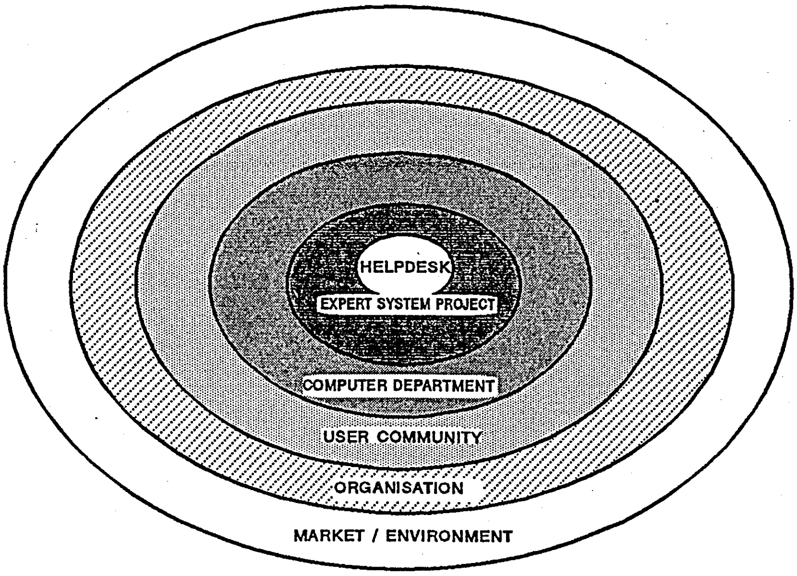


Figure 8.5: Knowledge Transfer Receptivity and the Boundaries of Diffusion

The impact of the Help-desk at an organisational level was marginal. Although as a first project, it was intended that the system should not have major business consequences as a 'support' function, it was hoped that awareness of the project would be greater than it was. Similarly, there was no impact at a market/environment level other than details of the help-desk being used in vendor promotional literature.

An important pre-requisite for knowledge transfer is what Morieux and Sutherland (1988) define as an 'appropriate organisational culture '. Although the computer department was an appropriate environment for evaluation and development, it was perhaps less effective as a domain for a first project because its departmental culture was distinct and noticeably different from the rest of the organisation. As well as being physically separate , the staff language and behaviour was different and interaction with the rest of the organisation was on the basis of providing a technical service. As a consequence, other functions in the company tended to label the computer department as being 'different' and therefore tended not to equate the benefits and potential of the help-desk technology with their particular business area, thus diminishing its value as a transfer agent.

b) Receptivity Towards the Development Framework in the Client Organisation

The development framework had an immediate effect upon individuals in the ES project team in that their direct participation was critical to its success. The transfer of ES ideas, methods and techniques was therefore concentrated and after a year of use two project members in particular became competent in ES assessment and development. There is always a potential danger in the concentration of organisational information to a few key individuals because the company is left in difficulty should these people leave the organisation. One of the benefits though of constructing a development framework is that it reduces this dependency by providing organisational access to a 'knowledge base' of information. This was especially important in this study because the author's involvement in the company was restricted to two years. Furthermore, since there was no current priority attached to developing expert systems (because of a back-log of projects and organisational changes and restructuring), it was necessary that the framework addressed future needs when the situation might be reversed.

At a group level, there were useful exchanges of information between the computer department and the author; the former showing interest in the prospects of integrating ES tools with current computer equipment and of finding out more about the validity of ES methods such as prototyping in conventional systems development. Similarly many of the skills of the computer department (software engineering, project management, maintenance and systems support for example) could be applied to the ES development programme.

The development framework was more effective than .the help-desk at the organisational level because there was a wider choice of formal methods of transfer such as presentations ,interviews and check-lists, and informal methods such as the process of IDEFo. Despite this, diffusion using these mechanisms was low at the end of the study because these earlier attempts at raising general awareness and understanding were not exploited and therefore the impetus was lost. This underlines a key feature of knowledge transfer in that it is a dynamic process.

8.4. A Model of Knowledge Transfer for Expert Systems Innovation

The justification for developing such a model is embodied in the calls for change made by Dawson (1988):-

> "Within contemporary debates a great deal of attention has been given to the implementation and initial operation stage of technological change. Little attention has been given to developing conceptual frameworks for explaining the process of technological change, from the initial decision to invest technology through to the routine operation of a computer aided operation. " (pp 57)

The purpose of this section is to make use of the lessons and experiences at both personal and industry levels in order to develop a model of knowledge transfer for expert systems which fulfils the above need. The basic Accessibility-Mobility-Receptivity (AMR) structure described in earlier sections provides a useful basis for a model of knowledge transfer shown in Figure 8.6.. There are additionally further requirements and attributes which have been incorporated into the model's design and use. These are described below:-

a) Levels of Knowledge Transfer.

With an undue emphasis upon accessibility in technology transfer models, the focus of transfer analysis is at the organisational level and especially at the interface between organisations. Although this is important, West and Farr (1989) argue that it does not consider the importance of individual and social mechanisms which are 'instrumental to the transfer process'. Indeed Peters and Waterman (1982) argue that individual characteristics shape group and organisational forms. Therefore the model shown in Figure 8.6. adopts four principal levels of transfer: *market/ environment* (which comprises all sources of knowledge, products and services external to the organisation); *Organisational*; *Group*; and *Individual*. Clearly a vendor will have organisational, group and individual characteristics; however the use of the model is from a user-organisation's perspective.

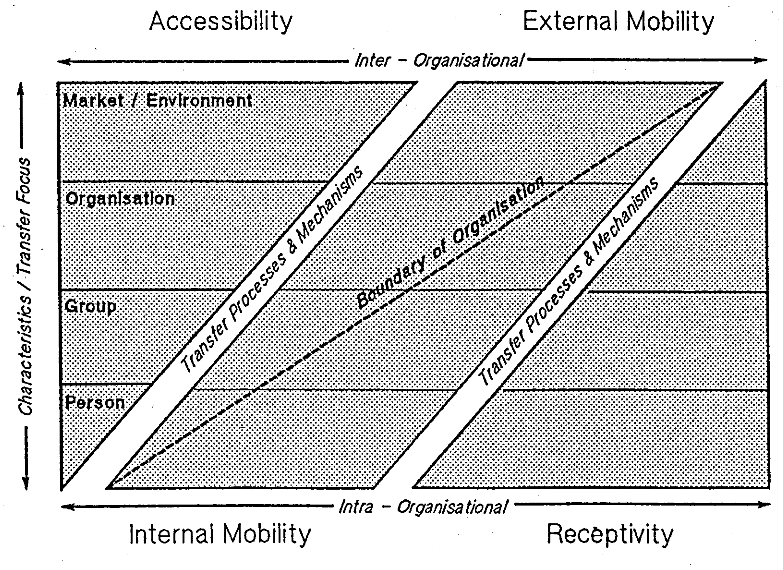
b) Characteristics of Transfer.

Weitz (1988) notes the importance of understanding corporate culture, company policy and values in shaping the knowledge transfer process. However this is true not just at the organisational level but at all levels and therefore the model defines *characteristics* of recipients and donors at each of four levels affected by transfer as well as transfer interactions. For instance at a market/environment level, characteristics may include attributes of the products, services, marketing strategy, size and customer base; and structure and role of government, research and academia. At an organisational level, characteristics may included culture, company working practices, structure and size and style of management. Whilst at an individual level, characteristics may include personal qualities such as leadership and communication skills and technical competence for instance.

c) Transfer Processes and Mechanisms

As well as defining the characteristics of components of transfer, the model in Figure 8.6. indicates the nature of transfer in terms of *processes* and *mechanisms*. In the same way that the development framework described in Chapter 4, distinguish between formal and informal methods of development, so the model should recognise the formal and informal processes and hard and soft mechanisms of knowledge transfer . The purpose of knowledge processes might be communication, training, diffusion, technology transfer or information dissemination for example. Possible transfer mechanism include technology, people, methods and entities (organisations groups etc) within and between levels.

Figure 8.6: A Model of Knowledge Transfer for Expert Systems Innovation



Processes and mechanisms also have different value attached to them according to their level of use in the organisation. For instance at a group level, e.g.. project level, passive support from the project champion is acceptable but for the project to succeed there must be active support from the expert and end-user. By contrast, at an organisational level, a champion is critical to gaining organisation support and access to resources. This is shown with respect to AMR in Figure 8.6. Thus during accessibility, the transfer focus is upon the characteristics of the market and its interrelationship with the user organisation; group and individual characteristics are less important at this stage. However, as transfer progresses from issues of accessibility and mobility between organisations to consider the movement and use of knowledge within a company the transfer focus shifts to consider the individual and group characteristics.

d) Types of innovation

The model should also be able to accommodate different types of innovation as well as technical. These may include: organisational innovations brought about by a change in culture or approach such as the concept of 'knowledge sharing' for example; administrative innovation (such as re-organisation of an office or department); and personal/group innovations such as re-training, job rotation and quality circles.

e) 'Emergent' Model of Technology Transfer

From the diversity of methods and approaches used by respondents to the user survey this suggests that there is no 'one-best' way to enact knowledge transfer. Rather, the model should be used to communicate that necessary transfer processes and mechanisms emerge formally and formally in a way specific to organisations. This is why this study emphasises the use of 'frameworks' and 'approaches' rather than pre-determined methodologies.

f) Intra- and Inter-Organisational Factors

The implementation and operations of ES is, as Markus states, often a political process of negotiation at an organisational level between vendors or other external sources and the potential user company. The effectiveness of transfer across this 'boundary' is dependent not only on the objectives, assumptions and values of those who make decisions about its potential role and use in the company , but also on the processes of implementation and knowledge transfer. A .distinction is therefore made in the model between the external processes of mobility which enable knowledge transfer up to the boundaries of the organisation, and internal processes of mobility which determine how knowledge is made available for use within the company.

g) Enhancing the Model using Multiple Perspectives Concepts (MPC)

The boundaries between accessibility, mobility and receptivity and between levels also marks discrete changes in perspective and settings mirroring changes in roles and viewpoints. Thus, it is possible to complement the model by learning more of the transfer situation using Linstone's Multiple Perspective Concepts.

8.4.1. Using the Knowledge Transfer Model

As with MPC, the use of the model is as a communication device and mechanism for understanding the knowledge transfer situation and the context of change. In that it is an empirical model, it is the product of personal experiences and those of a sample of companies in diverse manufacturing sectors. It may thus be used to explain why, first of all, knowledge transfer is so difficult and accounts for the present difficulties experienced in manufacturing; but more positively, it may be used to identify appropriate knowledge transfer processes and mechanisms which satisfy organisational needs and characteristics.

The model shown in Figure 8.6. shows a shift in the importance of levels and subsequent processes and mechanisms. At the phase of accessibility, the necessity of providing information and communicating technology potential between external sources and the recipient organisation makes this phase insensitive to group and individual characteristics. Instead, the transfer focus is more concerned with the external attributes of the recipient such as company profile, position in market, type of manufacturing sector, and the company's catalogue of current technologies for instance. Furthermore, individuals and groups involved at this stage function at the same level of abstraction. However, as the knowledge transfer process progresses beyond the interface between donor and recipient (marked by the dotted-line in Figure 8.6.), then this role is reversed as the transfer focus shifts to issues of internal mobility and receptivity. Here, it is necessary to embody an 'internal view' of the organisation, which IDEFo sought to provide in Chapter 5., by understanding the formal and informal dynamics of group and individual interactions (e.g. politics, communications, conflicts etc) and frameworks of behaviour (e.g. culture, policy, strategy, procedures, etc). These determine the 'reality' of mobility and actual use of the technology within the organisation). Figure 8.6. shows that such individual and group characteristics are critical to achieving a desired state of receptivity. Moreover, by focussing transfer at this level, the change process can be managed more effectively so that this state is attained.

The model distinguishes between two forms of mobility. External mobility is concerned with providing an organisational framework for technology assessment at this level and supporting a planned use of mechanisms and processes which allow the transfer of a technology or technology concept such that it is the intellectual or physical property of the recipient organisation. Here the critical interface is thus between donor and recipient. By contrast, internal mobility addresses how the technology may be adapted or applied within the organisation and is therefore committed to providing the internal capabilities for exploitation and development. At this point, property of knowledge, whether manifest through a technology or concept, is assigned to a particular group or function in the organisation that has the specialist skills or other attribute, such as political leverage for instance, to advance the transfer process in the organisation.

At various stages in the transfer lifecycle, key individuals and groups emerge as being critical to the success of the project. Furthermore, the boundaries between levels and phases, shown in Figure 6.8, indicates that the actual value of each across the transfer process will vary. For example, at an ideas stage (assessibility), an individual may have the technical awareness and acumen to recognise a technology's potential but not necessarily have the organisational (mobility) skills to add credence to its use in the company. This is precisely why in many of the organisations that responded to the user survey, the diffusion of the technology was highly localised and the development process itself was bottom-up and 'distinctly hobbyist'.Many of the projects were also dependent upon key individuals in facilitating knowledge transfer. For instance, the user survey showed that in 15% of projects, the developer was also the expert and the user. There are problems in this in that the movement of knowledge becomes localised to a particular group or individual. Moreover, where diffusion does take place, the knowledge of expert systems may be associated with a particular application by recipients, such as diagnosis, and therefore diminish the full potential of the technology. As D'Agapeyeff and Hawkins comment too, overdependence upon single individuals leads to a lack of feedback and a critical view of the system such that design weaknesses are carried through into operations.

8.5. Conclusions

There was no suggestion from survey work and other research that some sectors of manufacturing are more competent at developing ES, although there were differences in the state of understanding and awareness. Common to all sectors is that problems encountered derive through a basic lack of attention to the processes and mechanisms of development rather than the technology alone. To understand these problems requires that first the characteristics of the actors of transfer are understood. In the case of ES suppliers, this chapter has shown that despite improvements to the usability of the technology (better functionality, better interfacing and integration facilities for example) of so called 'second wave' expert systems, the problem still remains that the vendor service structure strongly favours the support of technology products rather than supporting and helping to define the needs of user organisations. Furthermore, the use of consultancies and intermediaries as the most likely source of this help in the market, is discredited for the reason that many actually vend particular ES products or applications rather than offering a truly 'independent and objective' service to users. Collaboration with academia and research as a potential solution to this failing of the market too has its problems. There appears for example, to be a divergence between what university research believes to be useful in terms of knowledge development tools and methods and what industry is ready to try. Indeed, one of the more significant findings from the user survey was the lack of use of ES development methodologies and techniques. There are two probable reasons for this: firstly, and related to the above, manufacturing companies have found these methods too esoteric and complex, and demand significant resources to implement: given that the level of commitment for many users does not go beyond the demonstration stage then such demands are unlikely to be met. A second reason is that companies in the survey have tended to adopt their own methods based on a hybrid of simple ES concepts and existing computing and development methods and capabilities. In nearly all cases however, there is a neglect of pre-implementation issues such as problem selection and application specification. Moreover, less than 5% of respondents made use of business planning and development tools.

Organisations also showed a disregard for post implementation issues such as testing and evaluation, but particularly maintenance. Perhaps the greatest uncertainty and confusion displayed by respondents was in defining maintenance needs and costs. In fact, 20% of the sample indicated that they were unsure how to go about performing maintenance work or were willing to allow the system to go 'static'. Operational experience in using the help-desk revealed how easy it is to do the latter without a formalised structure and plan for maintenance. In terms of responsibility , both the survey and experiences with the help-desk testify the difficulties of delegating maintenance tasks to end-users and experts and therefore it is recommended that this role is taken up by the specialist knowledge engineer or systems developer.

A principal finding, from a receptivity perspective, is the lack of research which addresses how expert systems are actually being used. In this sense, the User Survey provided a useful and important means of analysing receptivity needs. However this only provided a static and post-hoc evaluation of these needs when in fact for effective change, the transfer structure should nurture a situation whereby user organisations constrain the transfer process. Clearly however, if knowledge is used for commercial advantage by a company, it is unlikely to be shared: such barriers prevent feed-back into other companies and the market thus retarding the change process. The diffusion of ES knowledge, generally in manufacturing is low and this is reflected by the survey which shows that most respondents have progressed little beyond the 'demonstration of capabilities' phase. Constraints which have prevented further development such as a lack of awareness of technology and benefits and lack of support, again suggests a failure to consider the choice of appropriate knowledge transfer processes and mechanisms.

The reality of expert systems use is that applications are relatively straightforward (they are stand-alone, use basic hardware and rule-based shells for instance) and tend to be used in proven domains. This also reflects a reluctance to commit more substantial resources towards ES development until the technology is proven. The benefits of ES are expressed most strongly in added-value terms, such as improving the accuracy of decision-making and quality of work, and increasing problem solving ability rather than in cost reduction terms. An important role in enforcing a balanced organisational perspective on the role and value of ES is the management function inside companies. This is evident from the survey which shows that many potential projects under the responsibility of technical staff were prevented from being implemented because management failed to see their organisational value. This conflict would not arise if management were involved from the onset: instead development of expert systems tend to be undertaken by 'enthusiasts' and therefore separated from the company's mainstream IT and computing activities.

The principle of knowledge transfer in preference to technology transfer was that it implied a bias towards receptivity issues defining the types of knowledge required by the company as an organisationally driven activity rather than the prevailing market driven tendency towards suggesting how the technology should be applied. This technology focus is heightened by the fact that the initiative for ES innovation is more often taken by technologists rather than management in an organisation. Experiences in the client company showed that in order to combine both perspectives it was necessary to construct a formal structure and programme of evaluation and assessment. However it is clear from the user survey that this level of commitment from respondents is unattainable because of the overheads required to support this structure. As well as a bias towards the technology, there were problems of over-dependence upon key individuals such as the expert or user and therefore awareness and knowledge diffusion in the organisation tended to be localised to a particular function.

There are three dimensions to knowledge transfer: firstly, 'phases' of knowledge transfer which focus upon issues of accessibility, mobility and receptivity; second, discrete levels of knowledge transfer (market/environment, organisation, group, individual); and third, attributes of knowledge transfer which considers the characteristics, processes and mechanisms of knowledge transfer. The interactions between dimensions provides the basis for a model of knowledge transfer which may be used to help organisations plan and manage the transfer process. The model may be enhanced further by learning more of the transfer context and its settings at a particular phase or level by using the conceptual framework of MPC.

The knowledge transfer model and more generally a knowledge transfer perspective highlight the need to examine the processes by which organisations respond to and are influenced by new technologies rather than to concentrate analyses exclusively on either the technological or organisational determinants of change as research studies and disciplines appear to do. On this basis , criticism can be directed at the present structure of expert systems transfer at a market level particularly. Programmes such as Alvey (and more recently Esprit), the Government's Manufacturing Intelligence Initiative, and the general orientation of ES suppliers and intermediaries, pay little attention to the problem of organisational mobility and receptivity in the introduction and use of expert systems. This is ironic given that the function of these programmes and organisations pivots on demonstrating the value of ES in manufacturing.

The effectiveness of knowledge transfer in the client company was measured with respect to receptivity of the help-desk as a transfer mechanism, and the development framework in providing a knowledge transfer structure. It was found that although knowledge diffusion was at its most concentrated about those people and groups directly involved or affected by the helpdesk's development, its value as a transfer mechanism at an organisational level was limited. By contrast, the worth of thedevelopment framework was that it provided an appropriate structure and discipline of transfer which allowed it to be used at this level. As well as levels, consideration must also be given to the time scale of transfer, and, although 'product' based transfer was the most cogent, its impact was ephemeral. Conversely, a great benefit of the framework is that it can be re-used and continually applied in the longer term, thus establishing a continuity of knowledge transfer in the company. Ideally, both mechanisms should be used to complement the deficiencies of each other.

The concept of receptivity is useful in distinguishing between technology innovation (as an accessibility issue) and organisational innovation. The latter in particular gives credence to the political and social processes of change. Thus, although by virtue of the structure of industry and the nature of competition, an organisation is bound to adapt to changes in technology, expert systems being only one of a string of new information technologies, the actual changes that take place within the organisation are more a product of its culture and environment than a reflection of the enabling characteristics of the technology itself.

Chapter 9.

Summary, Conclusions and Further Research

9.1. Introduction

The purpose of this chapter is first to summarise the principal themes of this study from which a number of conclusions may be drawn. It outlines too the main contributions that this study has made in the expert systems field. From this, a more detailed evaluation is undertaken of the major components of technology assessment, development and evaluation used in the study. These include the following:-

i) An evaluation of MPC as a conceptual framework for technology assessment and development.

ii) An evaluation of the Development Framework as a methodological basis for technology assessment and development.

iii) An evaluation of IDEFo as a tool and process for modelling the organisation.

iv) An evaluation of the selection, development and operations of the Help-desk.

v) An assessment of the evaluation process itself and the viability of the knowledge transfer model.

These mechanisms are judged with specific reference to the study and its requirements, but also in terms of their relevance to technology innovation in general.

Finally, in accepting the constraints in the scope of this study, this chapter concludes by suggesting further research work. It also provides the opportunity to voice some of the wider philosophical issues and concerns which emerged from this study which ties in this author's personal viewpoint and draw upon his earlier published work. In generalising in this way, discussion looks to the use of other conceptual approaches, and in particular, the contribution of human-centred concepts as a benchmark and motivating force in future technical change.

9.2. Research Summary and Conclusions

An expert systems programme was defined from which an application was identified, designed, developed and implemented to the satisfaction of the client organisation. The assessment and development process introduced novel approaches to examine user and organisational needs and to consider wider issues in innovation and technology transfer. These new approaches moreover, enabled a critical, though constructive, evaluation of current expert systems development to be formulated.

During the substantive work, methodological contributions were made in the application of IDEFo, as a precursor to technology assessment, and MPC, in areas where they had not been applied before, with the latter being extended in concept.

From this study, a new model of technology transfer has emerged as a useful evaluative framework for understanding and planning the technology innovation process. As with many of the tools used in the thesis, although they have been applied to expert systems in particular, it is the belief that most are valid for all information technologies. As well as these contributions to expert systems research, and strengths of the study, it is also important to recognise its weaknesses and limitations. Thus, the remainder of this chapter provides a critical appraisal of the various methodological tools and processes used, beginning with a summary and main conclusions to the thesis.

Common to the three themes of this study, technology assessment, development and evaluation, is the technology itself, expert systems. As with decision-support tools in the 1970s and relational databases in the 1980s for instance, there has been a tendency to overstate information technology potential. Although it is necessary to recognise the enabling properties of expert systems, it is more critical to understand that they are simply 'tools' and their value and effective use ultimately lies with the user organisation in managing the innovation process properly. A multiple perspective approach reduces the uncertainty of change by providing a greater understanding of all dimensions of the business problem, and by identifying different organisational needs and human processes throughout the life-cycle.

Chapter 2 stated that the bulk of current research on expert systems innovation focused at a tool level, with an apparent disregard for organisational assessments and processes of change other than from a limiting but dominant technical perspective. Organisational and personal perspectives, and concepts derived from these, have the potential to contribute significantly towards expert systems assessment and development. This study has provided some indication of how conceptually and methodologically this may be achieved in ES development and assessment. However applying these new concepts, generates a new set of processes, values and priorities which in practice may be difficult to reconcile with the formal culture defined by the use of methodologies and structured tools and associated working practices which have evolved around these in an organisation. Therefore applying a multiple perspective approach requires careful management and respect of the current situation whilst suggesting ways in which it may be enhanced through the use of other perspectives besides a technical one.

An implication of applying multiple perspectives at a tool level was the understanding that no single methodology was sufficient in describing the total process of assessment and development. Rather than a methodology therefore, the notion of an eclectic development framework was defined in which tools were selected according to their ability to perform specific tasks, but also that they could be applied in a way which was appropriate to the development context. Thus, the method should be crafted by the organisational settings. By contrast, current expert system methodologies are significant innovations in themselves which prescribes a blueprint of organisational and technical behaviour which may conflict with the current company culture and practices.

To apply this framework required a logical, or functional, understanding of the necessary tasks of assessment and development (this was called an activity framework) and a knowledge of the range of possible tools and processes which may be applied to achieve this. From this, organisational characteristics and the development context at any stage in the innovation life-cycle defined not only which tools were appropriate, but also *how* they should be applied. Experiences in the client company show that the delivery of a tool or method is often more important that its technical ability to perform a particular function. Thus, for example, a formal information modelling tool may be used informally to determine business strategy, whilst in contrast, soft methods have proved to be useful in some settings to define the information characteristics of a problem situation. In both cases, the use of the tool was shaped by the organisational settings.

A critical stage in using the development framework, and of technology assessment more generally, is the need to acquire a detailed understanding of the organisation, its characteristics and problems. This is why it was argued that senior management roles were more suited to perform or oversee these functions that engineers and technologists for example. For the author, who was new to the organisation, this was especially important. IDEFo was chosen because, as an activity based model, it spanned different levels and viewpoints in the organisation and had the potential to link high level modelling approaches such as strategic mapping and conflict resolution with more detailed, low level approaches such as information and entity modelling. However, in order to satisfy each of these roles, IDEFo was used in various formal and informal ways. Where 'hard' benefits derived from the model itself as an end-product, more important 'soft' benefits emerged from the actual process of undertaking the modelling exercise. The latter provided significant insights and understanding of the company culture, problems, conflicts and needs. Therefore it provided a vehicle for providing a legitimate and rapid understanding of the organisation.

As an informal output though, the greatest value of the modelling exercise was to those that were directly involved. Indeed, the value of modelling to the company was limited but for the fact that an additional output from this exercise was a listing of organisational problems and conflicts, each of which, through the application of selected business planning tools (notably critical success factors, value-chain analysis, and business impact grids), were attributed a business and organisational significance to the company. Identifying and placing organisational problems in a context in this way is of value irrespective of whether expert systems are developed or not. As a front-end to expert systems problem selection however, it identified those problems identified which were appropriate as 'first project' applications: these had a low business risk but were of organisational value by providing a service or support role and therefore aimed at improving operational efficiency in some way. A number of studies show that targeting appropriate expert systems was one of the main difficulties faced by manufacturing companies: many chose targets that were either too ambitious or risky, whilst others chose organisational areas where other developments competed with resources and time.

As well as meeting organisational requirements, clearly it was important that problems were identified and selected that were appropriate and feasible for expert systems technology. The match between problem attributes and technology capabilities was assisted through the use of problem identification and feasibility check-lists. These, together with numerous practical constraints imposed upon selection, were successful in reducing a large number of possible alternatives down to a short-list of three candidate applications. More detailed assessment at this stage of selection concentrated upon gaining a greater appreciation of the application settings and context in which potential systems would be built. Multiple perspective concepts were again applied during the analysis and showed the problem focus where the balance of each of the perspectives met. In one of the projects, this revealed that there were likely to be potential organisational problems in development and that the main motivation for pursuing development were mainly political and instrumental. This was shown by a dominant personal perspective defining organisational and technical roles.

Beyond this stage of problem selection, it was very difficult to distinguish between the two remaining projects with the limited information available and therefore evaluation prototypes were built using a PC based expert system shell. As prototypes, it was important to recognise that their role was only to provide design information, about ease of knowledge representation and user-interface requirements for example, and that they would not be allowed to be developed beyond this role. This deterred a natural temptation to enlarge the knowledge base to an eventual full scale system in an unplanned and unstructured way. Two prototypes were built and provided a significant amount of development information, some of which was used in the planning and design of the eventual full-scale system. For selection purposes, it showed clearly the likely difficulties and limitations of one of the proposals and, conversely, showed the potential in developing the other, albeit with a number of modifications in design. Even at this level however, when scaling problems were not encountered, a significant amount of time was spent in developing the prototypes and providing working examples for the experts, users and management to review. This was compensated to some degree by the fact that building these systems provided a useful and effective means of learning about the technology and developing programming and knowledge acquisition skills without the pressures of a 'live project' operating to time and cost constraints.

The subsequent development of the help-desk had two distinct phases, mirroring a progression in personal learning and expectations of what was achievable from an expert system, and more particularly a rule-based shell. The first phase was characterised by three shortcomings: -

i) a failure to estimate the knowledge content in defining the size and scope of the problem domain. The initial scope of the help desk was to include Mainframe, network and personal computing hardware. However this was reduced by a third as it became apparent that performance targets could not be attained.

ii) a failure to judge the time and effort required to verify knowledge. Although the expert/developer cycle of knowledge elicitation, intermediate representation and verification was very successful, particularly in that it made use of existing company tools and techniques, such as IDEFo, to communicate with the experts; as a manual process, it was also laborious and slow. Computer aided knowledge acquisition tools were available and could possibly have improved productivity, but represented a level of commitment which was beyond the scope of development as a first project.

iii) a failure to recognise the complexity of interface design and an overestimation of the capabilities of the expert system tool. There were two levels of interface: the man-machine interface between the operator of the help desk and the help-desk itself, and the interaction between the operator and help-desk caller. In the revised system, the nature of both was greatly simplified, although the economic feasibility of the system was subsequently reduced.

Programming the expert system was straightforward at a code level, but was complex in terms of structuring the knowledge base. Furthermore, the structure of knowledge bases could only effectively be determined at an intermediate representation and design phase and not during programming itself since the developer was too often embedded within the code to appreciate the shape and form of the global system. This has important implications for maintenance since the person most suited to the task must have a clear understanding of the knowledge base structure as well as have programming skills. Therefore in the case of the help-desk, expecting the expert as an end-user to maintain the system, especially given the limited training he received, was clearly ambitious.

Following completion, the revised Office Systems Help-desk was implemented and remains in operation, although in a reduced capacity from that originally intended.

Furthermore, since the knowledge-base is not updated regularly, for the reason described above, its organisational value is diminishing.

Throughout the programme of assessment and development, the existing computer department support structure was used and this worked well since it provided an established development support and communications network which was well suited to expert systems. However, the computing department was less well suited as the location of the first application since, in organisational terms, it was seen as a partitioned function in the company. This might also explain why the impact of the help-desk and associated concepts progressed little beyond the computer department.

The issue of impacts, costs and benefits raised a final question in this study of how experts systems should be evaluated. One method used was to compare experiences in the client organisation with those of other manufacturing companies. However documented experiences from a user perspective is still unavailable and therefore the study was reliant upon a survey of 134 companies in the UK. The results show confusion over the organisational role for expert systems; clear disenchantment with the currently available development methodologies and approaches; and a difficulty for these companies to take development beyond an experimental /prototyping stage. Furthermore, there appear to be a clear set of pitfalls and barriers to implementation which derive not from any particular effects of the technology *per se*, but because of a failure to manage the innovation process more effectively. Where expert systems were being used operationally, there was a clear tendency for companies to use systems which were small, simple in scope and design and, in business terms, addressed System and Operational type business problems. This belies the expectations and claims by the supply industry whose defend the use of large scale integrated applications. It demonstrates moreover, a basic lack of feedback from user organisations back to the supply industry, and indeed much of the research community, with the effect that products and tools will continue to be developed which have no direct relevance to the needs of the manufacturing sector. It also suggests that with the current imperfections in the market structure and transfer process for expert systems innovation, then there is an additional responsibility upon the user organisation to attribute more importance to technology assessment and greater rigour in problem identification, application selection and requirements specification.

The problems and issues generated in the introduction and take-up of expert systems knowledge in an organisation call for a second model of evaluation which looks beyond the delivery of 'end-products', such as the IDEFo model or the Help-desk, to consider the effectiveness of the whole process of innovation and technology transfer as a life-cycle of change within the company and also in terms of its relationships with other organisations. By adopting a 'knowledge transfer' approach more particularly, the transfer focus shifts from what is feasible to what is required. A review of current transfer models though, show a strong bias towards the supplier of knowledge rather than the needs of the recipient. This is mirrored in the structure of the expert systems market where the focus of discussion is upon how Government, vendors and academia can make ES concepts, information and technology potential most appealing and marketable. Clearly such issues of knowledge presentation and availability are important, but they are driven independently of the development and process needs of user organisations. Problems of technology mis-use, a failure to develop the technology beyond a prototype stage and difficulties in implementation all derive from this point.

9.3. An Evaluation of Multiple Perspective Concepts (MPC)

The foundations of this study have been the adaptation and incorporation of what is a simple concept, multiple perspectives, into nearly all analyses. The notion of looking at any problem or situation from different perspectives and breaking the problem down into different settings is, despite its simplicity, a very powerful framework. In Chapter 2, MPC was applied as a means of distinguishing between different types of research literature on expert systems: from a very dominant technical perspective in the coverage of early expert systems, to the understanding more recently that organisational and personal perspectives shed more light on why ES innovation is not as successful as it could be. It has also identified a 'humanist' i.e. dominant personal perspective, which will be discussed later in terms of human centred design concepts, in which clear human goals are set, but there are uncertainties how they might be achieved in organisations through the use of technology for example. As a single dimension philosophy therefore, it too has problems.

An analysis of perspectives has also been useful in rationalising approaches towards innovation. A primary purpose of the study within the Client Organisation (CO) was to define an approach for the introduction of expert systems in the CO. As a 'leading-edge' technology in the market place as well as being new to the CO, a major T (Technical)-perspective effort was required foremost in order to explore the technology's capabilities and potential. Since it was a new technology, the organisational and personal perspectives were highly underdeveloped in comparison to existing technologies such as Computer Aided Design where individuals and organisations have taken positions on the issues arising from the technology in operational use. Therefore in order to compensate for this strong technical drive, it is necessary that either the organisational or individual perspectives provide the basis for technology assessment, or what Linstone calls the 'organising principle'. In the study, assessment was driven by a primary organisational perspective by modelling the company and defining a hierarchy of problems which were of particular business and organisational significance. Placing a human perspective foremost was practically difficult, given the present business culture, and therefore it served best as an ancillary viewpoint to the organisational and technical perspectives. Sometimes though, it was very difficult to distinguish between perspectives: during interviewing as part of the IDEFo study for example, personal needs and viewpoints of top management were expressed in organisational terms; Linstone called this a 'situational-personal' view because these individuals perspective is closely related to the formal 'company view'. By contrast, at lower management levels individual perspectives became less aggregated, informal and more personalised.

The use of MPC provided a basis for determining the application context in the Development Framework described in Chapter 4; it was also used in the design of 'balanced' check-lists using during application selection; and was adopted implicitly throughout development and evaluation - more formally in the latter through its suggested incorporation with the model of knowledge transfer outlined in Chapter 8. These many applications suggest that MPC can be no more than a conceptual framework. It cannot be 'intellectualised' or produced into a theory: it is simply a vehicle and communications device which enlarges the capacity of people to see any situation from different settings and to recognise different points of view.

Earlier chapters have shown that by applying MPC, new tools and concepts become available; in practice this meant that the author was faced with multiple learning-curves, making it necessary to understand and model the organisation, the technology, the application and specific development context, and those individuals immediately involved in the programme. However it is seldom in an organisation, that a single individual will be expected to perform all these tasks, but rather each will be delegated to other individuals or functions. In this case, MPC is useful to the facilitator of this change process in retaining a balanced approach.

As an enquiry system, MPC has no defined use or pre-conceived assumptions about settings or perspectives , other than to fully comprehend the problems or situation from all perspectives and settings. Therefore it does not promote a 'partial view' which in turn may result in a partial solutions. As Linstone comments,

> Advocates of a technology, whose career and fortune are tied to its development are likely to be critical of other participants, particularly those who may be playing a decision-making or judgemental role' (1981; p301.)

This generates two potential conflicts in the use of MPC. The first is that the potential user of the framework may fail to see the importance of other perspectives, or naturally attribute greater importance to some settings than others. Secondly, asking an experienced engineer or analyst to think in organisational, business or human terms may cause political problems and resentment. Conversely, people in such roles who are aware of the need to adopt other perspectives may feel remote or helpless because the institutional mechanisms were not available to achieve this. Both problems raise the question 'who in an organisation should use MPC?'. In the client company some organisational roles demand a balanced viewpoint, such as a functional head, manager, or policy and group decision maker, and are more likely to understand and empathise with MPC and apply it more effectively: whilst others are not required to operate in other than a single dimension, the 'computer programmer' for example. In other roles, MPC provides a useful paradigm, in design for example, but the individual may be under cost, budget, time, and other organisational constraints which prevent him or her from using it effectively. Therefore although in the author's case, MPC was used tacitly in all analyses, it is likely that in most other situations in a manufacturing context, its use will be restricted to a policy level and in high-level decision-making.

9.4. An Evaluation of the Development Framework

The basis of the development framework was as an alternative to the 'all encompassing' methodologies which often fail because they become either too large and cumbersome to manage by most organisations, or are 'all-encompassing' but only in only one dimension. In contrast, the development framework was committed to providing an evaluative structure by which a company could determine the most appropriate tools, disciplines and processes for the organisation. It defined, as a minimum specification, those tasks which were essential to development through the 'activity framework' and assumed that the technical tools were available which could accomplish these tasks. The final choice of approach was then based upon the organisational settings, identified through IDEFo in this case, and the development context to which the tools are applied; here MPC was used for this purpose.

A problem in using the development framework, is that although it shows sensitivity to how a tool should be delivered, whether a tool should be applied formally or informally for instance, in practice the author often had no control over these events to change them accordingly, since there were organisational constraints in which the framework operated. An implication from this is that the use of the framework, like MPC, is likely to prove more effective at a strategic planning level in determining a hierarchy of tools and methods of assessment and development for a chosen organisational group, rather than being considered as an operational tool. A strategic view is necessary in using the development framework because the choice to adopt a particular tool or method may be motivated by local difficulties and may therefore fail to support other stages of development. This does not mean however that the person or group charged with applying the framework, is not sympathetic to this level of needs, this can be ensured by defining appropriate approach mechanisms at this level, but that these should be placed within a wider organisational setting so that there is continuity between different levels of tools.

9.5. An Evaluation of IDEFo Organisational Modelling

A principal conclusion from this study is that technology is crafted by a unique blend of internal (personal, political, cultural) and external (market, business, customer, supplier) factors which define the 'make-up' of an organisation. By definition, in order to be able to assess the potential contributions of a technology, such as expert systems, and develop it in an appropriate way, then the assessor and those managing the development process must understand these factors in order to define how the technology will be applied and adapt to these settings. This requires a detailed and comprehensive understanding of the organisation itself: not just at a explicit level of technology layout, organisational structure and business strategy for example, but also from an tacit level in terms of company culture, ethics, values and politics. This requires a strategic view of the organisation which is able to rationalise the organisational 'mess' whilst retaining a deep level knowledge of organisational operations. To acquire both requires time and correct positioning in the organisation. Since the author had access to neither than it was necessary to undergo an accelerated organisational learning process which was achieved through modelling the company using IDEFo. The model itself was secondary to the processes of investigation that were involved.

Since an objective in applying IDEFo was to span all organisational viewpoints and levels, it was inevitably a large undertaking demanding company resources and time. Its value therefore, should not be associated with single ES developments but be considered as being of organisational value to the company whether systems are developed or not. However since this potential value was limited to those that actually undertook the study, other than through the production of formal deliverables such as the IDEFo model and an 'issues and conflicts' report, then the full benefits could not be transferred directly to the organisation.

With respect to the expert systems programme in the client company though, the IDEFo modelling exercise was of significant value in both formal and informal terms. It provided the basis for a top-down business driven approach to problem identification and selection and was used often during the design and development of the help-desk as a communications device and means of bounding expert's knowledge. It is uncertain whether other organisations would gain the same benefits from IDEFo which is why its use has been stressed as a vehicle for secondary analyses rather than because of specific virtues of the tool itself. Certainly other organisational modelling techniques may better suit other organisations: the important point is that some form of modelling necessarily precedes technology assessment and, in the case of expert systems, problem selection. Whichever tool or process is chosen, this study discovered that to model the organisation effectively, it must satisfy four basic criteria:-

i) It should span organisational levels.

ii) It should be flexible enough to incorporate different organisational viewpoints.

iii) It should provide a common communications language and forum for discussion.

iv) It should not focus upon particular end-states (conflict resolution, business analysis etc), but promote a process of awareness and understanding of situations and needs.

9.6. An Evaluation of a 'Check-list Approach'

The design of the problem identification check-list is not unique, but rather a composite of currently available check-lists which address this particular phase of problem identification. It is different though in the way that it is enhanced by organisational modelling and business techniques to provide a more integrated approach than one based on feasibility alone.

The first application of a check-list in the study was for problem identification whereby problem attributes were identified by functional managers where expert systems technology may be appropriate. This was very much a 'technology fitting' exercise defining where expert systems were potentially feasible rather than necessary or useful. Therefore it was important to relate this bottom-up approach to the top-down business information provided in part by the IDEFo study in order to provide a context for the use of the check-list. In this capacity, IDEFo was very useful for two reasons: -

i) It helped to identify causal relationships between identified problems.

ii) It showed the current functional structure, technical layout and business significance of activities and technologies, all of which, as Earl identified, are contingent factors in shaping how the computer department should be structured, for problem selection in this case (1989).

The process in which the check-list was used as part of a wider programme of technology assessment also differed significantly from other commentaries which describe its use as a 'self-help' development guide. In the client organisation by contrast, problem identification was distinguished from application selection, the latter of which was managed and undertaken by the computer department.

A second check-list was used during the second phase, application selection, in order to assess the technical feasibility of candidate applications identified from the first check-list. This check-list has no claim to being complete, nor did it adopt some of the more rigourous methods of feasibility assessment such as score rating. Nevertheless, the process of systematically evaluating applications using a common set of criteria provided a useful basis for describing and comparing applications.

Where the first two check-lists defined application potential and technical feasibility respectively, a third was used to assess the total suitability of a proposal for development. This check-list brought together a significant amount of design information stemming from the design of evaluation prototypes as well as other assessments. The design of the development suitability check-list itself was motivated by each of the settings and viewpoints of MPC and therefore was potentially a multidimensional study. However, since it was unlikely that the check-list would be used by anyone outside the IT function, the study would be carried out with a strong technical bias. Therefore it was necessary to explicitly state human and organisational assessments and provide guide-lines to technical staff as to how these could be carried out. Although this could hardly be described as 'multiple perspective', it was an attempt to broaden the scope of assessment beyond the customary single technical dimension. Clearly, the more 'alive' the user of the check-list is to multiple perspective concepts, the more likely the check-list will be used in this fashion.

9.7. An Evaluation of the Development and Operations of the Help-desk

The approach taken in the development of the help-desk may appear long and protracted, particularly since the 'deliverable' affected only a small proportion of the organisation in the end. However this approach taken went beyond the investigation of single projects and instead concentrated upon defining an infrastructure for a programme of evaluation and developments. In terms of payback though, the help-desk was not viable because the opportunity costs savings of displacing the expert with a cheaper resource were diluted between helpdesks in the revised system and moreover, a number of costs from the phase 1 development were carried over and classed as 'sunk costs'. The most significant cost component of the help-desk was the expense of interviewing the expert; this tied up both the expert and the developer for a significant period. The economic viability of an expert system is therefore highly sensitive to correct estimates of the duration of the knowledge acquisition phase. Yet, it remains the least well understood and least well documented aspect of development. The quality of estimates depends upon the ability to gauge the knowledge content of a problem domain and establish the appropriate representation formalism that will ensure rapid encoding and effective verification. Experience with the help-desk shows that this is a craft rather than a science and there appears to be no substitute for intuitive estimates made by the 'experienced developer'.

Operational experiences in using the help-desk show that its value was twofold: as training aid to assist trainee office systems engineers; and as a decision-support adviser to the expert. It is significant that neither roles were considered during the initial phase 1 design specification. Advocates of the 'RUDE' model of expert systems development, described in Chapter 4., would perhaps claim that this endorses their view that the outcome of ES development cannot be planned or specified in advance, but rather is an iterative design process based around prototyping. However, these experiences merely places a closer boundary over what can be specified and planned and what must emerge from the process of prototyping. Certainly though, the original notion of a centralised help desk service attending to many different types of end-users' requests was clearly ambitious. Expertise is context sensitive; it is also judged in delivery terms which, in the case of the help-desk, meant that really only the expert was suited to use the help-desk.

From this lesson, a further conclusion may be drawn, in that the organisational skills required to apply the system were as important as the ability to understand and use the help-desk itself. The distinction lies in the types of knowledge generated at both interfaces. At the man-machine interface, Chapter 7 referred to the presentation of 'digital' knowledge; this lacked a context and required interpretation and application by the help-desk operator so that it could be 'translated' into an analogue form which would be understood at a second organisational interface. These skills could not be replicated within the tool itself, as originally intended, but required more training and greater knowledge and communication skills by the help-desk operator instead.

Problems emerged in the development of the help-desk which would probably not have arisen for the 'seasoned' knowledge engineer; if it had been the second or third project rather than the first, thus allowing the development framework to be finely tuned to the organisation; and, possibly, if it had been built in another organisational setting. Despite the technical, organisational and human factors which make the help desk unique, there are a number of general lessons to be learnt from this experience. However, it is limiting to restrict discussions on the effectiveness of the assessment and development programme to the value of the help-desk alone. The help-desk has proved only a part of a larger process of knowledge transfer process which has equipped the organisation with the skills to assess future expert systems ideas, classify their business impact and organisational value, and begin to assess its development suitability and specify development requirements from which cost estimates and formal justifications may be made. These are clearly pre-development skills, and the programme has been less successful in transferring development skills, such as programming, knowledge acquisition, verification and so on - these are very much personal skills which the author was unfamiliar with at the beginning of the study. However, it is at the level of development rather than assessment where outside consultancy and vendor assistance is most easily sought and also the most valuable.

9.8. The Direction of Further Research

There are a number of issues which have been addressed in this study which point towards the need, both qualitatively and substantively, for a new research agenda. Qualitatively in the sense that those in research should shift perspectives from essentially technical settings focusing upon expert systems potential and 'accessibility' issues, to consider and redefine needs from a user organisation's perspective. This new perspective provides a new set of priorities and values which stress all aspects of innovation, but particularly technology assessment and knowledge transfer and thus depart from the current 'technology fitting' structure to one which identifies the potential for a certain type of processing to handle particular types of business and organisational problems. At a substantive level, this section proposes three principal areas for further research work:-

9.8.1. Understanding the Knowledge Transfer Process

The knowledge transfer model is a product of this study and is based upon a certain set of experiences, although the value of speaking in terms of phases of innovation (accessibility, mobility, receptivity), levels of transfer , and the characteristics, processes and mechanisms between phases and levels, can be seen clearly within this context. Future research work may wish to apply the model to new situations, possibly with new technologies, and therefore validate its use beyond the specific circumstances in which it was developed. The model may also be used in different ways: one example is to use the model to promote a receptivity focus during ES innovation. Thus, the process would begin from the receptivity phase and work backwards to define the necessary processes and mechanisms of mobility and accessibility which achieve this state. In the client organisation, this was actuated by modelling organisational problems, needs and conflicts. In other organisations, this may be unacceptable and therefore other mechanisms will be required. The necessity is, at this level, and more generally at an inter organisational level, that accessibility processes and mechanisms (such as government sponsored programmes, academic research, consultancy, and the vendor/supply market) are crafted and shaped by receptivity factors. However to achieve this requires firstly, more research which provides a richer picture of how expert systems are actually being used presently in manufacturing; what the problems and barriers are; and what management understand expert systems to be and how they go about justifying the use of the technology. The user survey in this study provided some indication, but more detailed, more frequent and more widespread investigations are necessary in order to gain a true representation of their use. Yet as a recent publication identified, such 'hard information' is simply not forthcoming:-

> 'The expert systems field is currently characterised by a wide range of theoretical viewpoints, but an absence of generally accepted theory in many important areas. Opinions on the commercial value of expert systems vary from the extremely negative "expert systems have failed", through to the dismissively positive "expert systems are now routine in commercial

> applications". There is a shortage of hard information, compounded by the widely varying definitions that seem to be applied' (Bramer: 1990; p2.)

As long as the present situation remains poorly understood, it is difficult to imagine how a (user-driven) agenda for change can be devised and implemented effectively. What is perhaps the most discouraging point of all, is that without such hard information which show graphically that current failures in ES innovation are because of receptivity problems such as poor management, wrong choice of business or organisational problem, lack of interest, and the wrong methods and processes of change, commentators will continue to define failure in technical terms alone-inefficient programming, poor interfacing, "knowledge acquisition bottlenecks" - which though clearly important, are themselves only derivatives or symptoms of these wider and more important issues.

9.8.2. Costs and Benefits

A great uncertainty in ES innovation is precisely how manufacturing companies should go about defining the return on investment of expert systems. Presently, reflecting the state of maturation of the technology as much as anything else, many expert systems projects require no justification process because costs are covered by 'research and development' budgets and other experimentation contingencies. This has tended to defer this issue such that it has been absent from the mainstream discussions on expert systems.

The simplification that expert systems 'add-value', however true, is insufficient as an explanation of its business and organisational effects. In a commercial manufacturing environment, these benefits will need to be defined precisely. The business tools described in this study, alongside a problem mapping exercise like IDEFo, go some way to defining and classifying the strategic business and organisational value of candidate expert system applications. Further research would be deserving to see if each business class of ES require different cost-benefit approaches. For example, the Help-desk, as a 'System' type application was justified using discounted payback to measure the benefits of reducing costs through improving functional efficiency. However, it is likely that such an approach would be limiting in defining the benefits stemming from 'Tactical' or Strategic' applications. Tools at this level are likely to be relative rather than actual, through using Critical Success Factors or Value-Chain analysis for example, although Primrose and Leonard, amongst others, have argued that these benefits too can be defined in cost terms (1988).

A second and related uncertainty is in quantifying the costs of ES development. The economic viability has been shown in this study to be sensitive to time and resource estimates, and particularly on the assumptions made about maintenance, and the effectiveness with which the 'knowledge content' and subsequent acquisition lead-times may be determined. A response by the research community to these uncertainties has been to develop development tools and approaches which automate and systemise a number of processes and therefore improve predictability, and thud the repeatability of estimates. Although this is a valid approach in principle, in practice these tools are large and expensive and transgress the current level of commitment shown in manufacturing. Therefore more viable and inexpensive approaches are required in the form of experiental guide-lines or 'best-practice' procedures for example. This again however, requires a more co-operative culture between industries so that experiences and feedback may be channelled to provide a central body of knowledge in this area.

There are three models of estimating which emerged from help-desk development experiences and a third which is proposed in future. The first and most obvious is to make use of expert opinion. A second approach is to make estimates on the basis of analogy. This is a more formal approach to the first option, and makes direct comparisons to past projects: thus the effort for a similar previous project is taken as the initial estimate for the new ES project. The initial estimate is then adjusted depending on the difference between the two projects. If the previous project was only a prototype, then it is critical that 'scaling factors' are accounted for in the estimate. A failure do so in the case of the help-desk resulted in a 40-50% error factor.

A third method of estimating is the software engineering model of decomposition. This involves breaking up a project into its smallest components or lowest level tasks. Estimates are made of the effort required to produce the smallest component or perform the lowest level task. The project estimates are then made by summing the component estimates. However this approach raises the question of how the component and task estimates are made ? In normal project management, they are usually based upon 'standards' or average values of effort which are then adjusted on the basis of the complexity of particular components or the difficulty of specific tasks.

In all likelihood, a mixture of all these approaches will be used: however common to all is that previous experiences and knowledge of ES development is required in order to define an 'estimating disciplines'. These methods are of limited use therefore to the first time developer. A fourth approach which may resolve this problem, and certainly merits further research, is 'group estimating' based upon group techniques such as the wideband DELPHI method for example (see Boehm: 1981). The principle behind such approaches is that the person most familiar with a particular estimating activity should be involved in the estimating process itself. Thus for example, the domain expert is most likely to be able to estimate the size and scope of a knowledge base, the end-user to define interface functions, the system developer is most familiar with programming effort and so on. Although this requires that the domain expert is aware of knowledge engineering techniques for example, it does ensure that those closest to the problem actually participate in the estimating and specification process.

A further reason why estimating costs is so difficult may be because inappropriate costing models and assumptions have been adopted. For instance, a current issue of debate is whether the costs of maintenance should be construed as capital or operating cost ? As a capital cost, expenditure is abrogated over the lifetime of the system. However, it was clear from the help desk experiences that maintenance costs were not constant and, in fact, could conceivably be exponential, with initial maintenance costs being low in the first few months, whilst after a year major structural changes to the knowledge-base, demanding significant re-work, were required.

It may even be more useful to consider maintenance costs in financial terms by considering company knowledge held in the ES as an asset. This allows tools such as depreciation to be used to determine at what point knowledge loses its value. A particular tool, Average Annual Equivalent Value, was mentioned in passing in this study, which could be applied to reveal the optimal time for replacement of knowledge, and therefore by implication, the frequency and scope of maintenance. Further research in this direction may help to establish cost functions for maintenance, and more generally, models for estimating development costs.

9.8.3. Human-Centred Technology Assessment ?

The use of MPC in this study has shown that the pervading culture of technology assessment, development and evaluation has been strongly technological. Moreover, calls have been made to incorporate mechanisms and processes which promote an organisational perspective, and to a lesser degree a personal perspective also. It is perhaps fitting to end this study by returning back to the qualitative level on the research agenda and forecasting the future state, not of expert systems, since it is likely that like all technical innovations they will be superseded by something better or developed to a state where they are not recognisable from their original form, but from the point of view of how technology assessment and technology development as structures of change will take place in future years.

The worst case scenario is that a technology focus may become even stronger and that this spills over to reshape cultural and societal values and define a 'technocratic society'. The author prefers to adopt a more sanguine picture in which technology assessment and development itself is fashioned by human and socio-cultural factors. Current research into 'human centred' systems and anthropocentric design adopt the second paradigm and argue that people have unique skills, such as imagination, creativity, insight and knowledge, which cannot be replicated , nor is it socially or culturally desirable to do so. The implication from this, is that technology assessment should follow a path in which technology is judged on its merits to enhance human worth by strengthening and supporting these creative skills rather than endeavouring to replace them. A second implication also is that there are more mundane and repetitive human tasks which may benefit from technological advancements. These ideas are explored by this author elsewhere (in general terms see Holden:1990a; and relating specifically to expert systems in manufacturing see Holden: 1990b). There is however, an overiding problem in adopting a dominant human/personal perspective in this way which stems from the difficulties in reconciling 'human worth' within an economic society defined in financial and business terms. Therefore the processes and mechanisms to which the Human Centred School of Thought subscribe appear naive and sentimental in the reality of commercial practice. However in the same way that cultural changes over the past decade have made 'green' and environmental concerns a major political and economic issue, a return to the 'quality' of human life may redefine the way in which technology assessment, development and evaluation is undertaken in the next decade. MPC shows, above all things, that different levels and viewpoints of debate, from the abstract and philosophical down to the information level, cannot be separated in their causes and effects.

BIBLIOGRAPHY

Ackroff et al. (1990) 'SARTS AUTOTEST-2' (Ackroff, J., Surko, P. Vesonder, G. & Wright, J.) in Bramer, M. (ed.) Practical Experience in Building Expert Systems, pp209-226., John Wiley and Sons Limited UK.

Ackermann, David, Angehrn, A., Luthi H., Arnoldi, M. (1989) 'A Context-orientated approach for decision support' in Tauber, M.J. & Grony, P Visualisation in Human Computer Interaction. Berlin: Springer lecture Notes.

ACME (1990) Applications of computers to manufacturing engineering, Directorate of the Science & Engineering Research Council, Research Projects Status Report July 1988- June 1990. SERC. Swindon UK.

d'Agapeyeff, A. (1984) 'Making a start: a review from British Industry' in Expert Systems, pp3-13., Pergamon State of the Art Report 12:7, Pergamon Press, UK.

d'Agapeyeff, A. & Hawkins, C.J.B. (1987) Report to the Alvey Directorate on the second short survey of expert systems in UK business; IEE on behalf of the Alvey Directorate, August 1987.

d'Agapeyeff A. & Hawkins C.J.B. (1988) 'Expert Systems in UK business: a critical assessment ' in the Knowledge Engineering Review pp185-201. Vol. 2(1).

d'Agapeyeff A. & Hughes, K. (1989) 'Expert systems applications in business: a management view ' pp308-318. in in Research & Development in Expert Systems V. (ed. Kelly, B. & Rector, A.L.). Proceedings of Expert Systems 1988, 8th. Conf. of British Computer Society's Specialist Group on Expert Systems, Brighton, 12-15 December, C.U.P. UK.

Alty, J.L. & Coombs, M.J. 'Expert Systems: Concepts and Examples ' National Computing Centre, Manchester, 1984.

Andrews, B. (1988) Successful Expert Systems , Financial Times Management Report, Financial Times Business Information 1988 UK.

Anjewierden, A. (1987) 'The KADS system', in proceedings of 1st. European Workshop on Knowledge Acquisition for knowledge based systems, Institute of Electrical Engineers, UK.

Atwood, M.E., Brooks, R. & Radlinski, E.R. (1986). Causal Models: the next generation of expert systems in Electrical Communications, vol. 60(2) pp180-184, USA.

Avison, D.E.& Fitzgerald, G (1988) 'Information systems development: current themes and future directions' in Information and Software Technology, vol 30(8), October 1988, pp458-466. Butterworths.

Avison, D.E., Fitzgerald, G. & Wood-Harper. A.T. (1988) 'Information systems development: a toolkit is not enough' The Computer Journal 31(4), pp379-380

Aylett, R. (1990) The use of knowledge acquisition tools' in AIAI Newsletter, No. 10. June 1990, pp1-7. UK.

Bader et al (1988) Bader, J. Edwards, J. Harris-Jones, C. Hannaford, D. 'Practical engineering of knowledge-based systems' in Information & Software Technology vol 30(5), June 1988 Butterworths, p266.

Bader, J. & Weaver, J (1988) 'Knowledge based systems: user benefits and making the business case' presented at Software Tools 1988, On-line Publications, Pinner UK.

Badiru, A.B. (1988) 'Successful Initiation of expert system projects' in IEEE transactions on Engineering Management Vol 35(3), August 1988, pp186-190. USA.

Beer, S. (1985) Diagnosing the system for organisations, John Wiley & Sons UK, 1985.

Beerel, A (1987) Expert systems: strategic implications and applications, Ellis Horwood Limited, 1987.

Benyon, D. & Skidmore, S. (1987) 'Towards a toolkit for the systems analyst' The Computer Journal, 30(1), 1987 pp2-7.

Berkin, J (1986) 'Expert systems: organisational, business and strategic implications' p151, presented at KBS 1986, July, London, Online publications, UK.

BIS (1989) BIS Applied Systems Methodology, Report Reference c89/3, Parts 1-8, BIS Limited, 1989, London.

Black, I., (1987) Replacement theory and annual average equivalent values Research Notes by Ian Black, Centre for Transport Studies, Cranfield Institute of Technology.

Boden, M.A. (1989) 'Artificial Intelligence ' in Intelligent Systems in a Human Context (ed. Linda A. Murray & J.T.E. Richardson), p11., Open University Press, UK.

Boehm, B.W. (1981) Software engineering economics, Englewood Cliffs NJ. USA, pp36.

Born, G (1988) 'Guide-lines for the quality assurance of expert systems' CSA working paper on quality assurance and expert systems; working paper issues 1.1, 23 November 1988, Computer Services Association UK.

Bradley, J. & Harbison-Briggs, K. (1989) 'The symptom-component approach to knowledge acquisition' in SIGART Newsletter, April 1989, No. 108, pp70.

Bradbury, F. et al. (1978) Transfer Processes in Technical Change (with Jervis, P., Johnston, R. & Pierceson, A.) Sijthoff and Noordhoff (pubs.)

Bramer, M. (1984) '___' in New Information Technology, edited by A. Burns, Ellis Horwood Limited, 1984, pp148.

Bramer, M. (1988) 'Expert systems in business: a British perspective' in Expert Systems Journal, vol. 5(2), pp104-117.

Bramer, M. (1990) Practical experiences in building expert systems; editor, John Wiley & Sons UK.

Braunwalder, K. & Stefek, Z. (1990) 'RBEST: An expert system for disk failure diagnosis during manufacturing ' in Bramer M. (ed.) Practical experiences in building expert systems, pp125-146. John Wiley & Sons UK.

Bright C., Inman A. & Stammers, R (1989): "Human factors in expert systems design: can lessons in the promotion of methods be learned from commercial DP?" in Interacting with Computers, 1(2), August 1989, Butterworths.

Bobrow, D.G, Mittal, S. & Stefik, M.J. (1988) 'Expert systems: perils and promise' in Pham, D.Y. (ed.) Expert Systems in Engineering, IFS Publication, Springer-Verlag (UK).pp.19-42.

Bobrow, D.G & Stefik, M. (1985) Knowledge based programming, Alvey news No. 13 p13-14 UK.

Brodner, P (1986): "Skill based manufacturing vs. 'unmanned Factory'- which is superior ?" in Int. Jrnl. of Industrial Ergonomics 1(86) 145-153, Elsevier Pub.

Brodner, P (1988) ' Options for CIM "unmanned factory" versus skill-based manufacturing' Computer Integrated Manufacturing Systems vol. 1(2), pp.67-74.

Bryant, N. (1988) Managing Expert Systems, John Wiley & Sons 1988, UK.

Buchanan, B. & Feigenbaum, E. (1978) 'Dendral and mega dendral: their application and dimensions ' in Artificial Intelligence, vol. 11, pp5-24. USA.

Buchant, J & McDermott, J (1984) 'R1 revisited: four years in the trenches' The A.I. Magazine, Vol. 5(3), Fall 1984, pp21-32.

Candy L. & Lunn, S. (1988) 'Design Strategies for expert systems: a case study' in Berry, D. & Hart, A. (eds) Proc. Human & Organisational Issues of Expert Systems (Joint ICL and Ergonomics Society Conference) Stratford-Upon-Avon, UK.

CAPM (1988) Report of the computer aided production management Workshop and tutorial; ed. J.G. Waterlow & F.J. Clouder Richards, SERC, 6/7th September, 1988.

Carver, M. (1988) 'Practical experiences of specifying the human-computer interface using JSD ' in Contemporary Ergonomics (ed. Megaw, E.), Taylor & Francis, London.

Chamberlin, G.F. & Butler, A.R. (1988) 'The Aries Club -experiences of expert systems in insurance and investment ' in D.S. Moralee (ed.) Research and Development in Expert Systems, proceedings of the fourth Conference of the British Computer Society Specialist Group on Expert Systems, Cambridge University Press, UK.

Chandrasekaran, B (1988): 'Generic tasks as building blocks for knowledge based systems: the diagnosis and routine design examples', Knowledge Engineering Rev. 1988.

Chandrasekaran. B, & Mittal R. (1983) 'Deep versus compiled knowledge approaches to diagnostic problem solving' International Journal of Man-machine Studies 19, pp425-436.

Checkland, P B. (1981): Systems Thinking, Systems Practice. John Wiley & Sons UK.

Checkland, P.B.(1989) 'Soft systems methodology' in Human Systems Management, vol 8(89), pp273-289.

Citrenbaum, R.L. & Geissman, J.R (1986) 'A practical cost-conscious system development methodology': Proceedings of the 2nd. Annual A.I. and Advanced Computer Technology Conference, Calif. USA. pp.56-61.

Clancey, W.J. (1985) 'Heuristic Classification' in Artificial Intelligence vol 27 (1985) pp289-350. UK.

Clark, R. (1988) 'Legal aspects of knowledge-based technology' Journal information technology, vol. 3(1), March 1988.

Clegg, C (1988): 'Appropriate technology for humans and organisations' J. Inf. Technol.3 3, 133-145.

Clegg, S. & Dunkerley, D. (1980): Organisation, class and control, Routledge & Kegan Paul, New York USA.

Compton, P. et al (1989) (Horn, K. & Quinlan, I.., Lazarns L. & Ho, K.) 'Maintaining an expert system'pp366-384 in Quinlan, I. & Ross, J. (1989) 'Application of Expert Systems: Volume II ' from 3rd and 4th Australian Conferences, Addison Wesley, 1989 1st ed.

Cooley, M. (1988): ' Creativity, skill and human-centred systems' in Goranzon, B. & Josefson, I.(eds) Knowledge, Skill and Artificial Intelligence, Springer-Verlag, Berlin, pp.127-137.

Cooley, M (1989) European Competitiveness in the 21st Century: integration of work, culture and technology. European Expert Group, F.A.S.T. report June 1989.

Cooley, M (1990) 'The new shape of industrial culture and technological change ' paper presented at International Workshop on Industrial Culture and Human Centred Systems, Tokyo Keizai University, Tokyo, May 14-18 1990.(published in Japanese)

Cordey-Hayes, M. (1990) Models of Technology Transfer Personal correspondence and discussion with Professor Cordey-Hayes, Innovation & Technology Assessment Unit, Cranfield Institute of Technology, Cranfield, Beds, MK43 0AL.

Crispin, E. & Montgomery, T.A. (1989) 'GEMINI: Government expert systems methodology initiatives'; Proc of 5th. International Expert Systems Conference, 6-8th July, London, Learned Information UK.

Curnow, R. (1987) 'the organisational and structural impact of KBS'; presented at KBS '87, online Publications, Pinner UK. pp1- 7.

Damodaran, L. & Eason, K.D. (1981) 'Design procedures for user involvement and user support' in Coombs M.J.& Alty J.L. p373-388.

Dawson, P. (1988) Intelligent knowledge-based systems (IBKS): organisational implications in New Technology, Work & Employment p56-65, Vol 3(1), Spring 1988, Basil Blackwell UK.

Davis, R. (1986) 'Knowledge Based Systems' in Science vol 231, No. 4741, pp957. USA.

Demarco, T. (1980) Structured analysis and system specification, Yourdon Press, USA, 1980.

Diaper, D. (1988): 'The promise of POMESS:(A people orientated methodology for expert system specification)' in Berry, D. & Hart, A. (eds) Proc Human & Organisational issues of expert systems (Joint ICL and Ergonomics Society Conference May 4-6th.) Stratford-Upon-Avon, UK.

Dreyfus, H.L. (1972) What computers can't do: the limits of artificial intelligence', Harper & Row, New York.

Dreyfus, H. & Dreyfus, S. (1986) 'Why expert systems do not exhibit expertise ' in IEEE Expert, pp86 - 90, Summer 1986. USA.

Drummond *et al* (1988) 'A framework for technology transfer with the A.I. Applications Institute' (by Drummond, M. Macintosh, A., Tate, A. Barlow, D. & Greenwood, M.) in Knowledge Engineering Review Journal, pp.159-168, vol.2 (2) UK.

Duda et al (1978) Development of the Prospector consultation system for mineral exploration, (Duda R.O., Gashing, J.C., Hart, P.E., Konolige, K., Reboh, R., Barret, P and Slocum, J.) SRI Publication, Menlo Park, Calif. 1978.

Durham, T. (1987) 'Moving experts forward one small step at a time', Computing, vol. 17., September 1987.

Earl, M.J. (1989) Management Strategies for Information Technology, Prentice-Hall Int (UK) Limited.

Eason K.D., Harber, S.D.P., Raven, P.F., Brailsford, J.R. & Cross, A.D. (1987) 'A user-centred approach to the design of a knowledge based system' in Bullinger & B. Schackel (eds.), proc. Human-Computer Interaction, Interact 1987, North Holland Elsevier

Emberton, J. & Mann, R. (1988) 'A methodology for effective information systems planning' in Information and Software Technology vol 30(4), May 1988. Butterworths UK.

Edmonds, E. A. (1987) ‘Good software design. What does it mean ?’ in Bullinger & B. Schackel (eds.), proc. Human-Computer Interaction, Interact 1987, North Holland Elsevier

Ernst, G.W. & Newell, A. (1969) GPS: A case study in generality and problem solving, Academic Press, New York, 1969.USA.

Esprit (1987) ‘The KADS methodology: analysis and design for knowledge based systems ’ Report Y1: P1098 Synthesis Report, Esprit ed. S. Hayward.

Feigenbaum, E.A., McCorduck, P & Nii, H.P. (1988) The rise of the expert company Macmillan Press, New York., USA.

Friedrich J. (1990) ‘Industrial Culture Influences on the Development of Information Technology’ paper presented at International Workshop on Industrial Culture and Human Centred Systems, Tokyo Keizai University, Tokyo, May 14-18 1990.(published in Japanese).

Frost & Sullivan (1990) The west European market for expert systems’ Report No. E1266ID, Winter 1990, Frost & Sullivan USA.

Fry, M. (1989) The Helpdesk Protocol International Publishing UK.

Gabriel,C. (1989) ‘Clearing the Hurdles ’ in Informatics Journal, February 1989, p65. UK.

Galliers, R (1986) ‘Information Systems and technology planning within a competitive strategy framework’ in P. Griffiths Information Management: State of the Art Report, Pergamon, Oxford 1986.

Gane, C. & Sarson, T. (1984) Structured systems analysis: tools and techniques, Prentice-Hall Inc, Englewood Cliffs, NJ, USA, 1984.

Gevarter, W.B. ‘The nature and evaluation of commercial expert system building tools’ IEEE Computer (May 1987) pp24-41. UK.

Gill K S (1986) ‘Knowledge based machine : issues of knowledge transfer’ in Gill, K.S. (ed.) AI & Society, John Wiley, Chichester, UK.

Ginsberg, M.L. (1987) Readings in non-monotonic reasoning, Morgan Kaufmann Publishing, USA.

Godwin, A.N., Gleeson, J. & Gwillian, D., (1989) ‘An assessment of IDEFo notations as descriptive tools’ in Information Systems, vol 14(1), pp29-45, 1989.

Goranzon, B. (1988) ‘The practice of the use of computers. A paradoxical encounter between different traditions of knowledge’ in Goranzon, B. & Josefson, I (eds) Knowledge,skills and artificial intelligence Springer-Verlag, Berlin FRG pg. 9-18.

Green ,D. Colbert, M. & Long, J. (1987) ‘Towards a Transaction approach to KBS development’ in proc. of int. conference on expert systems, London 2-4th June 1987 learned Information, Oxford.

Green ,D. Colbert, M. & Long, J. (1989) ‘Some requirements and suggestions for a methodology to develop knowledge based systems’ in Ergonomics vol 32.(11) pp1499-1511. Taylor & Francis Limited.

Greenwell, M. (1988), Knowledge Engineering for Expert Systems, Ellis Horwood UK.

Greenwood, A. (1985) ‘Expert advice for bankers’, in Banking Technology, October 1985, paraphrased from Stevenson (1989) below.

Guida G. & Tasso C. (1989) 'Building expert systems: a structured bibliography' in G. Guida & C. Tasso (eds.) Topics in Expert System Design - Methodologies and Tools. North-Holland, Amsterdam NL.

Gullers, P (1988) 'Automation-skill-practice' in Goranzon, B. & Josefson, I (eds.) Knowledge,skills and Artificial Intelligence, Springer-Verlag, Berlin FRG pp.9-18.

Gutierrez, O. (1988) 'An experimental approach to support some aspects of expert systems analysis' in Berry, D. & Hart, A. (eds) Proc Human & Organisational issues of expert systems (Joint ICL and Ergonomics Society Conference) Stratford-Upon-Avon, UK 4-6th May).

Hammer, M. & Mangurian, G. (1987) 'The changing value of communications technology' in SMR Forum, Sloan Management Review Journal, Winter 1987, p65. USA.

Hammond,N., Jorgensen,A., Maclean,A., Barnard,P. & Long,J. (1983): 'Design practice and interface usability: evidence from interviews and designers' IBM Hursley Human Factors Report HF082 Hursley Park, Winchester, UK.

Harbridge, M (1989): Expert systems: how to identify suitable applications by es(Connect) consultants, 1989, London.

Harmon, P. (1990) Expert systems strategies, Arlington, MA, USA.

Harmon, P & King, D (1985) Expert systems: A.I. in business, John Wiley & Sons Inc.

Harris et al (1990) A report on a computer board initiative on knowledge based systems prepared by Harris, D.G., Freathy, P., Dawson, J.A. and Davies, B.K. Institute for Retail Studies, University of Sterling. (Unpublished).

Hart, A. (1986) Knowledge acquisition for expert systems, Kogan Page London.

Hayes-Roth, F. F. Waterman, D.A. and Lenat D.B. (1983, 1984) Building expert systems, Addison-Wesley, Reading MA, USA.

Hayward, S. (1986) 'A structured development methodology for expert systems' paper presented at KBS '86, on-line publications, Pinner UK.

Hayward, S Wielinga, B.J, Breuker, J.A. (1987) 'Structured analysis of knowledge 'in Int. J. Man-machine Studies Vol 26, pp487-498 1987.

Hazeltine (1987) Knowledge systems technology transfer, proc Annual Artificial Intelligence and Advanced Computer Technology Conf., Long Beach, CA, USA, April 22-24, pp271-272.

Herrmann, F. (1990) 'OCEX- Order Clearing Expert System' in Bramer, M. (ed.) Practical Experience in Building Expert Systems, pp29-49., John Wiley & Sons, UK.

Hewett, J. (1986) 'Commercial expert systems in the USA and Canada ' Proc of International Conference on Expert Systems, London, July 1986, Online Publications Limited, pp177-186

Hickman, F (1987) 'KBS: commercial fact or fiction ?' presented at KBS'87, On-line.publications, Pinner, UK.

Hickman, F (1989) 'The pragmatic application of the KADS methodology' in 5th Int. Conf. on Expert Systems London 6-8th June 1989 Learned Information Limited UK.pp55-66.

Hirschheim, R (1985) Office Automation: A social and organisational perspective, John Wiley & Sons UK.

Hirschheim, R & Smithson,S (1987): "Information systems evaluation: myth and reality" in Information Analysis (ed. Robert Galliers) Addison-Wesley.

Hoare, C.A.R. (1981) 'The emperor's old clothes' in Comms of the Association of Computing Machinery 24, pp75-83.

Holden, P.D. (1989): " 'Working-to-Rules': A Case of Taylor-made expert systems" in Interacting with Computers 1(2) August 1989, 197-219. Butterworths UK.

Holden, P.D. (1990a):'Human-Centred Expert Systems: A response to Taylorism and the Scientific Paradigm' in Interacting with Computers, Vol. 2 (1) Butterworths UK .

Holden, P.D.(1990b) 'Applying human centred concepts to the development of expert systems in manufacturing' paper presented at the International Workshop on Industrial Culture and Human-centred Systems at Tokyo-Keizai University, Tokyo, Japan May 14-18th. 1990.

Hollnagel, E. (1989) The reliability of expert systems Ellis Horwood Limited UK.

Holroyd, P., Mallory, G., Price, D. & Sharp, J. (1985) 'Developing expert systems for management applications ' in Omega vol 13(1) pp.1-11. 1985.

Hoos, I.R. (1979) 'Societal aspects of technology assessment' in Technological Forecasting and Societal Change 13, 3, 191-202.

Hurst, D. K. (1984) 'Of Boxes, Bubbles and effective management' Harvard Business Review, May/June 1984 USA.

Huxor, A. 'Knowledge based systems as a medium for skill enhancement ' in Berry, D. & Hart, A. (eds) Proc Human & Organisational issues of expert systems (Joint ICL and Ergonomics Society Conference May 4-6th.) Stratford-Upon-Avon, UK.

ICAM (1981), Integrated computer aided manufacturing (ICAM) architecture, Part II Vol. IV. Functional Modelling Manual (IDEFo) Softech Inc. Reference AFWAL-TR-81-4023.

ICAT (1986), 'IN-ATE: fault diagnosis as expert system guided search 'in Computer Expert Systems (eds. Block & Coombs), Springer-Verlag, New York, 1986 USA.

Ingersoll (1987) Technology in Manufacturing, Consultants Report, Ingersoll Engineers, Rugby, UK.

Ince, D. (1989) 'Software prototyping and Artificial Intelligence based software tools' in Research & Development in Expert Systems V. (ed. Kelly, B. & Rector, A.L.). Proceedings of Expert Systems 1988, 8th. Conf. of British Computer Society's Specialist Group on Expert Systems, Brighton, 12-15 December, C.U.P. UK.

ITSC: 1989 A development method for expert system environment R2.0 International Technical Support Centre, IBM, March 8th. 1989, Document No. GG24-3355, IBM USA.

Jain, H.K. & Chaturvedi, A.R. (1989) 'Expert systems problem selection: a domain characteristics approach' pp245-253. in Information and Management, vol. 17(1989), Elsevier, 1989.

Jamieson, R. & Szeto R. 'Impact of knowledge based information systems on organisations' in Journal of Information Technology, 4(3), September 1989, pp145-158.

Jenkins, P.G. (1987) 'Quantifying and assessing potential applications' paper presented at KBS '87, Online publications, p407., Pinner UK, 1987.

Johannessen, K.S. (1988) 'Rule following and tacit knowledge' in A.I. & Society vol. 2(4), pp.287-302, Springer-Verlag.

de Jong, L.S. (1988) 'Engineering of Expert Systems' in Information and Software technology 30(7) Sept 1988, pp.418 Butterworths UK.

Judd, J.S. (1990) Neural Network Design and the Complexity of Learning, MIT Press, Cambridge Mass, USA.

Kahn, G.S. & Bauer, M. (1989) 'Prototyping: tools and motivations' in Topics in Expert Systems Design, (G.Guida & C. Tasso eds.), Elsevier Science Publishers B.V. North Holland, Netherlands.

Kehoe, L. (1986) "White collar robots go to work", Financial Times, 5th. August, 1986, p.27., paraphrased from Stevenson (1989) below.

Kelly, B (1986) 'System development methodology: defining approaches'; paper presented at KBS 1986, On-line Publications, Pinner UK 1986.

Keller R. (1987) Expert systems technology: development and application; Englewood Cliffs, NJ, USA: Yourdon Press.

Kennett, D. & Totten, K.A.E.(1989) 'Experiences with an expert diagnostic system shell'; in British Telecom Journal, vol. 7(3), July 1989, UK.

Kidd, A. (1985) "What do users ask? - Some thoughts on diagnostic advice" in Merry, M (ed) Proc. Expert Systems 1985, Fifth Technical Conference of the BCS Specialist Group on Expert Systems Cambridge University Press, UK.

Klein, J. Powell, P. & Connell, C. (1988) 'Organisational structure: the implications of expert systems' in Berry, D. & Hart (eds.) proc. Human & Organisational issues of Expert Systems (Joint ICL and Ergonomics Society Conference) Stratford-Upon-Avon.

Kochen, M. Yufei, Y. & Barr C. (1986) 'On the value of human resources' in Technology forecasting and social change'; vol.30, pp.93-110.

van Koppen, J. (1988) 'A survey of expert system development tools ' pp 43-58., in Expert Systems in Engineering (ed. D.T. Pham) IFS Publications/ Springer-Verlag, 1988.

Koriba, M (1988)'System Modelling for the competitive edge ' CME, April 1988, pp38-39.UK.

Lawrence, A. (1989) 'Survey' in Informatics Journal, January 1989 p74. UK.

Le Fever, D.B. (1991) Technology Transfer and the role of intermediaries, Doctoral thesis (in preparation), Innovation and Technology Assessment Unit, Cranfield Institute of Technology, Cranfield, Beds. MK43 OAL.

Leininger, D.R. 'Criteria for selecting an expert system development tool ' pp59 -62. in proceedings of KBS '87, Online Publications, Pinner, UK.

Leonard-Barton, & D. Sviokla, J (1988) 'Putting expert systems to work' in Harvard Business Review, March/April 1988, USA.

Leonard-Barton, D. 'Implementing innovations: the automation of knowledge-based production tasks' working paper #9-785-003, Harvard Business School, Boston MA., July 1984.

Liebowitz, J. (1989), 'Problem selection for expert systems development' in Structuring Expert Systems, (ed. Liebowitz, J. & Desalvo, D.), Prentice Hall Inc. USA.

Linstone, H.A. (1978) 'The use of structured modelling for technology assessment' Report 78-1 Futures Research Institute, Portland State University USA pp132.

Linstone, H.A. (1981) 'The Multiple Perspective Concept' in Technological Forecasting and Social Change 20, 275-325.

Linstone, H.A. (1984) Multiple perspectives for decision-making, North Holland Amsterdam.

Lock-Lee, L.G. (1986) 'A preliminary methodology - systems analysis for knowledge based systems in industry'. Proc ACC 86 461-473.

Lu, M. & Guimaraes, T. (1989) 'A guide to selecting expert systems applications', in the Journal of Information Systems Management , Spring 1989, pp8-15. USA.

Lundeberg, M., Goldkukl, G. & Nilsson, A. (1981) Information Systems Development: A Systematic Approach, Englewood Cliffs, Prentice Hall, 1981. USA.

Lydiard, T.J. (1989) 'The order of crystallisation' in PC Tech. Journal, October, 1989, pp22-30.

Luconi, F.L., Malone, T.W., Scott-Morton, M.S. (1986) 'Expert Systems: the next challenge for managers' in Sloan Management Review, Summer 1986, Mass., USA.

Maji, R.K. 'Tools for the development of information systems in CIM ' in Advanced Manufacturing Engineering, Vol 1, October 1988, p26. Butterworths & Co. UK.

Markus, M. L. (1984) Systems in Organisations: Bugs and Features, Pitman Press, USA.

Markus, A. & Hatvany, J. (1987) 'Matching A.I. tools to engineering requirements' pp311-315., Annals of the CIRP vol 36(1), Budapest. Hungary.

Markus, M. L. & Robey, D. (1988) 'Technology and Organisational Change: causal structure in theory and research' in Management Science may 88 583-598.

Martin, A. & Law, R.K.H. (1988) 'Expert systems for selecting expert system shells' in Information and Software Technology vol. 30 (10), Butterworths, UK.

Martinez, D. & Sobol, M. (1988) 'Structured analysis techniques for the implementation of expert systems' in Information and Software Technology 30 (2), pp. 81-88, Butterworths UK.

Mayhew, PJ. & Dearnley, PA 'An alternative prototyping classification' The Computer Journal Vol 30(6) pp481-484, 1987.

McDermott, J. (1981) ' R1: The formative years ' in A.I. Magazine, vol. 2 (2), pp21-29, 1981 USA.

McFarlan, F.W. (1984) 'Information Technology changes the way you compete' in Harvard Business Review, May-June, 1984 pp98-103.USA.

McLoughlin, I., Rose H. & Clark, J. (1985) ' Managing the Introduction of new technology ' in Int. Journal of Management Science 13(4) pp251-262., 1985 Pergamon press.

Meyer, R.J. (1987) 'Artificial Intelligence and Expert Systems: In pursuit of CIM' in CIM Technology, February 1987, pp15.

MI (1989) Manufacturing Intelligence: market surveys and applications; report by the Department of Trade and Industry, MIT Division, Crown Copyright, 1989 UK.

Michaelson, R.H., Mitchie, D. & Boulanger, A. 'The technology of expert systems ' BYTE Magazine, vol 10 (4) pp303-312, 1985.

Milne, R. (1987) 'Strategies for Diagnosis' in IEEE Transactions on Systems, Man & Cybernetics, vol. SMC-17 (3) May/June 1987 USA. pp.333- 339.

Milne, R. (1990) 'Intelligent Data Interpretation'; pp103-123. in Practical Experiences in Building Expert Systems, Bramer, M (ed.), John Wiley & Sons UK.

Monk, P. 'Characteristics of IT innovation' in Journal of Information Technology, December 1987, vol 2(4) pp164-170.

Montgomery, T.A. & Crispin, E. (1989) 'GEMINI: government expert systems methodology initiative' in in 5th Int. Conf. on Expert Systems London 6-8th June 1989 Learned Information Limited UK.pp.45-54.

Moralee, S (1987) 'Expert systems - some user experiences' in Intelligent knowledge based systems: an introduction. ed by T. O'shea, J Self & G. Thomas, Harper & Row, 1987.

Morieux, Y. & Sutherland, E. (1988) 'The interaction between the use of information technology and organisational culture' in Behaviour and Information Technology vol 7 (2). pp 205-213 .

Morris, A (1987) 'Expert systems -interface insight' in Diaper, D. & Winder, R. (eds.), People & Computers III. Proc. 3rd. Conf. BCS HCI SG Cambridge University Press UK.

Motta, E. Rajan, T. & Eisenstadt, M. (1989) 'A methodology and tool for knowledge acquisition in KEATS-2' in G. Guida & C. Tasso (eds.) Topics in Expert System Design - Methodologies and Tools. North-Holland, Amsterdam NL.

Mumford, E and Weir, M. (1979) Computer systems in work design. The ETHICS Method, New York, J. Wiley & Sons p37.

Mumford E., & Henshall, D. (1983) Designing Participatively, Manchester Business School, 1983.

Mumford, E. (1987), 'The successful design of expert systems- are means more important than ends?' paper presented at Oxford P.A. Templeton College, Oxford, UK September 1987, (unpublished).

Mumford, E (1989) 'Managing complexity: the design and implementation of expert systems' in Knowledge Based Management Support Systems (eds. Doukidis, G.I., Land, F. & Miller, G.), pp304-312. Ellis Horwood UK.

Murdock, H. (1990) 'Choosing a problem - when is A.I. appropriate for the retail industry ?' Expert Systems Journal, February 1990, vol. 7(1), pp42-49. UK.

OU (1987) Artificial Intelligence Applications to Training; A research study by the Open University for the Training Technology Section of Manpower Services Commission, Crown Copyright, November 1987, UK.

Neale, I.M. (1988) 'First generation expert systems: a review of knowledge acquisition methodologies' the Knowledge Engineering Review, vol 3(2) p105-145. UK.

NIEVR (1989) 'Expert systems and its impact on labor ' National Institute of Employment and Vocational Research, NIEVR Publication Report No. 5. February 1989, Tokyo, Japan.

Niwa, K. (1988) 'Knowledge transfer: a key successful applications of knowkedge-based systems' in The knowledge Engineering Review, vol. 2(2), p147-151. 1988.UK.

O'Neill, M. & Morris, A. (1989) 'Expert systems in the United Kingdom: an evaluation of development methodologies', pp.90-99., in Expert Systems Journal, April 1989, vol 6(2), Learned Information UK.

Ostberg, O. (1988): 'Applying Expert systems technology: division of labour and division of knowledge' in Goranzon, B. & Josefson I. (eds.) Knowledge, Skill and Artificial Intelligence, pp169-183. Springer Verlag, Berlin, FRG.

Ovum (1988a) Expert systems: market and suppliers; by Johnson, T., Hewett, J., Guilfoyle, C. & Jeffcoate, J. Ovum Limited, 1988. UK

Ovum (1988b): Expert Systems in Britain; (with Segal Quince & Wicksteed), Report to the Department of Trade & Industry, British Crown 1988. UK

Parker, M., Benson, R.J. & Trainor, H. (1988) Information Economics, Prentice-Hall Int. 1988, USA.

Partridge, D. (1988) 'A.I. and Software Engineering: a survey of possibilities' in Information and Software Technology, vol 30(3) April 1988, pp146-152.Butterworths UK.

Partridge, D. & Wilks, Y. (1987) 'Does AI. have a methodology which is different from software engineering ?' in Artificial Intelligence Review vol 1(2) pp111-120.

Pedersen, K. (1989) Expert systems programming: practical techniques.for rule based systems John Wiley & Sons 1989 USA.

Perrolle, J.A. (1986) 'Intellectual assembly lines: the rationalisation of managerial, professional and technical work' in Computers and the Social Sciences 2(3) pg. 111-121; Paradigm press USA.

Peters, T.J. & Waterman, R.H. (1982) In search of excellence Harper & Row, New York USA.

Peterson, J.L. (1981) Petri-net theory and the modeling of systems; Prentice-Hall Inc, USA, 1981.

Pople, H. (1977) 'The formation of composite hypotheses in diagnostic problem solving- an exercise in synthetic reasoning' in Proc IJC A1-5 pp1030-1037. USA.

Porter, M E & Millar, V E (1985) 'How information gives you competitive advantage' Harvard Business Review July/August 1985 p149-160 USA.

Preece, A. & Gregory, S. (1988) 'Shell -shocked or why round applications don't fit into square tools'; paper presented at FringES '88, 8th. Conference of the British Computer Society Specialist Group on Expert Systems, Brighton, 12-15 December 1988.

Prerau, D.S. (1985) 'Selection of an appropriate domain for an expert system'; in A.I.Magazine, Summer 1985, pp26-30. USA.

Prerau, D.S. (1989) 'Choosing an expert systems domain'; in Topics in Expert Systems Design (eds. G. Guida & C.Tasso); pp27-43, Elsevier Science Pubs, North Holland, 1989, UK.

Primrose, P.L. & Leonard, R. (1988) ' An approach for negating the view that technology "competes with humans" ' in Robotica, 5, 251-255.

Probert, D (1989) 'Embedding A.I. in the enterprise ' in Research & Development in Expert Systems V. proceedings of Expert Systems 1988, 8th. conference of British Computer Society Specialist Group on Expert Systems, Brighton, 12-15 December 1988; ed B Kelly & A.L. Rector. CUP.

Quinlan, I. & Ross, J. (1989) 'Application of Expert Systems: Volume II ' from 3rd and 4th Australian Conferences, Addison Wesley, 1989 1st ed.

Rajin, T., Motta, M. and Eisenstadt (1989) 'ACQUIST: A Tool for Knowledge Acquisition ' pp113-125. in Research & Development in Expert Systems V. (ed. Kelly, B. & Rector, A.L.). Proceedings of Expert Systems 1988, 8th. Conf. of British Computer Society's Specialist Group on Expert Systems, Brighton, 12-15 December, C.U.P. UK.

Rasmussen,L B & Tottrup (1990) 'Anthropocentric principles -tools and organisation of industrial design'; paper presented at International Workshop on Industrial Culture and Human Centred Systems, Tokyo Keizai University, Tokyo, May 14-18 1990.

Rauner, F. & Ruth K. (1990) 'Prospects of anthropocentric production systems and industrial-cultural variability' paper presented at International Workshop on Industrial Culture and Human Centred Systems, Tokyo Keizai University, Tokyo, May 14-18 1990.

Ravden, S.J. Clegg,C.W. & Corbett, J.M (1987) Report on Human Factors Criteria for CIM Systems & Methods to Enhance their usability, Esprit CIM project No. 534.

Reed, E.S (1987) 'Artificial Intelligence or the mechanisation of work', Open Forum in A.I. & Society (ed. K. Gill), 138-150 Springer Verlag.

Rees, P. (1988) 'User Participation - Desires or Reality: The case of expert systems' in Berry, D. & Hart, A. (eds) Proc Human & Organisational issues of expert systems (Joint ICL and Ergonomics Society Conference) Stratford-Upon-Avon, UK.

Rifkin, G.(1985) 'Towards the fifth generation ' editor, Computerworld, Update, pp3-24., 6th. May, 1985.

Rockart, J.F. & Crezenzi, A.D. (1984) 'Engaging top management in I.T.' Sloan Management Review, Summer 1984.

Rodi, L.L. & Pierce, J.A. & Dalter, R.E. (1989) 'Putting the expert in charge: graphical knowledge acquisition for fault diagnosis and repair ' in SIGART Newsletter, April 1989, no. 108, pp56.

Rolston, D.W. (1988) Principles of intelligent and expert systems development McGraw-Hill USA.

Roman, D.D. *et al* (1983) International business and technological innovation; (with Puertt, F. Jnr.), Chapter 7, p159, North Holland Amsterdam.NL.

Rosenbrock, H.H. (1988) 'Engineering as an Art'; in A.I. & Society (ed. K. Gill) 2(4) pp.315-320, Springer-Verlag.

Ross, D.T. (1976) 'Structured analysis'; proceedings of the joint IBM/ University of Newcastle-Upon Tyne Seminar on Computing Systems Design (7th-10th September, 1976).

Rothenberg et al (1987), J.G., Paul, J., Kameny, I. Kipps, J.R., Swenson, M. Evaluating expert system tools: A framework and methodology; Report R-3542-DARPA, published by the RAND Corp.

Rothwell, R (1978) 'Some problems of technology transfer into industry: examples from the textile machinery sector, in IEEE Transactions on Engineering Management vol. EM-25 (1) February 1978, p15-20. USA.

Rouse, W.B. & Nancy, M. (1986) 'Understanding and enhancing user acceptance of computer technology'; pp.965-973. in IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-16 (6), November/ December 1986.

Rowen, R. (1990) 'Software project management under incomplete and ambiguous specifications'; in IEEE Transactions on Engineering Management, vol. 37(1), Feb 1990, p10-21.

SEAI (1988) Artificial Intelligence. Applications for business management, SEAI Technical Publications, Prentice-Hall, (2nd. ed.) USA.

Sell, P (1987): 'Strategic issues in introducing knowledge-based systems' in KBS 1987, Online Publications, Pinner, UK p401.

Sharman, D. & Kendall,E.J.M. (1988) 'A case study: acquiring strategic knowledge for expert systems development' in IEEE Expert Journal, Fall 1988, pp32-40 USA.

Shelton, J A (1989) 'The Individual "working-to-rules": reducing determinism in taylor-made expert systems' in interacting with computers: vol.1 (3), pp.338-342.

Shortcliffe, E. H. (1976) Computer based medical consultation: MYCIN, Elsevier, New York. USA.

SI (1990) 'Expert boom in Europe' editorial in Systems International Journal, May 1990. p12. UK.

Skingle, B. (1987) 'The Validation of Knowledge based systems' in KBS '87, On-line Publications, Pinner UK pp27-36.

Slagle, J.R. & Wick, M.R. (1988) 'A method for evaluating candidate expert systems applications' in the A.I. Magazine, Winter 1988, pp44-53. USA.

Slatter, P., Normura, T. & Lunn, S.(1988) 'A Representation for manufacturing sequencing knowledge to support cooperative problem solving' in Berry, D. & Hart, A. (eds) Proc Human & Organisational issues of expert systems (Joint ICL and Ergonomics Society Conference) Stratford-Upon-Avon, UK.

Smith, D. (1987) 'A.I. and the human EPROM' in A.I. & Society Journal, vol 1(2), pp146-150, Springer-Verlag.

Smith, R.G. (1984) 'On the development of commercial expert systems' AI Magazine 5(3) 61-73 USA.

Spinas, P., & Ackermann, D.(1989) 'Methods and tools for software development: results of case studies', pp511-521 in Man-Computer Interaction Research, MACINTER-II, ed. by F. Klix, N.A. Streitz, Y. Waern & H. Wandke, North Holland 1989.

Stefik, M. & Bobrow, D.G. (1986) 'Object Oriented programming: themes and variations ' in A.I. Magazine vol. 6(4) pp40-62.

Stevenson, H. (1989) 'Expert Systems in the UK financial services sector: a symbolic analysis of the hype'; in knowledge-based management support systems; pp.276, (eds. Georgios Doukidus, Frank Land, & Gordon Miller), Ellis Horwood, 1989.

Stow, R, Lunn, S. & Slatter, P (1986) 'How to identify business applications of expert systems' paper presented at Second International Expert Systems Conference, London, October 1986, Proceedings published by Learned Information.

Strassmann, P.A. (1985) Information payoff: the transformation of work in the electronics age, New York, Free Press USA.

Sviokla, J. (1986) 'The business implications of expert systems'; in Database, pp.5-18, Fall 1986, USA.

Taylor, D. (1988) 'Advanced Manufacturing Technology: the implications for human resource strategies' Applied Ergonomics, March 1988, pp17-20.

Taylor, F.W. (1911) 'Principles of Scientific Management' in Taylor F.W. (ed.) Scientific Management, Harper, New York. USA.

Teschler, L. (1988) 'Trends in the marketing of expert systems technology ' in The Knowledge Engineering Review, pp 175 -177, vol. 2 No. 2. UK.

Towriss J.G. (1988) Expert Systems and Road Freight; Centre for Transport Studies, Working Paper, Cranfield Institute of Technology. (unpublished).

Turban, E. (1988) 'A Review of Expert Systems Technology'; IEEE Transaction on Engineering Management, vol. 35(2), May 1988. pp71-81.

Veryard, R. (1987) 'Implementing a methodology' in Information & Software Technology vol 29(9) November 1987, Butterworths UK.

Wainwright, D. & Bowker, P (1987) Towards a Holistic view of manufacturing information systems implementation' paper presented at CIM 2000, Cambridge University Press.

Wang, M. & Smith, G.W., (1988), 'Modelling CIM systems Part I: methodologies' in Computer Integrated Manufacturing Systems, Vol 1(1), February 1988, pp13-17, Butterworths & Co. UK.

Walters, J.R.& Nielson, N.R..(1988) Crafting knowledge based systems: expert systems made simple/realistic , John Wiley & Sons Inc., USA, (Chapter 5. 'Conducting the feasibility study').

Ward, J.M. (1986) 'Strategic Information (IS) Management' in Information Management in Competitive Success, ed. P.M. Griffiths, Pergamon Press, 1986.

Ward, J.M. (1988) 'Information Planning for Strategic Advantage' Journal Information Technology, vol 3(3), September 1988, UK.

Waterman, D. (1986) A guide to expert systems Addison Wesley USA.

Weitz, R R (1988) 'Technology, Work and the Organisation: the impact of expert systems' INSEAD working paper, No. 88/51, July 1988.

Weizenbaum, J (1976) 'Computer Power & Human Reason: from judgement to calculation' W.H. Freeman, USA.

Welbank, M (1983) A review of knowledge acquisition techniques for expert systems; British Telecom Research, Martlesham Heath UK.

Wensley, A. 'Research directions in expert systems'; in knowledge-based Management Support Systems;(eds. Georgios Doukidus, Frank Land, & Gordon Miller), Ellis Horwood, 1989.

Wess, B.P. 'Commentary on the commercialisation of knowledge engineering: enterprise and product development'; in the Knowledge Engineering Review , pp. 169-174; vol. 2(2), UK.

Weitz, R.R. (1988) Technology, Work and the Organisation of Expert Systems, INSEAD Publication, Paris, France 1989.

West, A. & Farr, J.E. (1989) 'Innovation at work: psychological perspectives' in Social Behaviour, vol. 4. pp15-30 (1989).John Wiley & Sons Limited UK.

Whitaker, R. & Ostberg, O. (1988) 'Channelling Knowledge: expert systems as communication media' AI & Society 2(3).

Whybrow, M. (1988) 'Making everyone an expert' in Informatics Journal, pp.59-64, December 1988, UK. (Describes a Science Policy Research Unit report which shows the failures of the UK Alvey IKBS programme in transferring expertise from academic to industrial sectors).

Wiig, K. (1990) Expert systems: a manager's guide, Management Development Series No. 28, International Labour Office, Geneva, Switzerland.

Williams, C. (1986) 'Expert systems, knowledge engineering and A.I. tools- an overview ' in IEEE Expert Journal, Winter 1986, p66. USA.

Wilson, B (1984) Systems: Concepts, Methodologies and Applications, John Wiley & Sons Chichester, UK.

Wilson, M. Duce, D. & Simpson D. (1989) 'Lifecycles in software and knowledge engineering: a comparative review' in the Knowledge Engineering Review Vol 4(3) 1989, pp189-204.

Winograd, T & Flores F (1989) Understanding Computers & Cognition; Ablex Publishing, USA.

Wolf, W.A. 'Knowledge acquisition from multiple experts' in SIGART Newsletter, April 1989, No. 108, pp138-140.

Wood-Harper, A.T., Antill, L. & Avison, D.E. (1985) Information Systems Definition: The Multi-View Approach Blackwell Scientific Publications

Worden, R. (1988) 'Integrating KBS into IS: the challenge ahead ' presented at Software Tools '88, On-line publications Limited, Pinner UK. 1988 pp227-237.

Worden, R. (1988) 'Processes of knowledge and software' pp139-159. in Research & Development in Expert Systems V. (ed. Kelly, B. & Rector, A.L.). Proceedings of Expert Systems 1988, 8th. Conf. of British Computer Society's Specialist Group on Expert Systems, Brighton, 12-15 December, C.U.P. UK.

Wyatt, T. (1988) 'Methods and techniques for systems specification' in FMS Magazine vol. 6(2), pp91-95, April 1988, IFS Publication, Bedford, UK.

Young, R. (1989) 'Human Interface aspects of expert systems' pp20-34, in Artificial Intelligence- social aspects (ed. Linda Murray & J. Richardson), Open University Press.

PETER D. HOLDEN

A MULTIPLE PERSPECTIVE APPROACH TOWARDS THE ASSESSMENT AND DEVELOPMENT OF EXPERT SYSTEMS IN MANUFACTURING

INNOVATION & TECHNOLOGY ASSESSMENT UNIT
(SCHOOL OF MANAGEMENT)

VOLUME II.
(APPENDICES)

Ph.D. THESIS

CRANFIELD INSTITUTE OF TECHNOLOGY

INNOVATION & TECHNOLOGY ASSESSMENT UNIT
(SCHOOL OF MANAGEMENT)

Ph.D. THESIS

Academic Years 1987-90

Peter D. Holden

A Multiple Perspective Approach Towards the Assessment and Development of Expert Systems in Manufacturing

VOLUME II.
(APPENDICES)

Supervisors: Professor M. Cordey-Hayes
Dr. J.G. Towriss

February 1991

List of Appendices in Volume II.

Appendix I

Expert Systems: Technical Reference and Glossary of Terms

Appendix I: Part A

Reference: Understanding Expert Systems Technology

The purpose of this section is to provide a reference source on the technical design, tools and construction of expert systems technology. It is intended as a complement to the coverage of these topics in the main text, whilst providing more detail and introducing the reader to the terminology used in this field. A summary of expert system terms is provided in the Glossary in Section Two of this appendix.

1 How do Expert Systems Differ from Conventional Programs?

Expert systems differ from existing computer programming techniques in a number of ways and these differences allow expert systems to solve some problems which otherwise could not have been solved using conventional techniques. However, most problems are not exclusively one or the other type;· both may be feasible solutions to a particular problem, but expert systems may be more desirable because of the ease of construction, maintenance and other benefits which are described later.

A principal difference between conventional programs and expert systems lies in the difference between knowledge and information processing. Knowledge is made up of ideas, concepts, facts, rules, procedures, theories, relationships and ways to apply these to practical problem-solving. In some applications, the knowledge for an expert system can come straight out of a book, a policy or standard, manual or other similar sources. By reformatting the knowledge in existing documentation, an expert system can be created. However, in most applications, the sort of knowledge that is the most effective in expert systems and proves to be the most valuable is **heuristic** knowledge. Heuristic knowledge cannot be acquired from a manual or through instruction, but requires substantial experience and exposure to a wide variety of problems and situations. Heuristic knowledge lets experts solve problems quickly primarily because they know what works and what doesn't work in a given situation. This type of knowledge is inherently uncertain and notional and the expert may find it difficult to explain why such knowledge was used in a particular decision.

Conventional programming techniques make use of information rather than knowledge. Information is data and facts and can be in the form of spreadsheets, databases, word processing documents or engineering computations for example. Conventional programs, such as Basic, Fortran and Pascal for instance, are algorithmic in that programs are written that tell the computer precisely what to compute and how to make the computation in order to solve a given problem. This works well with repetitive tasks and such techniques are highly effective in storing large amounts of information and 'number crunching'. It also makes it easier to retrieve data because numbers and text can be easily coded into binary form suitable for processing and manipulation. Where conventional programs use data to process information, expert systems add meaning and apply information to create knowledge. Here, knowledge adds meaning and importance to information through analysis and understanding of how it is made up and applied.

To solve a conventional computer problem, a programmer first analyses the problem to determine exactly what the inputs are and what the desired outputs should be. The programmer then arrives at an algorithm that processes the inputs to produce the output. That algorithm is a clearly defined incremental procedure that does the desired manipulation. The programmer converts the algorithm into a sequential list of instructions, statements, or commands as defined by a programming language. In turn, that programming language produces the binary code that is stored in the computer memory. When executed, the program solves the problem exactly as specified. This form of processing is **procedural** because algorithms detail specifically how to solve the problem. By contrast, expert systems

use **symbolic** processing in order to represent knowledge in a computer. A symbol is nothing more than words, letters and numbers which are used to represent objects, actions and their relationships. By representing relationships in this way, it is possible to use search and pattern-matching techniques to identify satisfactory solutions where an optimum algorithmic solution may not exist. For example, the scheduling of the shop-floor makes use of precise inputs and specifications such as due-dates, operator sequences and materials to determine how to allocate the available machines, materials and personnel to finish a variety of parts orders in the least amount of time or by the scheduled due date. However, there is a qualitative dimension to scheduling also, where no optimal solution is attainable and decisions are based upon trade-offs and compromises in order that constraints are satisfied: it is here where expert systems may be used to great effect.

Table 1 summarises the differences between the two types of programing. It should be noted however that since such distinctions were made in the mid-1980s, there have been significant advancements in conventional computing methods and a trend towards the development and use of ‘hybrid’ software which makes use of both techniques. The divide between conventional and expert systems programs is therefore less unequivocal.

Conventional Programs	**Expert Systems**
Algorithmic	Heuristic
Right/Wrong	Probabilistic
Precise	Notional
Repetitive Process	Inferential Process
Solution Steps Explicit	Solution Steps Implicit
Works with data	Works with information
Procedural processing	Symbolic processing
Correct answers required	Some Wrong Answers

Table 1: A Summary of the Stated Differences between Conventional & ES Programs

Thus, for problems which require human judgement or where data is not complete, the environment is fuzzy, ambiguity exists or information is confusing, then it is likely that the best solution is to make use of expert systems. There are further differences between conventional programs and expert systems and these become evident through a discussion of the differences in architecture.

2. The Construction of Expert Systems

Figure 1A. shows a block diagram of an expert system. These are the main elements of an expert system, although not all types have every subsection. All expert systems have the knowledge-base, inference engine, and user interface, but other features will vary from program to program.

2.1. Knowledge-base

The symbols used to represent knowledge in an expert system are stored in the computer memory as characters or strings. Through the use of symbols, it is possible to create a **knowledge-base** which states various facts about the objects, actions, or processes, and how all of them are inter-related. In creating the knowledge-base, there are no specific guide-lines for storing information, so that new rules can be created and information added or taken away, without the need to follow rigid hierarchical or database sequences for example during information entry.

2.2. Inference Engine

Once a knowledge-base has been developed, a method is devised to actually use it. Basically, a program is needed that will use the knowledge to reason and think in an effort to solve a particular problem. This kind of program is generally referred to as an inferencing program or **inference engine** which is designed to make decisions and judgements based upon the symbolic data in the knowledge-base. The inference engine accepts external inputs about the problem and then attempts to apply the available knowledge to its solution.

The inference engine manipulates the symbolic information in the knowledge-base in a number of ways according to the **problem solving methods** used, and by the organisation of knowledge, or **knowledge representation**, in the knowledge-base, both of which are discussed in later sections. Despite the possible variances, the basic process centres on search and pattern-matching. The inferencing program is provided with some initial inputs that provide sufficient information for the program to begin. Using these initial inputs, the inference engine searches the knowledge-base looking for matches. It also controls the sequence in which rules are examined. The search continues until a solution is found. The initial search may turn up a match that, in turn, leads to another search and another match and so on. This process is fully automated and is totally invisible when running a consultation.

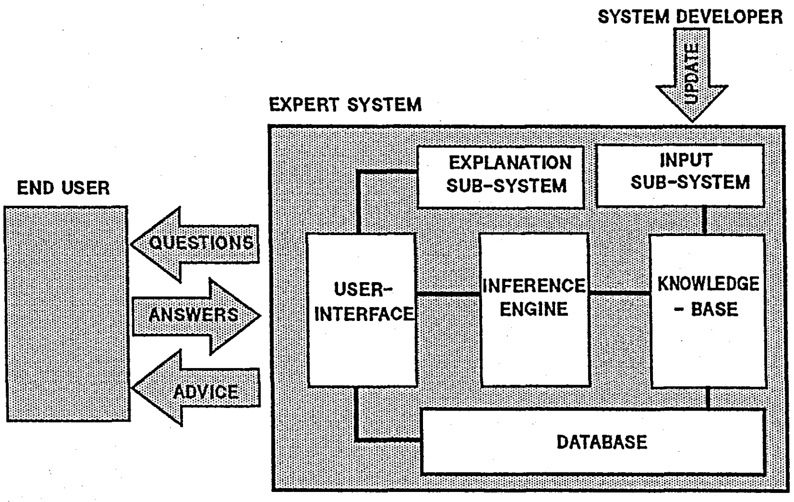


Figure 1A: Diagrammatic Representation of an Expert System

It can be seen from **Figure 1A.** that the control mechanism, the inference engine, is separated from the knowledge-base. A consequence of this is that it is possible to change the knowledge-base without having to re-write the program. This has benefits in terms of the maintenance and update of existing expert systems, but also in terms of increased productivity in developing new applications. A further benefit of separating knowledge and control is that the knowledge in the expert system can be built up bit-by-bit, a rule at a time. The system can still work even though the knowledge-base is not complete. This means that information and knowledge can be verified as the system is developed.

2.3 User Interface

The user interface is a collection of programs that work with the inference engine and the knowledge-base to provide a convenient means of two-way communication with the user. The user interface gathers input data in one of two ways. The expert system may ask questions to which the user replies by typing in answers. Or the user interface may operate by menus. These offer multi-choice questions, asking the user to select the correct choice from among several alternatives.

The user interface is also used during the inferencing process. Should the inference engine decide that there is insufficient information to arrive at a decision, it notifies the user via the monitor. It may do this by picking up a portion of the rule being tested and reformat it into a question. When the user enters the data asked for, the inference engine can continue its reasoning process. When this process is complete, an output result is presented on the monitor. Expert systems cannot always reach the 'right' answer, so the screen output may state that no conclusion or only a guess or estimate of the output could be reached. Or the result may be an answer, but with a degree or 'rightness' quantifying it.

2.4. Database

The data base, or working memory, is the portion of the computer's memory set aside for keeping track of inputs, intermediate conclusions, and outputs. Initial inputs are stored in the data base. As the knowledge-base is interrogated by the inference engine, conclusions are drawn and stored in the data base. The inference engine uses these intermediate conclusions as new inputs to search for new matches. At the end of a run, the data base contains the entire chain of facts or assumptions that include not only those entered initially but also those that were concluded along the way to the final decision.

2.5. Explanation Subsystem

Most expert systems contain a section designed to explain to the user what line of reasoning was used to reach its conclusion. Users put more trust in an expert system decision when they can understand its line of reasoning. With an explanation subsystem, users can ask "why?" or "how?" and the system can give an answer. For example, if the system asks for additional input data, the user might wish to ask why. Usually the system would respond by saying it needs the information to evaluate a particular rule and it might even show which rule or condition it is trying to satisfy. Users can also use the explanation facility for debugging purposes. During development, it can serve as a way to gain feedback on knowledge-base construction and sequence, enabling users to readily test the system on practical problems.

The capability of explanation by an expert system is often a valuable feature where critical decisions are being made. If users can question the decisions of the system and explore alternatives, they bring their own knowledge to bear on the problem and make the conclusion a considered 'joint' decision rather than a command. It must be understood however, that not all applications require or warrant an explanation. For example, a real-time process planning expert system makes instantaneous decisions which are used immediately to structure further conclusions: there is no time to question or analyse a decision until after the event as a decision report.

2.6. Knowledge Input Subsystem

Most expert systems contain a program or set of programs that enable the users to add to or modify the structure of the knowledge-base. Most expert systems operating in a manufacturing environment use knowledge that is changing, and therefore the knowledge-base must be continually modified to reflect these changes. The knowledge input subsystem provides a means of adding new rules and existing information. The subsystem is usually a specialised text editor which reads new information and revisions and performs knowledge-base updates, acting like a compiler converting the text into the correct format.

Maintaining the knowledge-base is essential. If the knowledge in a particular field, or **domain**, changes frequently, the expert system begins providing inconsistent and increasingly erroneous answers which are very difficult to trace back.

3. Knowledge Types and Representation Techniques in Expert Systems

It was mentioned earlier that the knowledge in the knowledge-base is symbolic. This section extends this further to look in detail at the different types of symbolic knowledge and how each type is translated into a form that the computer can use.

3.1. Types of Knowledge

Two basic kinds of knowledge can be put into a knowledge-base: declarative and procedural. Most expert systems will contain both.

Declarative knowledge, also called descriptive knowledge, is primarily a statement of fact about people, places or things. Declarative knowledge allows information to be stated explicitly, deduce relationships and classify objects. Using declarative knowledge nothing is explained, but it is possible to present truths and their association with each other. In expert systems, declarative knowledge representation schemes include semantic networks, frames and production rules (which are described in the next section).

Where declarative knowledge is responsible for the 'what?', **Procedural** knowledge describes the 'how?' of a knowledge situation. Procedural or prescriptive knowledge is explanatory; it provides a way of applying the declarative knowledge and shows procedures for performing a course of action. Procedural knowledge recommends what to do and how. A list of instructions for installing a program on a hard disk storage unit and a step-by-step sequence for assembling electric motors are both examples of procedural knowledge. Procedural knowledge is represented in expert systems as production rules and scripts. A third related type of knowledge is **control** knowledge; this states how procedural knowledge should be applied. It is a top level or **meta** control which overlooks the execution of production rules so as to avoid conflict between alternative answers to a given situation. A final type of knowledge described earlier is **heuristic** knowledge. This makes use of procedural, declarative and control knowledge to capture the inexact, non-rational rules-of-thumb of the expert.

3.2 Representation Schemes

Knowledge representation formalises and organises the knowledge of the expert. There are several ways of representing knowledge. The three most popular of these are **rules, frames** and **semantic nets** and these will be discussed here. Rule-based representation is a surface representation and good for descriptions of procedural knowledge, whereas schemes using frames and semantic nets are **deep** representations which capture both heuristic and declarative knowledge. Very often a balance is made in the construction of simple rule based representations, with limited functionality, and complex deep representations.

3.2.1 Rule Based Representations

The common method of representing knowledge in expert systems is with **production rules**, or simply 'rules'. These are two-part statements that contain a small increment of knowledge. The domain, or subject, to be represented in an expert system is divided up into many small chunks of knowledge. The two parts of a rule are a premise and a conclusion, a situation and an action, or an antecedent and a consequent. These statements are written in an IF-THEN format. The first part of the rule is prefaced by the word IF, to state a situation or premise. The second part of the rule is prefaced with THEN to state an action or a conclusion. Production rules are simple to understand and use and are ideally suited to a wide range of heuristic knowledge. Most knowledge domains are easily represented in this format. Some examples of rules are shown here:-

IF	the System fails to reboot
AND	the Vax machine fails to locate a disc
AND	the screen fails to show a VMS message
THEN	turn the DC off and on and repeat reboot

In this rule, the premise or situation begins with an IF but also contains two AND statements that are part of the situation or premise. If the conclusion is to be true, then all three statements in the premise must be true. There is no restriction on the number of AND statements in a premise.

IF	the switch is off
OR	the fuse is blown
AND	the motor fails to run
THEN	the circuit will be off

Another way to make a knowledge statement is to use OR statements in the premise. Along with the initial statement, one or more OR statements may also be included. In rules of this type, the conclusion stated in the THEN part of the rule will be true IF any one or more of the statements in the premise is true. This format provides a flexible way of representing some types of knowledge. Since rules represent only tiny increments of knowledge, it takes a considerable number of them to represent the knowledge of a particular domain. Small expert systems may have only ten or twenty rules but the more useful systems usually have well over a hundred. Large systems may have up to 6000 rules.

A main benefit of rules is that they facilitate creation, modification, and maintenance of a knowledge-base because the knowledge is modularised. Since much domain knowledge changes over time, new rules must be added and old rules revised to keep the knowledge-base current. With rules, these changes can be made relatively quickly. A further benefit of rules is that they are effective in handling uncertainty. There are various methods of doing so according to the structure of the information and requirements of the application.

3.2.1.1. Measures of Confidence and Handling Uncertainty

It was mentioned earlier that algorithmic software is not capable of dealing with ambiguity and uncertainty. An algorithm, such as a formula, needs specific input values supplied before it can compute an output. If you give correct inputs it will give correct outputs. Expert systems by contrast don't always need perfect inputs and outputs and it is a useful attribute of production rules that they are able to incorporate uncertainty into the knowledge set. When the expert system asks the questions, the user may not be able to supply the desired answers either because the answer may not be known at all, or the user is uncertain over its validity. Most expert systems are able to deal with these situations in a number of ways which fall into two broad categories:

a) Numerical Measures of Uncertainty

Numerical measures of uncertainty take the form of allocating a certainty value to an event (as a numerical threshold or probability limit) and the system giving the solution (or solutions) with various degrees of certainty. The benefits of such an approach are that an exact figure can be used and derived allowing for precise calculations of uncertainty, particularly when cumulative, and avoiding problems of interpretation. The overriding limitation of numerical measures is that often the events within the subject cannot be measured numerically, or the expert cannot think in terms of allocating values of probability against a particular decision. For example, three different experts may assign an outcome with a probability of 0.4, 0.5 and 0.6 respectively. In some instances this discrepancy might not affect the solution decision whilst in other situations it may do so in a critical way. A second and more controversial issue, which will not be discussed here, is the mathematical integrity in the use of certainty factors and probabilities for the representation of uncertainty.

i) **Certainty Factors**: A certainty factor (CF) is a numerical measure of the confidence held in the validity of a fact or rule. It allows the inferencing program to work with inexact information. A variety of certainty factor scales can be used, but the most common is a scale from 0 to 1, where 0 indicates a total lack of confidence and 1 represents complete confidence. Expert systems may use such scales as 0 to 10 or 0 to 100. Other arbitrary arrangements may be set up by the programmer. The following rule makes use of certainty factors:

IF	the regulator output is zero
AND	the regulator input is correct
THEN	the regulator circuit is defective (0.9)

This rule states that we are pretty sure that if the input is good but the output is zero, then the problem is in the regulator. But we don't give the conclusion a 1.0 confidence rating because there could conceivably be a less common problem, say a broken wire or defective connector.

While the programmer sets up the CF scale, the expert actually puts the correct value on the rule. Only the expert knows just how confident he or she is in the rule's outcome because certainty factors are nothing more than intelligent guesses based upon experience and available statistical data. The expert sets the certainty factors when constructing the knowledge-base, but they may have to be changed when the system's validity is tested. The expert system usually reaches a conclusion based on several rules in a chain. If each rule of conclusion has a CF, the outcome will have a composite CF.

ii) **Probability**: It is important to understand that a certainty factor is not the same as a probability. A CF is simply a number on an arbitrary scale that states to what extent the knowledge is believed to be true. Probability by contrast, is a number that indicates the chance of an action occurring or not occurring. Depending upon the type of knowledge involved, probability may be a more suitable way to deal with uncertainty than certainty factors, particularly if there are formal historical records of faults, causes, effects, symptoms, and so on. However, probabilities are more difficult to implement.
One popular kind of probability calculation is **Bayes' Theorem or rule**. This Bayesian probability is a formula that computes the probability of event X occurring if event Y has already occurred. Stringing such calculations along in a big system where a lot of rules are evaluated to reach a conclusion causes a lot of computing to

take place, slowing down the process. Despite its complexity, Bayesian probabilities are used in many of the commercially available expert systems.

b) Textural Measures of Uncertainty - Fuzzy Logic

One method of handling imprecise knowledge is a mathematical system called 'fuzzy logic'. An expert, when creating the knowledge-base, may wish to use imprecise terms such as likely or not likely, or expensive and inexpensive. The expert assigns a value between 0 and 1 to such a quantity, indicating the degree of possibility that it is within a given range. For example, if the likelihoods of a department failing to meet budget targets are:

Certain --- Likely --- Unsure --- Unlikely --- **Definitely** Not

and events are, True, Maybe or False, then a possible rule might be:

IF	High Capital Requests is True
AND	Date is March
AND	Budget Review is Unlikely
AND	Term End is October
THEN	Meet Budget Target is Unlikely

Fuzzy reasoning is helpful in dealing with imprecise information. It accepts that often, there are no clear divisions between categories of problem and that experts' responses may differ. Therefore fuzzy logic creates broad categories which leave choice open to the discretion of the expert. Fuzzy logic is also useful because it makes it possible to assign a numerical value to what may appear to the user or expert as a qualitative decision. However, fuzzy logic suffers from the problem of misinterpretation, especially at the user level. Perceptions of what constitutes 'trues' and 'maybe' will inevitably vary between people and this variation may be cumulatively substantial over a set of rules.

3.2.1.2 Blackboards

The example rules for troubleshooting faults on the VAX computer mainframe in Section 3.2.1. are from a single domain or area of knowledge. When diverse types of knowledge have to be handled, the rules are sometimes grouped into specialised independent sets, each corresponding to one type of knowledge. These so-called **Knowledge sources** all operate on a common central database, the **Blackboard,** and communicate their results to one another via this blackboard as shown in **Figure 1B.** For example, we may incorporate a system of evaluation which has other knowledge sources, such as computer system configuration, capacity planning, new and spare parts adviser, mainframe tuner, software development, training and programming aid and so on.

The blackboard is usually large and complex and requires powerful hardware capable of **parallel processing**, that is, performing several tasks at the same time. At present, there are a limited number of machines which can accommodate a blackboard architecture.

3.2.1.3 Rule Induction by Example

One way to impart knowledge to an expert system is to state a number of examples or case histories about the domain. This can be done by listing the conclusions, outcomes, or answers that the expert system is expected to give. Then for each of these, attributes or conditions are chosen which result in a specific outcome. the intention is to gather as many examples as possible from the domain of interest.

The examples are entered into a matrix similar to that of a spreadsheet. Each column in the matrix represents the decisions, outcomes, or results that derive from the various combinations of attributes. Once all possible examples are listed, an algorithm inside the induction system generates rules from the matrix. In other words, rules are induced from the data in the table. The resulting rules are either similar in structure to standard IF-THEN rules or are presented in decision-tree format from which rules are readily derived. A rule-base is created using this process. From this point on, the standard inference engine uses the rule base to draw its conclusions during consultation.

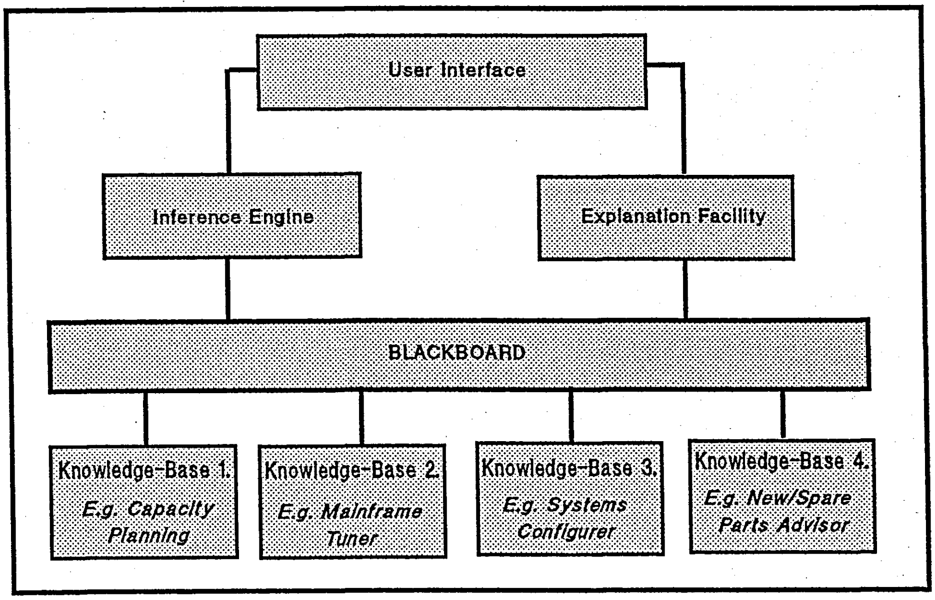


Figure 1B: Diagrammatic Representation of A Blackboard Architecture

Figure 1C. shows hypothetical examples entered into a matrix. A typical rule derived from E.G. 4 in this table would be:

IF the AC input is OK
AND the secondary voltage is zero
AND the filter output is zero
AND the regulator output is zero
THEN the problem is a bad fuse

The main benefit of rule-generation by induction is its ease of use. Creating the rules from the collected knowledge is sometimes difficult, but creating the matrix of attributes and outcomes is quite simple. It is possible for the expert to be able to create the matrix, thus minimising the complex and time-consuming development process. A second benefit is that it is possible to use data held in existing databases such as dBaseIV and even Lotus 123, so that rules may be generated automatically once attributes and conclusions have been defined.

The transparency of both rule-based and rule induction techniques has made it the chosen representation scheme for many expert systems, especially in situations where the domain expertise is founded on empirical observation of past associations. However, these representations tend to be shallow, in other words unable to describe adequately the fundamental principles in a problem area. In contrast, representation schemes using frames or semantic networks allow a deeper insight into underlying concepts and causal relationships and facilitate the implementation of deeper-level reasoning such as abstraction and analogy. It is these that we now turn.

Figure 1C: A Matrix of Examples from which Rules are Derived through Induction

AC VOLTAGE	SECONDARY VOLTAGE	REGULATOR INPUT	REGULATOR OUTPUT	SUSPECTED CAUSE
EG.1. O.K.	O.K.	NORMAL	ZERO	DEFECTIVE REGULATOR
EG.2. O.K.	O.K.	LOW	LOW	DEFECTIVE CAPACITOR
EG.3. O.K.	ZERO	ZERO	ZERO	DEFECTIVE RECTIFIER
EG.4. O.K.	O.K.	ZERO	ZERO	FUSE BLOWN

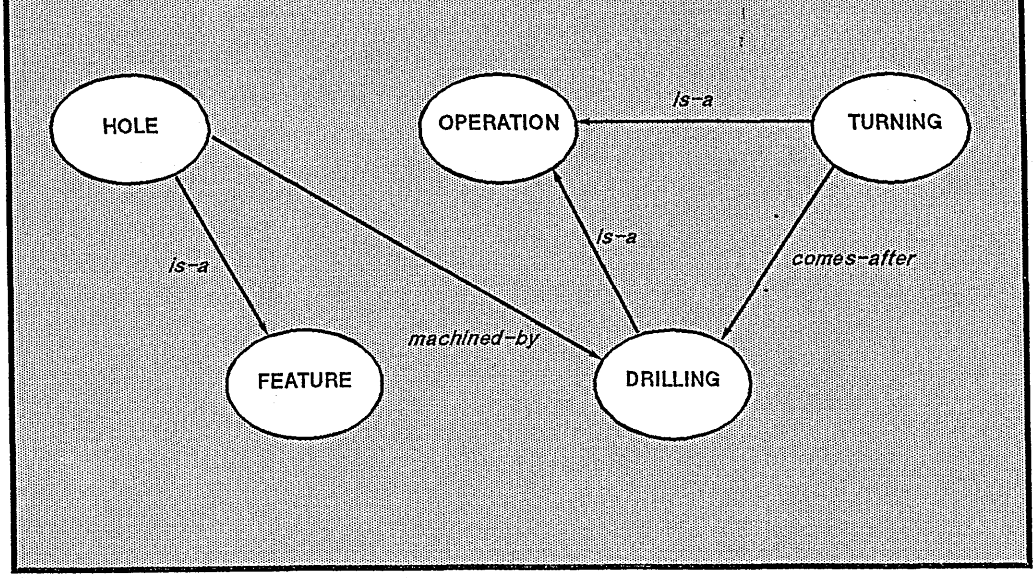
3.2.2 Semantic Network Based Representations

One of the simplest and most effective ways to represent certain types of knowledge is to use a semantic network or semantic net. A semantic net is a graphical representation of knowledge that shows the relations between objects. Semantic networks are excellent for representing declarative knowledge, particularly that which has a hierarchical structure. When the knowledge can be classified or categorised, it is a good candidate for a semantic net. An example of a semantic network is shown in **Figure 1D.** The circles are called ‘nodes’ and are used to represent people, places, things or ideas. The nodes are connected to one another to show relationships. These links between nodes are called arcs’. On each arc is a label that states the relationship between the nodes that it connects. while semantic networks are a useful visual took, they can also be programmed into a computer to form a complete knowledge-base.

Figure 1D. states the fact that the hole is a feature which is manufactured by drilling, which is an operation, which comes after turning. It can be seen that most of the nodes represent an object, but that other nodes represent attributes of the related object such as machine process, sequence or specification. An important characteristic of a semantic network is that some nodes may inherit properties or characteristics from other nodes. Since semantic nets are used to represent hierarchical information, some nodes will be higher in the hierarchy than others. Nodes that are lower in the hierarchy can inherit propèrties from the nodes higher in the network. This characteristic of a semantic network eliminates the need to repeat information at each node.

To solve problems with a semantic network, you ask questions about the domain being represented. The inferencing program searches through the various arcs and nodes looking for the key words in the question. If the knowledge is built into the system, it would be able to provide specific answers. For example, the question ‘How do we machine a hole?’ would be answered "drilling’. ‘After Turning’ would also be correct through **inheritance.**

Figure 1D: An Example of a Semantic Net Representation



3.2.3 Frames

The frame is a knowledge representation scheme that is designed to handle declarative knowledge. For example, a frame can be used to describe any object in detail. The frame is divided into discrete elements called **slots.** The slots contain the attributes of the object being described. In many ways, this knowledge structure is hierarchical and, for this reason, is similar to the semantic network. The primary use for frames is to represent what is referred to as **stereotyped knowledge** and is that which is known or can be expected with some certainty. Frames are useful for packaging well-known or generalised attributes of any person, place, thing, object, event, or idea. **Figure 1E.** shows a frame for the client company's locomotive. It is made up of subdivisions called slots and **facets.** Each slot describes an attribute which may, in turn, contain one or more facets. One facet may be the value of the attribute: another may be a default value that can be used if the slot is empty. A default value for the slot 'Thyristor' in Figure 1E. for example, is 'Oil-Cooled'. An 'if-needed' facet may also be used. If no slot value is given, the 'if-needed' facet, also called a **procedural attachment**, triggers a procedure that goes out and retrieves or computes a value. Thus, the cant deficiency slot in Figure 1E for instance, is filled by a procedural attachment that will go out and run a benchmark to obtain a numerical value that expresses cant deficiency in a way that it can be compared to similar locomotives. Frames may be interconnected like semantic networks to form a highly detailed knowledge-base. A particular slot in a frame may reference another frame that contains detailed information about that particular attribute. A slot in that second frame then could reference another and so on. When frames are linked in such a hierarchy, one frame may inherit the properties of a higher-level frame. This makes knowledge storage more compact and permits in-depth reasoning. Inferencing is carried out through a detailed search of the slots and frames.

The choice of knowledge representation technique can be seen to be a complex decision, but a useful rule-of-thumb is that if the knowledge is a poorly structure collection of isolated facts (i.e. shallow knowledge), then rules are the most appropriate. Conversely, if the knowledge is made up of patterns, hierarchies or relationships (i.e. deep knowledge) then frames, semantic networks and object orientation respectively are necessary. This section has described how knowledge is represented; but an equally important issue is how the inference engine arrives at the relevant knowledge required to solve a specified problem. As the following section shows, there are different techniques according to the type of problem.

Figure 1E: An Example Frame

Frame Name : Locomotive Engine	
Slots	**Facets**
Engine Type:	Diesel
Model :	91
Power Output	4700 kW.
Thyristor Type **(Default)**	Oil Cooled
Cant Deficiency	**Procedural Attachment**
Maximum Speed	240 Km./h.

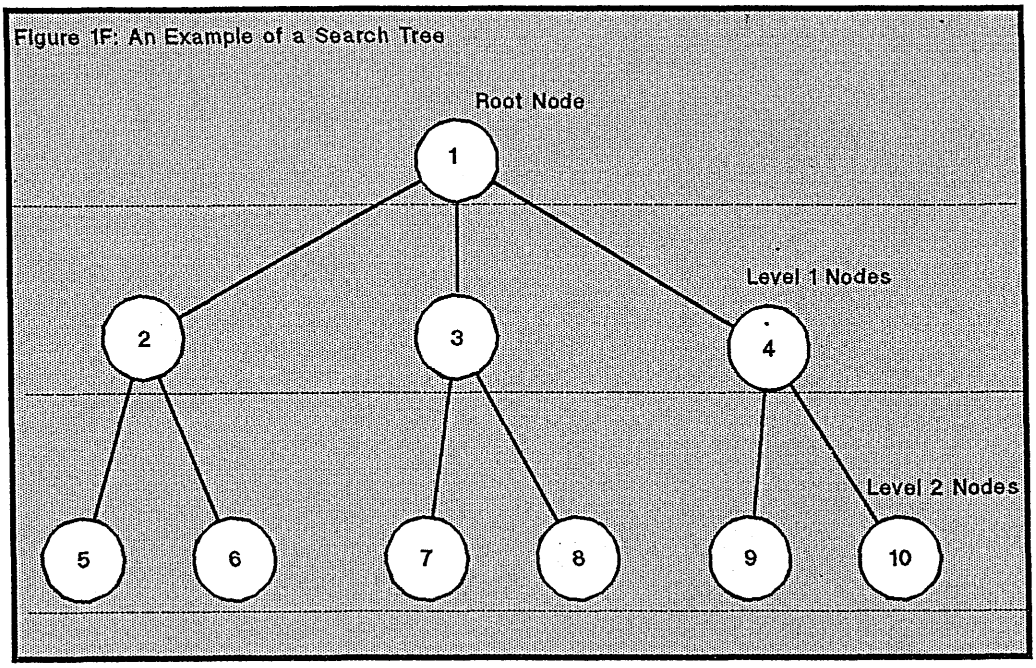
4 Problem Solving Techniques in Expert Systems

The basic problem solving method used in expert systems is **search.** Search is the process of examining a large set of possible solutions to a problem in an attempt to find the best solution. This is a trial and error method of looking through the knowledge-base attempting to match knowledge items to known facts. The knowledge-base, referred to as the **search space** comprises all final solutions to the problem and any intermediate solutions. In a workable expert system, the search space is usually a set of IF-THEN rules, and it might also be the nodes and arcs of a semantic network or a collection of frames.

The basis of most search techniques, or **inferencing control strategies** to use the jargon, is the **search tree.** A search tree, often called a decision tree, is a graphical method of representing the search space. In order to visualise the elements of knowledge in a knowledge-base, a picture can be drawn using nodes and arcs like those in semantic networks. Each node represents a single fact, rule or another knowledge element. The nodes are interconnected with arcs showing the relationships. A typical search tree is shown in **Figure 1F** overleaf. The basic structure of a search tree is hierarchical, and it is similar to an organisational chart, but with numbered levels of nodes. They are useful in describing how an inference engine searches through a knowledge-base.

The search technique controls the process by which the inference engine interrogates the knowledge-base. To make expert systems useful and practical on a computer, the search technique must be quick and effective. The most basic search technique is **blind search.** This is a crude method of searching through every node in the search tree seeking a solution. Blind search examines the entire search tree in an orderly manner in an effort to draw a conclusion. The blind search starts with the root node and them systematically works its way downwards in the search tree from left to right. While this approach is guaranteed to produce a solution, its disadvantage is that it is a slow and tedious process. At each node, the computer must do some pattern-matching, where rules, frames or portions of them are compared to a data string representing a fact to be verified. When a large tree is involved, the search for a solution may take an unacceptably long time, even hours. This is because the greater the number of nodes and the more branches per node, the more quickly the tree expands.

Figure 1F: An Example of a Search Tree



This expansion is known as a **combinatorial explosion.** A tree can become so large as to make blind search an unacceptable approach to problem solving. The solution therefore is to limit the searching area by focusing upon one part of the tree which is most relevant to the problem domain. There are three basic techniques which go about search in this way. These will be discussed briefly:

4.1. Heuristic Search This technique is used to help eliminate the possibility that the search will go off into some deep network of branches where there may be no possibility of a solution. It does this by setting limits on the depth of search until a reasonably likely branch of knowledge has been found which may solve the problem. A second method is to use **metarules** which state ways that the knowledge rules can be used. This approach creates a small knowledge-base about how to guide search processes in the most efficient way.

4.2. Forward Search This technique, also known as forward reasoning or forward chaining, begins with the top or root node and searches downward in the tree until a goal or solution node is found. Forward search begins by the inference engine searching for any available facts in the database and looks for those facts in the IF portions of the rules. If the IF part of the rule matches a fact in the database, the rule is fired. The THEN portion of the rule is said to be true and a new fact is inferred from this and stored in the database. With this new information, the inference engine moves forward to find this newly inferred fact in the IF part of another rule. This process continues until no further conclusions can be reached at which point the system gives an answer. Forward search is said to be **data driven.**

4.3. Backward Search This is the most widely used search technique. Also know as backward reasoning or backward chaining, the inference engine attempts to prove one of the conclusions, or hypotheses that it already knows. These conclusions are tested one after another based on given information. With sufficient input information, one of the hypotheses will be proven. In a backward search, the inference engine looks at the THEN part of the rule first and then attempts to prove the IF portion. It looks in its database for rules that conclude that portion of the IF statement. If none of those are possible, it asks the user to supply the necessary input. Should the necessary input not be available, the inference engine looks for other rules, or concludes that the particular hypothesis under test cannot be proven at all. It then moves on to the next hypothesis and each additional hypothesis in sequence until one is proven. The search may backtrack as required to find the proof for the selected goal. Backward search is said to be **goal driven.**

Forward Chaining	**Backward Chaining**
IF there is no power	THEN computer will be down (*Hypothesis*)
THEN the computer will be down	IF there is no power

5. Types of Expert Systems

This section looks at the range of expert systems in terms of software and hardware requirements and configuration with other systems, and also the way in which each are particularly suited to certain applications.

5.1 Expert Systems Software

Expert systems is software and as such, a number of the conventional software development techniques may be used to create them. Furthermore, they can be built using conventional programming languages. There are special packages however that greatly facilitate the creation of expert systems through both increased productivity and increased capabilities. These range from specific Artificial Intelligence (A.I.) languages which are highly complex and require specialised hardware and substantial programming skills, to 'shells' which require no programming knowledge and expedite the rapid production of small but effective tools.

The hierarchy of expert system software tools is shown schematically in **Figure 1G.** This is drawn deliberately as a triangle to emphasise that breadth and flexibility at the language level, narrowing progressively up to the application specific or 'off-the-shelf' type tools. Taking each in turn:

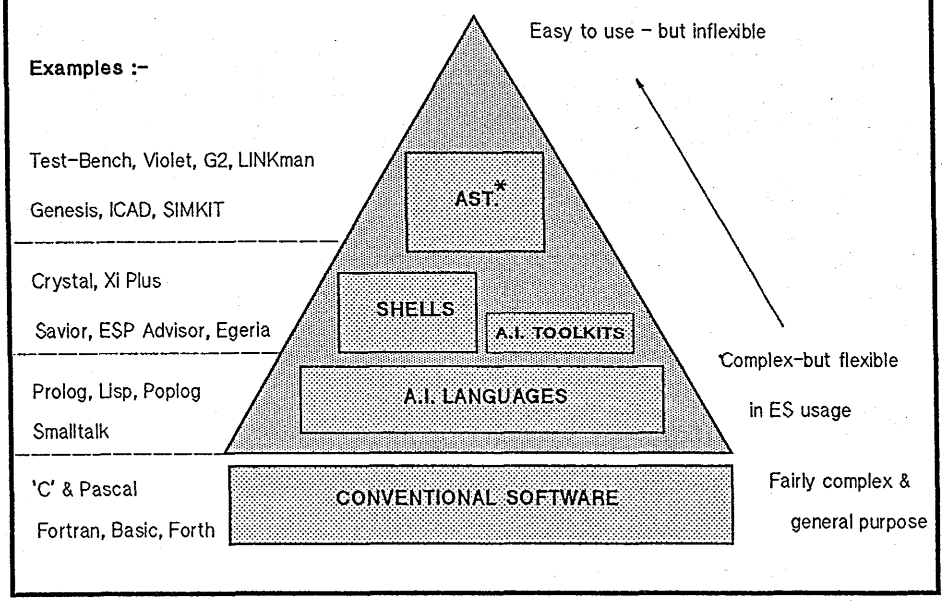
5.1.1 Conventional Programming Languages: Expert systems have been created using almost every major programming language including FORTRAN, BASIC Pascal, 'C', and assembly language. However, the trend for most A.I. programming is to use 'C', which is extremely flexible and produces fast code. If you are a competent programmer, experienced in one of these languages, it may be of benefit to create expert systems using that language because learning new A.I. languages, though more suited to expert system developments, can be time consuming and difficult.

When programming in a conventional language, every element of the expert system must be created from scratch. A format for the knowledge-base must be devised. The control strategies for the inference engine must be determined and programmed. The database, user interface, and other subprograms must be created. Even for a skilled programmer, familiar with a particular language, this represents a challenging project. For this reason, higher level tools with improved expert system capabilities and facilities have been developed.

5.1.2 A.I. Programming Languages: A number of programming languages have been created to deal specifically with expert system applications. The most popular

A.I. languages are LISP, Prolog and Smalltalk. With these, the software development process is eased, in Prolog more so than LISP or Smalltalk, and expert systems can be created more readily than with conventional programming languages. However, A.I. languages are still only suitable for trained software engineers and computer programmers. In general these languages, which have many hundreds of derivatives, contain no knowledge representation capabilities formalised as tools in the software, or inference mechanisms. The high skill levels required to use them effectively do not make this a viable starting point for problem solving for many applications. In addition to the skill required being in short supply, the amount of investment required in terms of effort and timescale means that this starting level is generally only used for bespoke problem solving where there is a considerable emphasis on run-time performance or a requirement for large scale distribution of the finished systems.

Figure 1G: A Hierarchy of Expert System Software Tools



5.1.3 A.I. Toolkits: A.I. languages have been used to provide a number of higher level tools for A.I. programmers to use (e.g. KEE, ART, etc). These toolkits (or 'environments') provide sophisticated knowledge representation techniques and interface mechanisms. Using these tools, complex expert systems can be built without having to design and code the knowledge representation inference mechanisms. However, the skill and experience levels required to master the use of these tools is considerable: three to six months of in-depth training is often quoted as a minimum requirement, even for skilled developers or **knowledge engineers.** Although extremely powerful, such toolkits have no task or domain knowledge embedded in them, and are in that sense still general purpose tools.

5.1.4 Expert System Shells: Designed for use by non-expert system specialists, shells such as Crystal, Xi Plus, and Adviser, provide simpler and less varied knowledge representation and inference mechanisms than the A.I. toolkits. An expert system shell is a collection of programs that enables you to create an expert system without using a programming language. Because no programming expertise is required to use a shell, these development tools have been responsible for fostering the development and use of expert systems. A shell is a completely implemented expert system without a knowledge-base. The shell contains an inference engine, user interface, and an explanation facility, and it usually has a convenient form for entering the knowledge-base. Shells can operate on personal computers, minicomputers and mainframes. They can also operate on dedicated hardware (which will be described later).

There are two basic types of shell available corresponding to the two types of rules; production- and induction-based. Hybrid systems are also available. Some rule-based shells also permit the creation of frames or semantic networks. Such shells can greatly simplify and speed development of expert systems. Given that no complex programming is required, they may lower costs as well since the skills required to become familiar with such shells can usually be quickly obtained. However, being simple tools, the scope of problems that can be tackled is necessarily somewhat limited: in practice, expert system shells have been used mainly to construct off-line consultative and advisory systems.

5.1.5 Application Specific Tools (AST): A.I. tools targeted towards specific types of applications or classes of application are just emerging as commercial products. Generally, these tools have been designed for use by the end user to solve particular types of problem, often in a specific domain. Such systems have carefully designed user interfaces, serving two purposes: firstly, the user is presented with a system that is easy to use and relevant to the application at hand, and secondly, templates are provided so that the users can express their knowledge by specification ("filling in the blanks" for example). This helps to acquire and structure the knowledge, without directly exposing the user to A.I. languages . A selection of some of the currently available knowledge-based AST's relevant to engineering applications is given in Table 2.

Task	**Tool**	**Supplier**	**Function**
DIAGNOSIS Tools	TESTBENCH	Carnegie Group	Manufacturing Diagnostic
CONDITION MONITORING	VIOLET	Intelligent Applications	Vibration based machine health monitoring
PROCESS M'MENT	G2	Gensym Corp.	On-line real-time advisory system
PROCESS CONTROL	LINKMAN	Image Automation Limited	Rule based real-time closed-loop process
PLANNING & SCHEDULING	GENSIS	Sira Ltd.	Production Scheduling
DESIGN	ICAD	ICAD Inc.	Knowlege-based CAD system
SIMULATION	SIMKIT	Intellicorp	simulation system

Table 2: *Examples of Application Specific Tools in Manufacturing*

In general, not all types of application can be abstracted to the point where problem solving frameworks of this kind can be devised and application specific tools produced. Such tools do exist as commercial products, such as the above, but they are relatively new (less than 3 years old) and have yet to be fully proven in use.

5.2. Expert System Hardware Configurations

Each of the above software tools have specific hardware requirements; for example, many of the A.I. Toolkit cannot operate on personal computers , whilst to use anything else for most shells would be a waste of computing resources. Similarly, hardware has specific configuration requirements depending on the way that the expert system is set up and used. These hardware configurations include stand-alone, hybrid or embedded, linked, dedicated, and real-time systems and these are discussed below:-

5.2.1. Stand-alone: A stand-alone expert system is one that runs by itself and fully occupies its host computer. The program is loaded from a floppy disc or transferred from a hard disc into the computer memory and then executed. Most of the expert systems used are of this type.

5.2.2. Hybrid Programs: Expert systems can also be embedded or integrated with conventional programs. Either the expert system is the main program with a number of algorithmic subroutines (to undertake mathematical functions etc.) or the expert system is a subunit of a conventional program (the knowledge-base would then be consulted when uncertain information is used or decisions are made).

5.2.3. Linked Software: Many expert systems are set up with links to external software packages such as spreadsheets or database management systems. In this way, the expert system can tap the data stored in them. Most of the good shells have built-in **hooks** to standard software packages that can provide this information. It is also possible to link multiple expert systems (small P.C. based expert system feeding to a larger mainframe expert system using the blackboard principle, or interactive terminals using a large mainframe expert system).

5.2.4. Dedicated Systems: Another type of expert system is one that is buried within a closed or dedicated computer. Like a stand-alone system, it uses one computer to solve one problem. A factory process control system is an example of a dedicated system. Microcomputers are also built into appliances, test instruments, and other equipment. These computers perform no function other than that for which they were intended. Many are hard-wired into the application and cannot be changed or accessed. Nearly all A.I. Toolkits require dedicated systems.

5.2.5. Real-time Systems: Real-time software programs are designed to respond quickly to inputs and to perform the necessary processing almost immediately. An example of a real-time system is process control and monitoring of automated machinery. Sensors provide the expert system with information on wear of tools, temperature, speed, and so on and decisions are made to determine how conditions should be modified by providing output controls and signals.

There are no hard and fast divisions between configurations, and very often an expert system could embody the features of two or more of the above types. Embedded and linked systems may be combined. A dedicated system may also feature real-time operations.

The choice in selection of software, hardware and configuration depends upon the problem characteristics of the application. As the following section shows, it is possible to group applications into one of seven main categories.

6. Generic Applications of Expert Systems

Expert systems have many potential applications, each require different hardware and software and must be developed in different ways. However, there are general classes of application which are created according to the ways in which they are used, and these will be discussed briefly giving examples for each class.

6.1 Analysing and Interpreting Systems: Expert systems are very good at analysing a large amount of information, interpreting it, and providing an output explanation or recommendation. When large amounts of information are involved in a problem, it is difficult for a human to remember and keep track of it all. An expert system however, could quite easily deal with this and consequently provide a more thorough analysis and interpretation than a human, often providing a better recommendation or more thorough understanding of the situation. Expert systems that perform interpretation, work from input data supplied from a keyboard, or by another computer program, or derived from electronic monitoring sensors. Once the information is available to the system, the inference program uses the input data along with a knowledge-base in an attempt to understand the data. It then provides an explanation or draws conclusions from this data. An example of an analysis/interpretation system is one that allows the intelligent interrogation of databases and historical records to provide recommendations and advice.

6.2 Predicting Systems: Expert systems are good at predicting results. By using input data about a given situation, a prediction expert system can infer future consequences or outcomes based on knowledge it has. Prediction systems are good at determining likely consequences of given conditions. The knowledge-base often contains trend data and historical information as well as cyclical patterns that are applicable. By applying these to the input data, likely outcomes can be reliably predicted. Examples might be in forecasting, demand analysis, shop floor loading, trend and impact analysis. A specific example might be .estimating the cost of manufacturing a particular motor, say during contract negotiations, where a ‘reasonable estimate’ of price can be quickly established.

6.3 Diagnosing and Debugging Systems: Another excellent application for expert systems is diagnosing and debugging. Both of these techniques are used in the troubleshooting and repair of equipment. A diagnostic expert system is given input data about the behaviour of the device, system, or individual. The system asks a series of questions in an effort to accumulate the input necessary to draw a conclusion. This input data takes the form of symptoms, physical characteristics, recorded performance, and any irregularities or undesired functions. With this information, the inference program scans the knowledge-base to determine what is wrong. While some expert systems perform only diagnosis, others also include debugging characteristics, which means they recommend suitable actions to correct their problems and deficiencies discovered. Fault diagnosis and debugging is by far the most common application of expert systems with examples from Printed Circuit Board testing, welding analysis and quality control to design test simulation.

6.4 Monitoring and Control Systems: Monitoring and control is basically the process of observing and acting upon inputs derived from sensors. A wide variety of sensors can be used to convert physical changes into electrical signals that can be used by the expert system. Monitoring expert systems look for observed behaviour that confirms their expectations about the condition of a state or their assumptions about possible deviant action. On the basis of this information, expert systems can prescribe new actions, remedy problems and simulate possible outcomes.

Monitoring and control applications can make sense of on-line data and enable quick decisions to be made which otherwise may have required detailed analysis. For example, using sensors connected to moving parts on motors under test, vibrational data is provided which when interpreted by an expert system allows for the diagnosis of faults and possible future failures before they arise.

6.5 Design Systems: Design is the process of creating a product, device, or procedure. Given a set of specifications, requirements, and constraints, the designer creates or develops the desired object or procedure. Design requires a large amount of knowledge, much of which can be contained within an expert system to assist a designer in creating their product or process. While design is essentially a creative process, much of it entails the application of standard rules and procedures. An expert system may contribute in this part of the design process.

6.6 Planning Systems: When a goal is acknowledged, all of the actions required to achieve it can be identified. Then these actions can be properly sequenced to achieve the desired goal. Planning is the process of putting steps in the sequence. The complexities of sequencing resources, information and actions are clearly evident during the planning of large contracts. Expert systems can help planners by providing and sorting that knowledge which is required from large amounts of input data and highlighting the constraints and requirements which need to be considered.

A number of commercial planning expert systems have been developed. IMACS is used to help plan manufacturing capacity and manage inventory. ISIS helps plan factory job schedules. PTRAN helps create a plan for manufacturing custom configurations of complex computer systems. While these systems are all used in manufacturing, the principles could be applied to any areas of the business such as management and strategic planning.

6.7 Instructional Systems: Many computer based training (CBT) or computer assisted instruction (CAI) programs are available to teach specific subjects. But such programs have a fixed content and sequence that is not best for every trainee's learning style. Expert systems can solve this problem by continually evaluating the trainee's level of knowledge and understanding. With this information, it can adjust the instructional process to the trainees' needs. Such systems can perform tuition and training functions; for example, as a help system for main-frame end-users or in updating engineers about new processes and techniques. In these areas, there is a long learning curve before the trainee can be considered good enough to be useful: the use of instructional expert systems can help to accelerate the learning process and also free the trained or human expert to perform other tasks.

7. Bibliography

Frenzel, Louis (1987): Understanding Expert Systems, Howard W. Sams & Co. (*first edition*), USA.

Daly, Donald (1988): Expert Systems Introduced, Chartwell Bratt, 1988 UK.

Bryant, Nigel (1988): Managing Expert Systems, John Wiley & Sons, UK.

Taunton, Chris & Ready, Colin (1988): 'KBS Technology in Engineering ' in proceedings of Software Tools Conference, June , London, 1988; Online Publications, Pinner UK (p208)

Appendix I: Section Two

A Glossary of Expert System Terms

Algorithm: a step-by-step procedure for solving a problem. A precisely defined group of rules or processes that leads to a desired output from a given set of inputs.

Artificial Intelligence: hardware and software techniques that make it appear as though a computer is thinking, reasoning, making decisions, storing or retrieving knowledge, solving problems, and learning.

Backward Chaining: a method of reasoning that starts with the desired goal and works backward, looking for facts and rules that support the desired outcome. A technique used in tree searches where a conclusion is hypothesized and the system works backward to find rules that support the hypothesis. Also know as goal driven reasoning.

Blackboard: a method of organising, presenting and communicating information. A central data structure for coordinating several knowledge sources.

Blind search: a general category of search technique that makes use of no knowledge or heuristics to help accelerate or simplify the search process. It is a time consuming and arbitrary search process that attempts to exhaust all possibilities in searching rather than rely upon information that can help narrow the search.

Certainty Factor: a number assigned to a fact, action, or relationship that indicates how likely it is to be true or to happen. A certainty factor of 1.0 means 100% true, 0.5 partially true and 0.0 not true.

Confidence Factor: a number or system of numbers indicating the certainty held in a specific fact, statement, or piece of evidence.

Data Driven: a kind of inference used in tree searches. Data-directed reasoning is forward chaining.

Decision Tree: a graphical structure of nodes and arcs that shows alternative paths for various decisions or outcomes.

Domain: a field of knowledge or expertise. A problem area of interest to expert system technology.

Embedded system: Expert system software built into or buried in and referenced by conventional software.

Expert systems: the most commercial subset of artificial intelligence. A program consisting of a knowledge-base, an inference engine or reasoning system, and a user interface. The expert system embodies all the facts, information, knowledge, and heuristics in a specific domain.

Forward chaining: a problem solving technique used in production and rule-based systems in which conclusions are drawn or decisions are made by starting from the known facts. A search procedure or reasoning process using known facts to produce new facts and to reach a final conclusion. Also know as data-driven reasoning.

Frame: an outline or an hierarchical structure containing slots for listing relevant facts and attributes.

Fuzzy reasoning: a method of determining an adequate solution from imprecise information.

Heuristic: the use of practical knowledge to assist in problem solving. Heuristics include rules-of-thumb, tricks, procedural tips and other information that help to guide, limit, and speed up the search process.

IF-THEN: a decision-making test that initiates an action if a specific condition is met.

Induction Shell: a shell which allows an expert system to be built by entering knowledge as examples in a matrix. The shell induces rules that are used in reasoning.

Inference Engine: that part of an expert system that actually performs the reasoning function.

Inheritance: the process by which one object takes on or is assigned the characteristics of another object higher up in a hierarchy.

Knowledge-base: a collection of data, rules, inferences, and procedures organised into frames, blackboards, semantic networks, scripts, rules and other formats.

Knowledge Engineer: the person who designs and builds the expert system.

Knowledge Representation: the process of symbolically structuring, encoding and storing knowledge.

LISP: a widely used A.I. programming language.

Meta rule: a rule that describes how rules should be used.

Nodes: places, goals or subgoals in a search tree.

Parallel processing: computers which allow the simultaneous processing of more than one program.

Pattern Matching: the automatic recognition or identification of figures, characters, shapes, and forms according to predetermined conditions and standards.

Probability: a number indicating the likelihood of the occurrence of a specific event. Often expressed as a percentage, it is used in production systems to deal with uncertainty.

Problem-solving method: a scheme for organising reasoning steps and domain knowledge in constructing a solution.

Prolog: a widely used A.I. language.

Real-time computing: processing that occurs fast enough to keep up with other actions or operations.

Rule-based: any program that uses a set of rules to draw conclusions, make decisions and solve problems.

Run Time System: a piece of software that enables a user to run a program created with a software package but not to develop or modify programs.

Semantic Network: a method of knowledge representation using a graph comprising nodes and arcs where the nodes represent objects, situations, concepts, or entities, and the arc represents links describing the relationship between the nodes.

Shell: an expert system generator. A software package that allows you to create an expert system without programming.
Symbolic processing: using symbols rather than numbers to represent and manipulate data, information and knowledge, in order to reason and understand.

User Interface: that portion of the expert system that communicates with the operator. Inputs are accepted from which outputs are generated with such techniques as natural language and menus.

Appendix II

An Account of the Structure and Operations of the Client Company

Appendix II

An Account of the Structure and Operations of the Client Company

Chapters 3 & 4 argue that a first stage of technology assessment is an analysis of the organisation rather than the technology. The author spent two years in the company at both sites and, through personal experiences and the use of mechanisms such as IDEFo, acquired a thorough, though not complete, understanding of the company's operations and culture. It is not the purpose here however to divulge details of the company's practices other than to provide a backcloth against which the context of expert systems assessment and development may be understood.

The Client Organisation is a subsidiary of a large corporate manufacturing-to-electronics company based in the UK, and its 'line of business' is in the design and manufacture of control gear and motor equipment for locomotive engines. The company specialises in one-off tenders and its customer base is world-wide with an important aspect of the business being in the service and repair of previous product designs and contracts. In all there are approximately 1300 employees in the CO.

One of the most significant features of the Client Organisation (CO) is the split site mode of operations. At a Manchester site (referred to as Site2 in the main text) is the executive headquarters, and it accommodates the main administrative, commercial, computing functions. There is also a large engineering operation where the design of electronic control systems, and electrical equipment assemblies for electric motive power units, is undertaken. This site also has a relatively small manufacturing operation concentrating on Printed Circuit Board assembly and light electro-mechanical equipment manufacture and assembly. On the same site is a sister company which is responsible for the project management of complete contracts and turn-key developments (Site1). As well as sharing facilities, the CO interfaces directly with this company with most contracts.

At a second site in Preston (Site3), is a large manufacturing operation which concentrates on the manufacture of heavier electrical machinery, as well as the fabrication and assembly of equipment cases, and the final assembly and testing of completed projects. There is also a smaller engineering function which focuses upon the design and development of machines.

The company's 'product' is essentially custom built to contract and necessarily requires a substantial amount of engineering work. In fact, a major strength of the CO, as a distinguishing criterion of competitive advantage, is its quality in design. Such is the variation of requirements across the global customer base that advantages of a standard product range are not attainable. However, efforts have been made to standardise manufacturing techniques, materials and components in a move towards adopting flexible specialisation and group technology techniques. This is especially important for the company since the structure of the market has changed from being based on single, large volume contracts often with repeat orders, to many low volume and one-off contracts requiring significant increases in the number of new designs and a consequent increase in the engineering content.

A related critical success factor to the CO is lead time, i.e. duration from acceptance of a tender contract through to delivery of the final product. Current lead times are presently around 17-18 months with engineering and design occupying up to 50% of this. Since more than one project is often carried out at a time, the company has adopted a mode of working where there is a substantial overlap between activities which previously had been carried out in series. The management and control of changes in design are therefore critical to the company's lead time performance.

Since 1983, the CO has undergone significant technical changes spearheaded by a Computer Integrated Manufacturing initiative. This has improved the company's capabilities to produce faster tender diagrams and drawings; provided a tighter control of standards, raw materials, components and equipment; made it possible to model assemblies before anything is built; and provided the ability to closely control the release and revision of drawings and manufacturing information. It has also improved communications between sites and thereby lessened the geographical restrictions imposed by the split site mode of operations. Despite these developments, the CO may be described as a 'medium' technology company, with examples of 'high-tech' or leading-edge information and design and manufacturing systems and associated working practices; but also low-technology activities which adopt very traditional manufacturing methods and practices, and moreover, retain a large body of semi- and unskilled workers for this purpose.

A characteristic of the company's organisational structure is that it is contract based and therefore functions are orientated and costed about specific projects. Resting upon this functional grouping is a support infrastructure made up of centralised service departments such as computing and finance. Developmental work is also contract based and driven by the needs of a particular project. There is little research and development work undertaken beyond this level.

Appendix III

Methods of Expert Systems Assessment and Development: A Review of Viable Technical and Organisational Tools.

Contents:-

Appendix III

Methods of Expert Systems Assessment & Development: A Review of Viable Technical & Organisational 'Tools'.

This appendix looks at a selection of current ES development tools and describes possible enhancements to them using approaches, methods and tools derived from other disciplines (in the case of technical refinements) and concepts (in the case of organisational enhancements).

A. Technical Enhancements to Assessment & Development

Chapter 4 has described two schools of thought in the development of expert systems. The first is based on the belief that in order to be commercially viable ES should adopt traditional software engineering methods of development based on a Specify-Prove-Implement-Verify or 'SPIV' model. By contrast, a second school believes that by virtue of the uniqueness of expert systems, and their specific development needs, the traditional development model is inappropriate and should be superseded by a second model based on Run-Understand-Debug-Edit or 'RUDE' which defines a central role for prototyping. The chapter concluded that both technical models had strengths and weaknesses and should therefore be combined to produce a 'hybrid' approach. In each of the three scenarios, methodologies were identified and these are described below.

A1. Methodologies based on the 'SPIV' Model

Two methods were identified in this category from the literature: 'KADS' which is a highly structured lifecycle methodology and emerged from a European research project (Esprit (1987) as a commercial 'product'; and the Structured Specification approach defined by Keller (1987).

A1.1. The Knowledge Acquisition Documentation System (KADS) Methodology

The KADS methodology adopts a waterfall model in that detailed analysis precedes any design decision and implementation follows design. The methodology is referred to as a 'science of methods'(Haywood:87) in its formal and structured approach to development. Development itself is viewed as a modelling activity, with different levels of abstraction defined, from data models and design models to conceptual models. Furthermore, the differences between SE and KBS development is expressed in terms of the latter requiring more complex design models rather than the two being conceptually or methodologically different.

These models are combined so as to provide a description of the knowledge engineering process, independent of implementation issues at this stage. This is the equivalent of a logical design in conventional systems development from which the physical design is developed and constrained by external requirements.

The logical design in KADS is made up of three elements functional, behavioural and structural. The functional viewpoint describes the actual functional decomposition of the KBS and the relationship between the decomposed functions in terms of data and control. The behavioural element describes how the functions can be represented using KBS techniques: and the structural element defines the logical architecture of the components of the final system. The KADS methodology provides a library of generic problem solving task structures, called interpretation models, which act as templates to guide the acquisition and subsequent analysis of knowledge. In order to achieve a real application, a number of interpretation models are combined to form the overall inference and task structure for the problem. In this way, the development process is based on model refinements and construction.

The strengths of the KADS methodology is in its detailed attention to requirements analysis. It distinguishes between internal and external requirements. The former includes knowledge analysis, analysis of expert and user tasks and construction of conceptual and design models. This explicit representation of knowledge and tasks by models enables the transportation of knowledge and its context of use away from the domain in which it was derived. This stage is supported by a computer based documentation system, a detailed handbook and tools to aid the developer in its use. The external view covers analysis of the current technical layout in the organisation, analysis of organisational objectives and constraints, and determination of functional requirements. The value of identifying organisational constraints is that it defines explicitly the costs, resource and technical constraints of the organisation or project before design begins. Determining functional requirements prior to technical requirements also improves the definition of project objectives, project plans, organisation and metrication (Esprit:1987).

The KADS methodology allows the developer to closely controls and monitors the transformation of elements of the conceptual model into the elements of the design model through constant evaluation and documentation. This accountability, as Hickman points out, makes verification, debugging and enhancements, and maintenance more routine and predictable procedures. The most recent upgrade of KADS (KBSPRIT:89) incorporates detailed project management and quality assurance procedures.

A1.2. The Structured Systems Approach:

One of the main problems in KBS development is in understanding the problem enough to define and elicit the appropriate expertise and provide a representation technique suitable for its coding and implementation in an KBS tool. Keller proposes an approach based on Yourdon's structured systems analysis methodology. It provides guide-lines on how KBS should be integrated into the organisation at the information level. It specifies users' needs in terms of functions to be performed and the data relationships between them to produce a 'structured specification'. This specification is used in the proceeding design and implementation phases. A logical description of the knowledge-base is also produced to describe the logical information needs of the project.

Martinez & Sobol(88) propose a similar technique based on the use of context diagrams, data-flow diagrams and decision table. In both methodologies, the emphasis is upon the structured representation of knowledge through logical and intermediate representations . This provides greater control over the knowledge elicitation process, and by being formalised in this way,. makes it possible to incorporate this stage into a sequential and structured life cycle model..

A2. Methodologies based on the 'RUDE' Model

The justification for the use of prototyping is based on the inadequacies of traditional systems development techniques for ES development. From the literature, two methods are apparent which fall into this category:

A2.1. The Greenwell Methodology

This is a cyclic methodology based on three stages of feasibility, a 'first-cut' phase, and a 'main-phase'(Greenwell:1988). The feasibility stage provides guide-lines on testing the validity of identified problems for first-cut prototyping. The first-cut phase involves the design and development of the equivalent to a 'throw-away prototype'. Once validity and full development potential has been verified, this prototype is discarded and main phase development commences on the basis of a document specifying a design approach. The main phase itself is cyclical although only to the extent that incremental growth of the design takes place until design objectives have been achieved. During this phase, the prototype is continually assessed using quality assurance procedures.

A2.2. *The Abacus Methodology*

In this approach, the emphasis is upon rapid prototyping in order to quickly produce a demonstration system and thereby limit the scope of the problem (Citrenbaum et al: 1986). This system is used to evolve an appropriate systems architecture, knowledge representation and searching strategy. unlike the Greenwell methodology though, iterations of the knowledge base continue until a fully operational system is produced. In the latter stages of the prototype cycle, two copies of the system are made. the first is frozen and used without any additions, whilst the second copy is allowed to evolve, only replacing the frozen system when a 'noticeable' difference in performance is evident.

A3. 'Hybrid' Methodologies which Combine SPIV & RUDE Models

Chapter 4 concluded that in order to 'engineer' an ES, the iterative features of the RUDE model needed to be incorporated within the structured and controllable SPIV model. Thus 'hybrid' methodologies have evolved. Two in particular have received extensive coverage in the technical literature:-

A3.1. The GEMINI Methodology

GEMINI stands for 'Government Expert System Methodology Initiative' and was set up to define a standard approach for the construction of 'large' Expert Systems (Montgomery et al:88). The basis of GEMINI is that the methodology should be integrated with conventional systems and have full compatibility with existing software development analysis and design methods. Although it is intended to be independent of tools, it is intertwined and heavily reliant upon SSADM, a conventional life cycle methodology. Consequently the structure of the methodology follows the the six stages of SSADM (1 -analysis of systems operations; 2-specification of requirements; 3 -selection of technical options; 4 - data design; 5 - process design; and 6 - physical design). However, instead of the sequential organisation of SSADM, each of these stages in GEMINI each receive their inputs and pass their outputs to a validation process based on incremental prototyping. The KBS specific tasks such as knowledge acquisition are included as methods available within the relevant stages of the SSADM method. The resultant methodology is a hybrid approach in which the role, contribution and interfacing of prototyping and other KBS specific activities are well defined and closely monitored by conventional software engineering components.

GEMINI covers the activities involved from the inception of the product through to design (Montgomery:1989). It does not cover the subsequent stages of system building, delivery and post delivery, nor does it cover the identification of 'appropriate' projects in human or business terms. Therefore, although it is an improvement on earlier methodologies which follow exclusively a SPIV or RUDE model, it shares their weakness in focusing upon design and analysis issues only.

A3.2. The Quality Assurance Methodology

This is more of an approach than a methodology and is based on the recognition that a standard method is difficult to attain and that prototyping is indispensable but has inherent problems in the management of implementation (Born: 1988). The principal role of prototyping, as part of a wider methodology, is to reduce the uncertainty of development by providing direct experience of the areas of risk. Therefore, methods are proposed which make prototyping more accountable through the use of SE principles of formal *documentation* and *change control.*

A3.2.1. *Documentation:* The method proposes formal documentation for objectives and acceptance criteria; knowledge acquisition; user requirements; constraints (e.g., organisational, time and resources); systems design and integration requirements; and technical and user documentation. The role of the documentation is to (after Born):-

i) confirm requirements and objectives that the prototype sets out to satisfy,
ii) formulate acceptance criteria for prototyping,
iii) provide visible record of prototype design activities for assessment,
iv) enable the prototype to be recreated later on in a defined state,
v) record explicitly user and expert input and feedback.

Each document is made up of reports, specifications, program listings and test records. The quality and validity of each document is monitored and ensured by following detailed guide-lines and procedures.

A3.2.2. *Change Control:* This is the means of controlling and recording changes to all a project's deliverables, including documentation and software for each prototype cycle. This is made possible through constant monitoring and the formal recording of progress; the use of change control procedures; and documentation. Detailed check-lists are also provided for project management and project review respectively.

The purpose of this approach is to provide a quality assurance structure which may be transplanted with in-house methodologies developed according to specific organisational and technical needs of the organisation. In this respect, this approach is much less prescriptive than its predecessors.

B. Organisational Enhancements to Assessment & Development

The above is essentially a literature review of current expert systems practice and suffers from the limitations of a 'technical perspective' described in Chapter 3. In order to enhance a technical model of development, Chapter 4 looked at the viability of applying approaches and development concepts which derive from a root organisational perspective. This was further broken down into concepts which take an external view of the organisation, essentially addressing business and strategic issues, and an internal view of the organisation which focus upon issues of problem definition and the management of change brought about by the introduction of ES into a company as a socio-technical system. In both cases, it was necessary to look beyond the ES field of literature in order to define viable methods and approaches.

B1. The Contribution of External Organisational Approaches

Three factors are stressed from this viewpoint. First the need to extend the scope of technology assessment and evaluation to consider the entire process of technology transfer; this issue is considered in detail in Chapter 8. Secondly, the importance of commencing technology assessment not by an evaluation of technology capabilities, but by an assessment of organisational needs: the whole of Chapter addresses this issue with respect to the use of IDEFo. Finally, it underlines the necessity of qualifying the business value and strategic importance of expert systems in some way. There are a plethora of business tools which may be used for this purpose; Earl (1989) provided a useful categorisation of these and examples are given in each case which were considered appropriate for expert systems particularly.

B1.1. Awareness Frameworks

These could be used to demonstrate how ES may be used for strategic advantage, and help senior management in an organisation assess the potential impact on their business, internally and externally. Awareness frameworks are more conceptual than prescriptive devices. Examples of awareness frameworks include:-

i) Re-focussing Models : these may be used to change and re-orient executive thinking towards ES potential. A possible tool for this role is the Strategic Opportunity Matrix (Benjamin:1984)

ii) Impact Models: these may be used to indicate the scale of strategic change brought about through the implementation of ES. The most common tool of this category, which may be adapted for ES evaluation, is Parson's Impact matrix (1983),
iii) Scoping Models: these help to identify the possible overall strategic scope of ES in the organisation. Porter's Information (knowledge) Intensity Matrix may be useful here (1985).

Awareness frameworks are therefore of use in identifying and communicating ES possibilities. They are likely to be of greatest value at the 'creative' opportunity phase of the Inside-out business strategy. However, they are too high level and too descriptive to identify specific ES applications.

B1.2. Opportunity Frameworks

Unlike awareness frameworks, opportunity frameworks are analytical tools which are used to systematically define an organisation's business strategy. They could also be used to clarify business strategies in order to demonstrate options for using ES. Earl identified four sub-classes of opportunity framework:-

i) System analysis models: these are techniques which investigate information flows, problems, limitations and strengths in organisational and business activities from a strategic perspective. The most frequently cited system analysis model is Porter and Miller's value chain analysis (1985),

ii) Application search tools: these match the characteristics of the application domain, as a business entity, against the characteristics of ES for goodness of fit. They would also be useful in suggesting the direction that ES application development should follow. Earl notes Ives & Learmouth(1984) as an example of this class.

iii) Technology fitting frameworks: this approach define the common attributes of ES to see if they can be applied to particular business problems. In this case, the common attributes could be based on empirical characteristics of 'successful' ES projects. This approach is used extensively in ES development, although biased towards defining technically feasible ES solutions (e.g.Prerau:1985) rather than defining appropriate business applications. These frameworks are subsequently more effective if they are preceded by a business strategy framework.

iv) Business Strategy Framework: these help to define or verify an organisation's business strategy. They also provide a business context by which technology can be used to exploit or improve competitive advantage. Earl uses Porter's Five Forces Model as an example of this class (Porter:1980).

B1.3. Positioning Frameworks

These frameworks are orientated towards implementation rather than formulating business strategy. They may be used to help clarify the current IT situation of the organisation against which new ES developments may be evaluated. The aim therefore is to improve understanding of how ES should be managed according to the specific structure and layout of the organisation. Earl identifies three classes of positioning framework:-

i) Scaling Frameworks: these help to indicate the current and possible future importance of ES to a business from which management policies and practices are re-defined. An example of this class is McFarlan & McKenneys' Strategic Grid (83).

ii) Spatial Frameworks: these could be used to help indicate the characteristics of ES applications and ES management in different parts of the organisation or sector-Earl's own sector model (1989) covers the latter.

iii) Temporal Frameworks: these help to assess the evolutionary position of an organisation in using and managing IS to highlight problems and inconsistencies. They may then be used to suggest a next stage of development with or without ES. The generic technology model of McFarlan & McKenney(83) could be used in this capacity.

Chapter 4 describes how these models and frameworks may be linked to one of three business strategies: top-down; bottom-up; or inside-out. These relationships are summarised in Figure 3A below:

Framework Strategy	AWARENESS FRAMEWORKS			OPPORTUNITY FRAMEWORKS				POSITIONING FRAMEWORKS		
	Re-focusing	Impact	Scoping	Systems Anal.	Search tools	Fitting	B. Strategy	Scaling	Spatial	Temporal
Top-Down Business Objectives							●			
Identify C.S.F.'s				●			●			
Identify Problems				●					●	
Identify ES Potential	●	●	●		●	●				
Define Application Plan					●			●		●
Bottom-Up Define Application Plan								●		●
Identify ES Potential		●			●	●			●	
Define Development Priorities	●		●				●			
Identify Problems				●						
Understand Present Situation				●				●		
Inside-Out Define Business Objectives							●			
Identify Business Opportunities	●			●				●		
Identify Appropriate Areas		●		●	●	●			●	●
Define ES Opportunities		●	●		●					
Define Technical Feasibility						●				

Figure 3A : The Positioning of Business Strategy & Frameworks for Expert Systems

B2. The Contribution of Internal Organisational Approaches

Internal organisational approaches view the organisation as a socio-technical system and contribute towards development in two ways in particular: first by providing techniques which help to define a consensus view of the organisational problem-Checkland's Soft System Methodology is representative of this genus (1981); and secondly, by providing the means to manage the process of change more appropriately through participation and involvement in the design and development process- Mumford's ETHICS method was chosen in this case (198.

B2.1. Soft Systems Methodology

The conceptual basis of SSM is described in Chapter 3 and is considered in Chapter 5 as a possible means of modelling the organisation. It is intended more as a 'learning and enquiry system'(Checkland:89) or 'problem-solving framework' (Checkland:1985) rather than a specific technique however, and it uses system based models to understand what Checkland calls 'real-world' problems. The steps within the methodology, shown diagrammatically in Figure 3B., are categorised as 'real-world' activities and 'system thinking' activities.

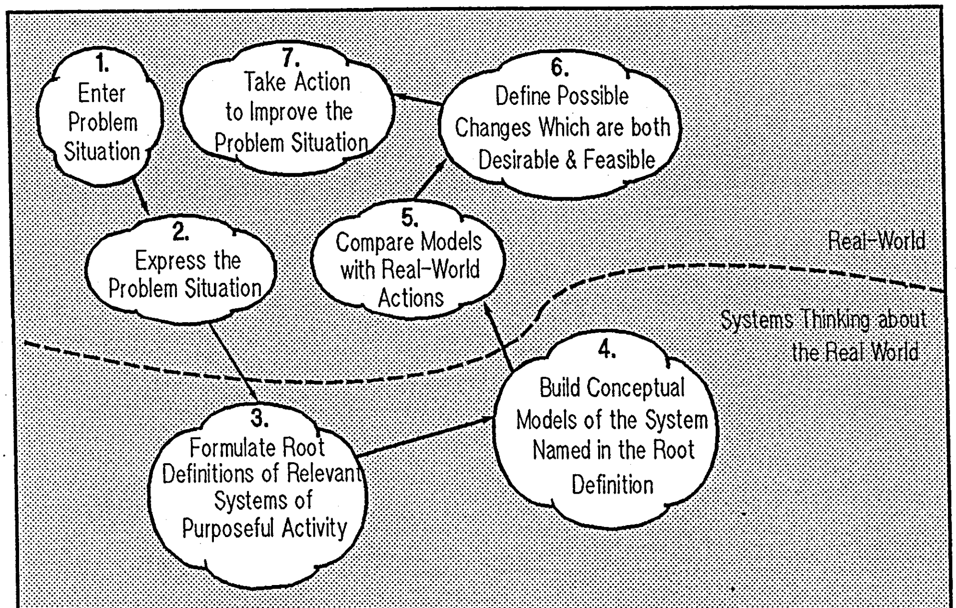


Figure 3B: An Overview of Checkland's Soft Systems Methodology (adapted from Checkland:1989; Fig 3; p281)

This process is non-sequential in that Checkland claims that a project can be started at any stage, and further that backtracking and iteration are essential to defining acceptable solutions. The steps in the former are carried out by those people within problem situation and involves a process of identifying relevant 'human activity systems' (relevant denoting focus upon an identified socio-technical issue or organisational conflict). The steps involved in the 'system thinking' activities attempt to provide conceptual models of the real world based upon individual viewpoints expressed by different 'root-defintions' of the chosen problem. These models are then compared with real-world perceptions of the problem situation from which a consensus or 'accommodation' is arrived at based upon agreement of the problems and the changes by which they may be resolved.

Checkland argues that by using the methodology, these changes are 'desirable' as a result of the insight gained from building and selecting root definitions and creating conceptual models. The changes are also 'culturally feasible', Checkland claims, in that the methodology takes account of the characteristics of the organisation and organisational roles and relationships.

B2.2 The 'ETHICS' Method of Participation

Mumford divides ETHICS into two sets of objectives, as *Figure 3C.* shows: first technical and economic goals; and second, goals for improving job satisfaction of those who work with and around the system (Mumford: 1983). Mumford defines 'job satisfaction' as the match between the expectations that people bring to the job and the requirements of the job as defined by the organisation. The main purpose of ETHICS is the identification of compatible pairs of alternative technical and social designs after establishing technical and social objectives. From this, technical and social alternatives are developed independently by different groups of analysts, but with the participation of users. The technical alternatives are then compared and matched up with the social alternatives compatible with them. This provides a set of 'feasible' socio-technical solutions, which are evaluated against the technical and social objectives for the system and ranked. A detailed design is then prepared for the best alternative and implemented if acceptable.

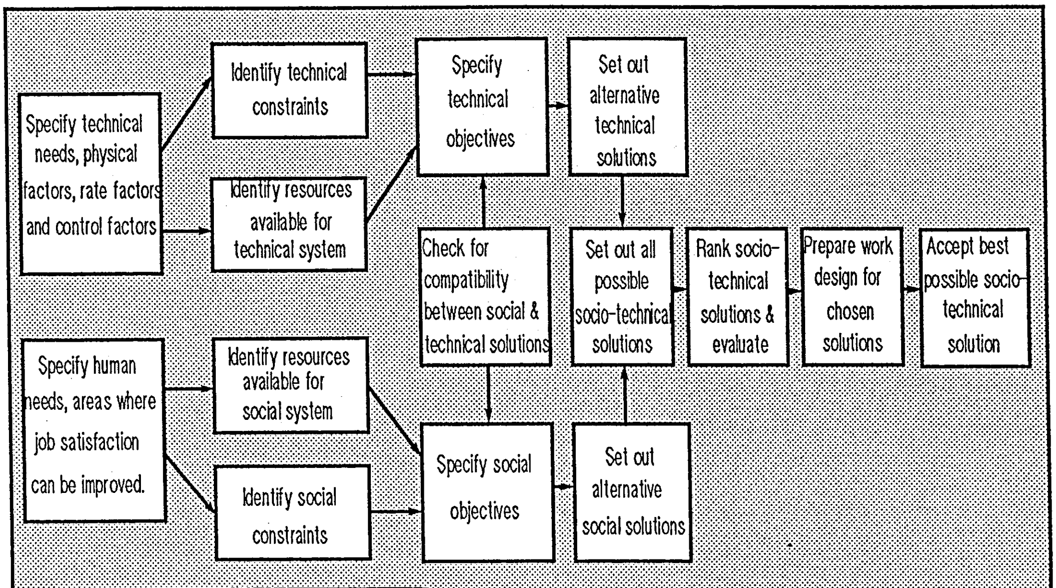


Figure 3C: An Overview of the 'ETHICS' Methodology

Source: Based on Mumford & Weir (1979):p37

Appendix IV

A Functional Analysis of the Client Organisation Using IDEFo

Part A: A Top-level Business Model of 'Computing Services'

Here, IDEFo is used to model the computer department as a business support function. Diagrams are drawn on a standard graphics package (Hewlett Packard 'Drawing Gallery')

Part B: A Physical Model of Computer Systems Architecture

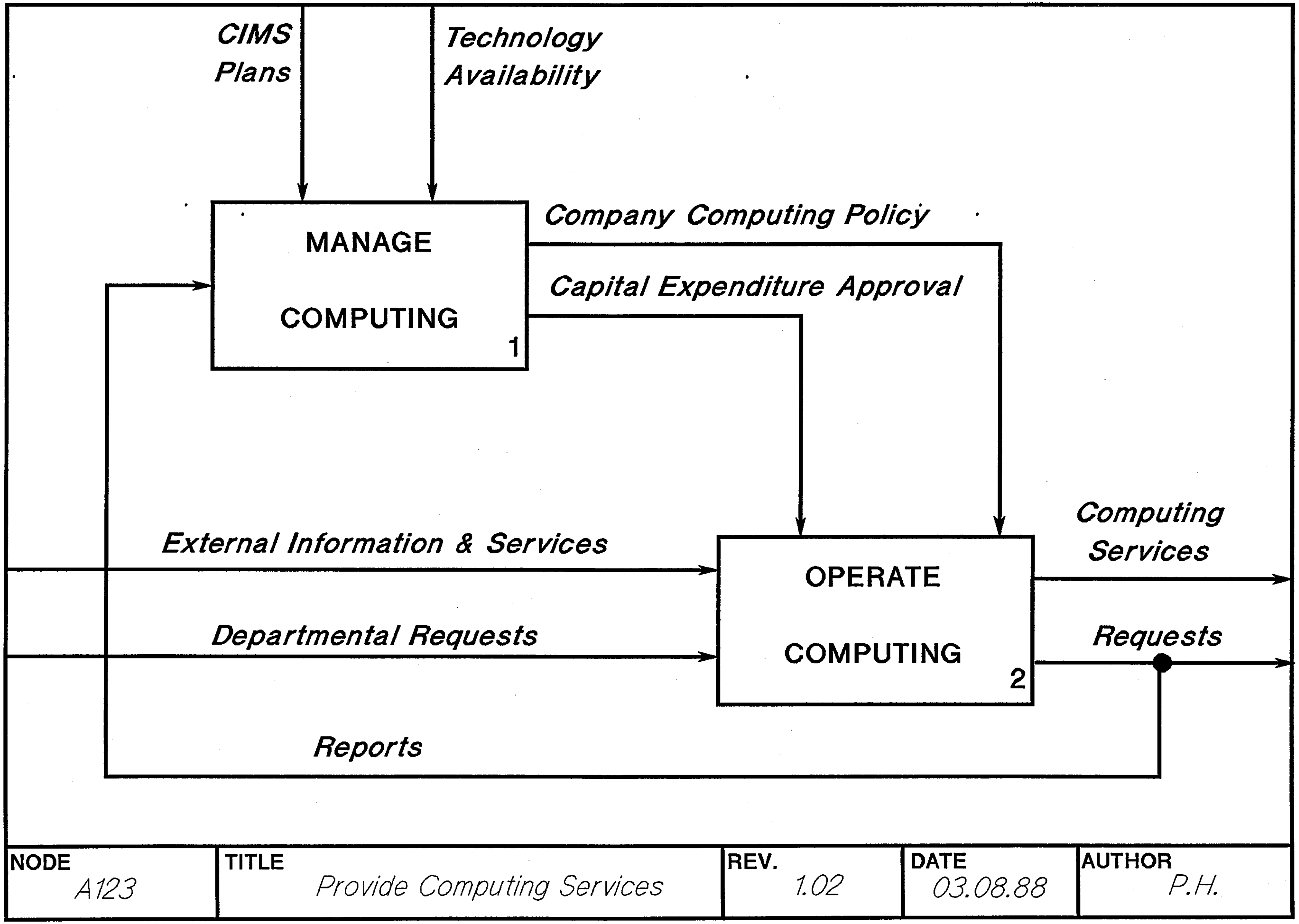
This shows the use of the model at a lower organisational level in terms of physical computer systems and information flows. The model is orientated about the use of computer systems as **mechanisms** *and therefore describes the actual use of these systems in the company in performing information transactions. The model was drawn using a dedicated CASE tool.*

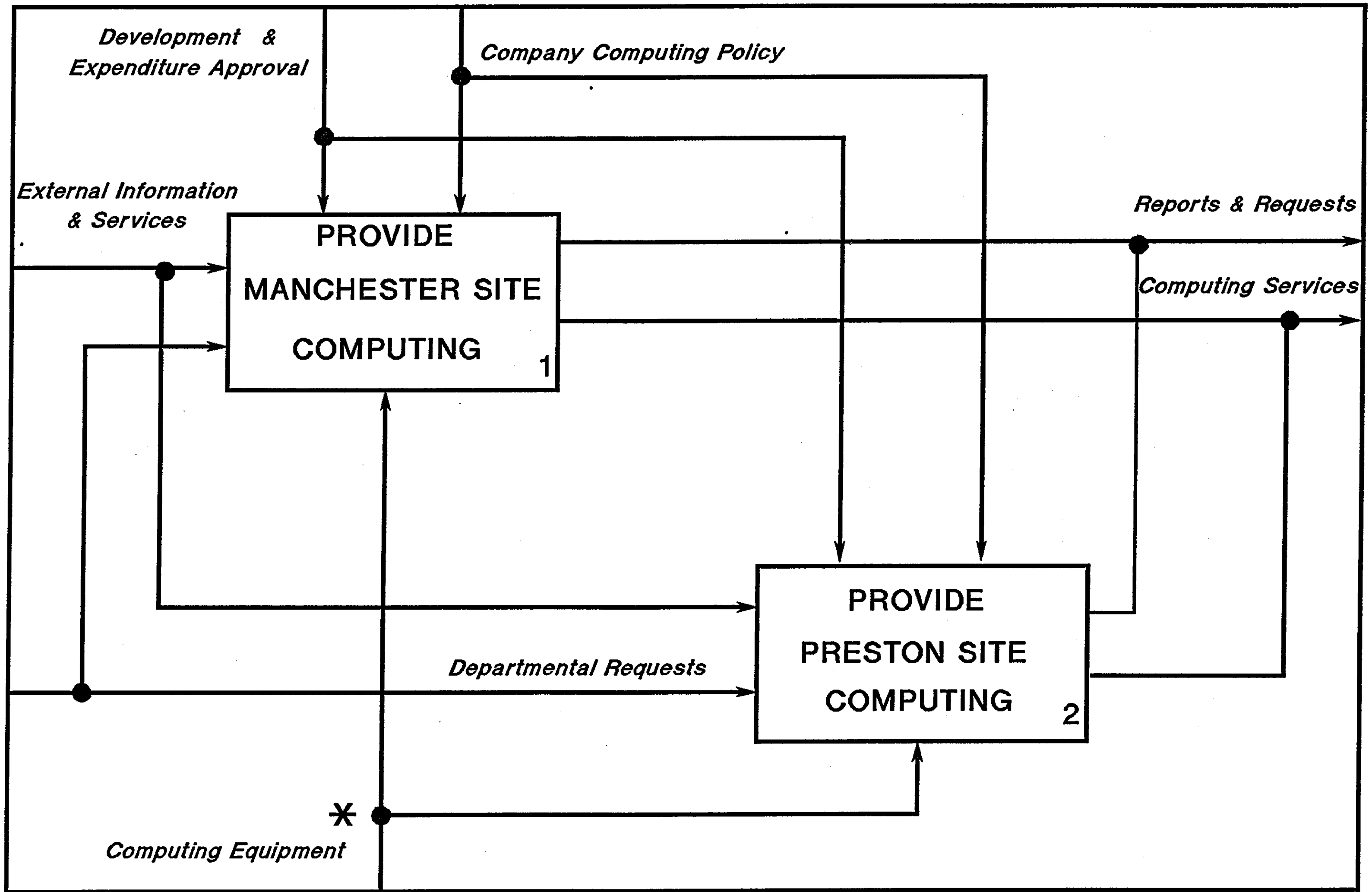
Appendix IV: Part A.

A Selection of Nodes from the IDEFo Top-Level Model *

---

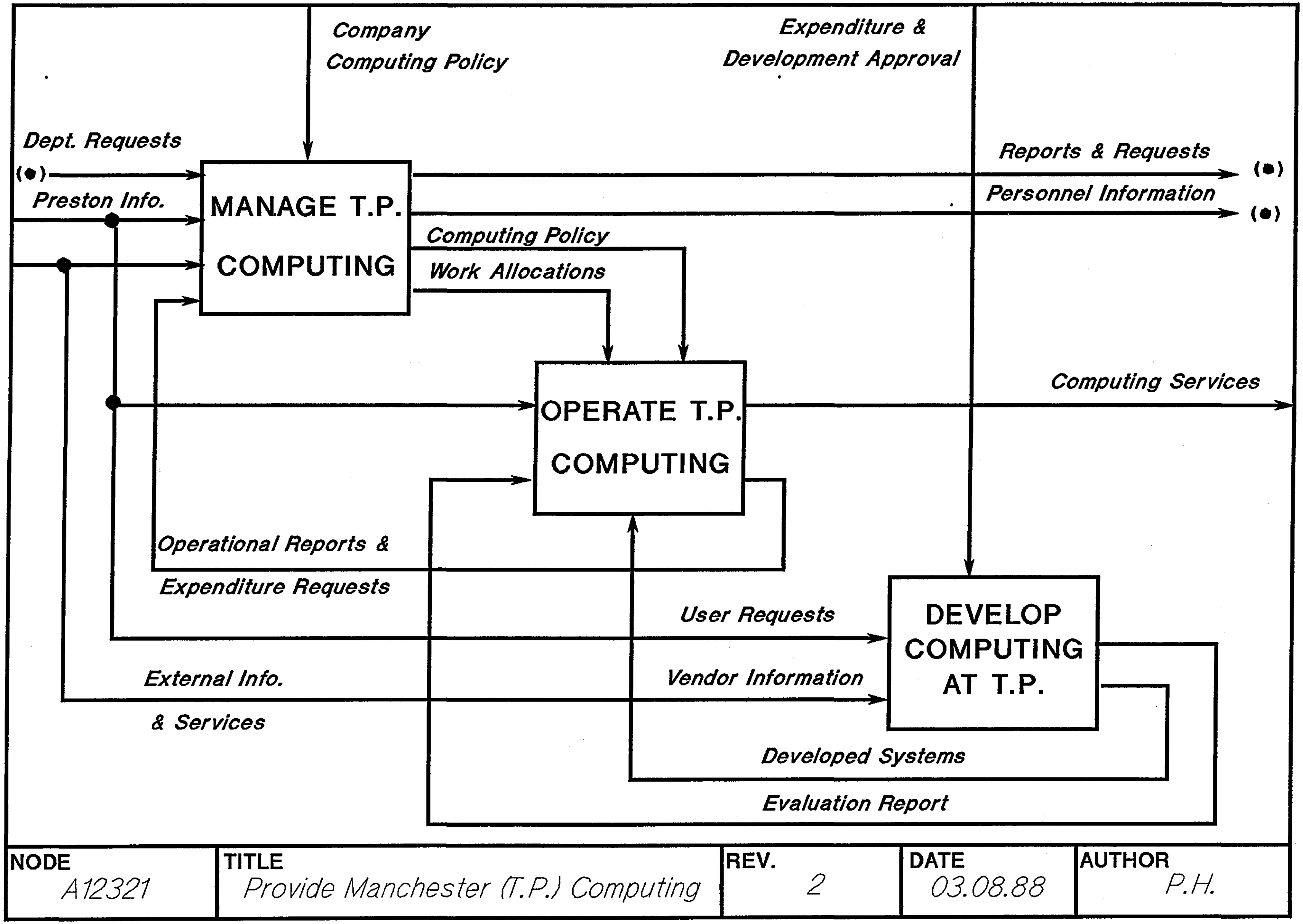
* *These diagrams may not be reproduced; nor may information be used from them without the permission of the author and ultimately of the sponsoring organisation.*

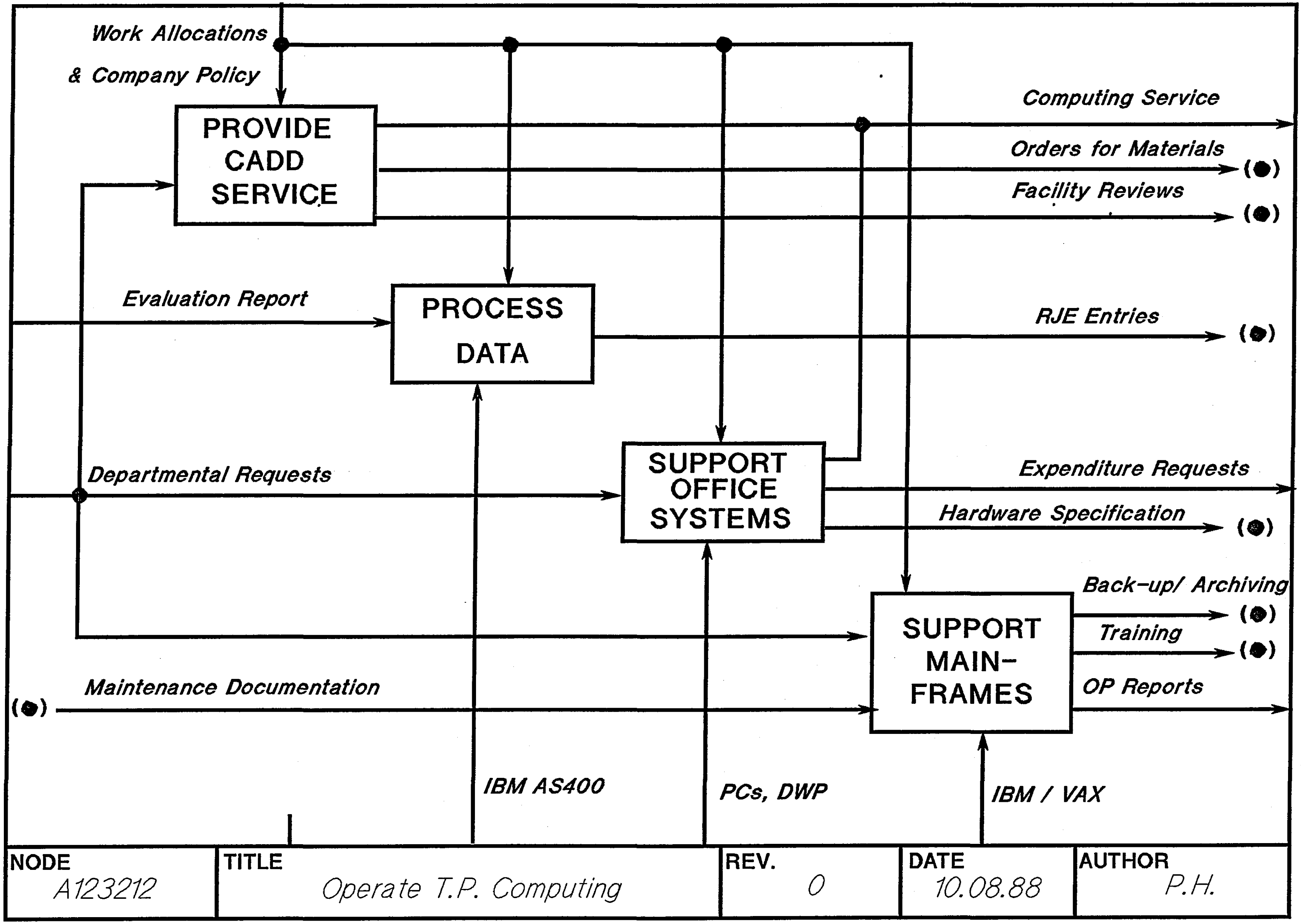

CIMS Plans
Technology Availability
MANAGE COMPUTING
1
Company Computing Policy
Capital Expenditure Approval
External Information & Services
Departmental Requests
OPERATE COMPUTING
2
Computing Services
Requests
Reports
NODE
A123
TITLE
Provide Computing Services
REV.
1.02
DATE
03.08.88
AUTHOR
P.H.

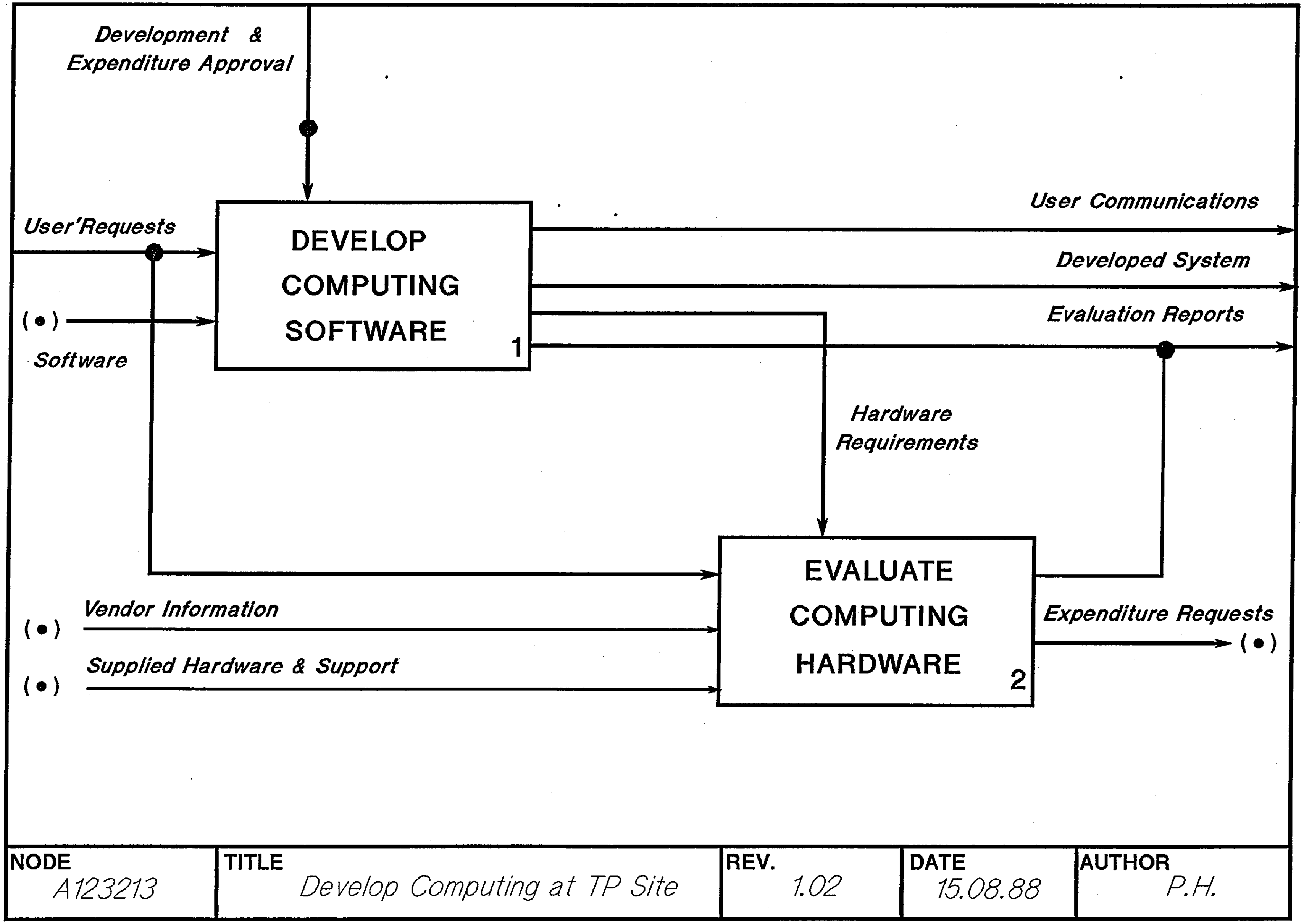


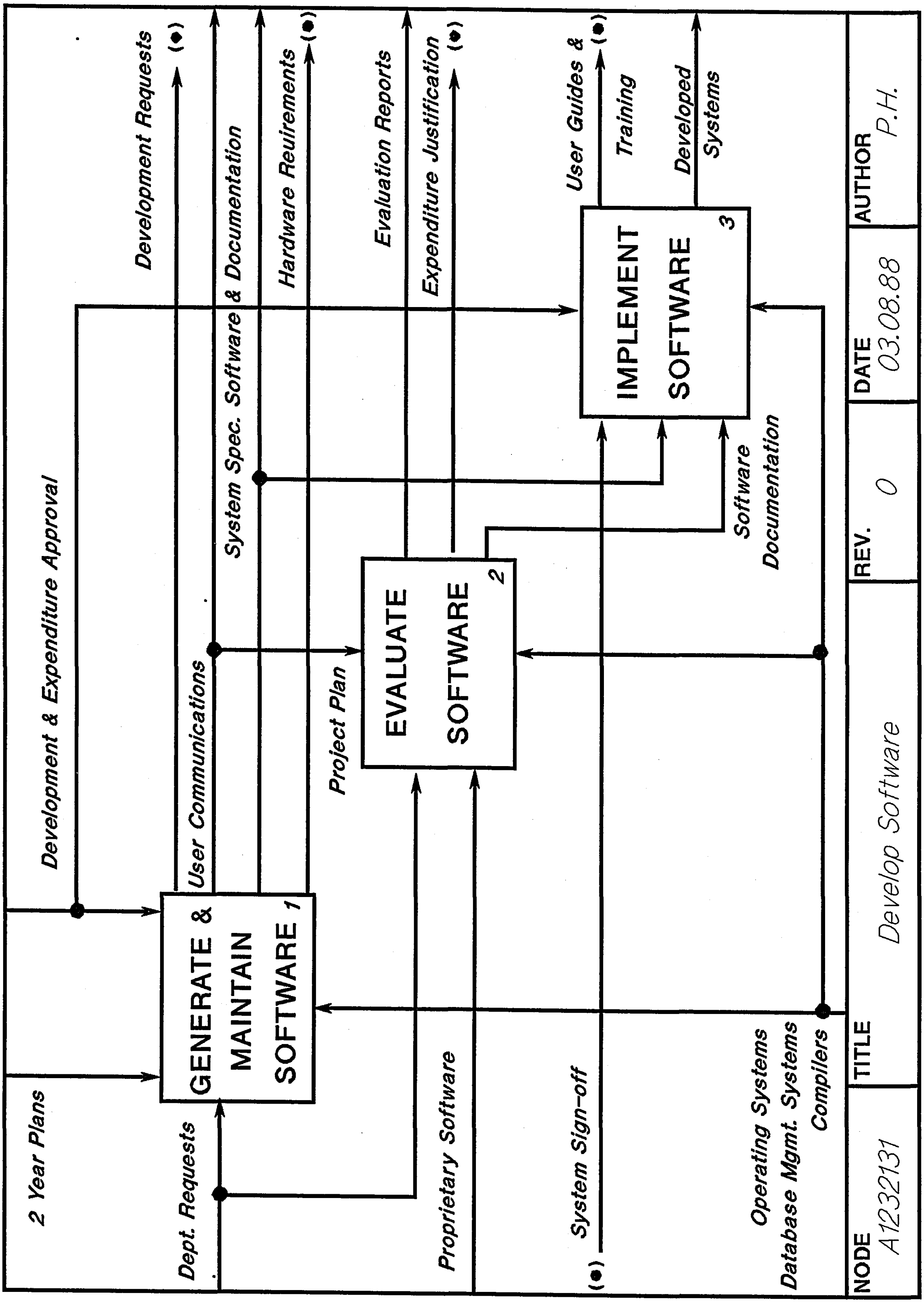
NODE A1232	TITLE Operate Computing Services	REV. 1.02	DATE 03.08.88	AUTHOR P.H.

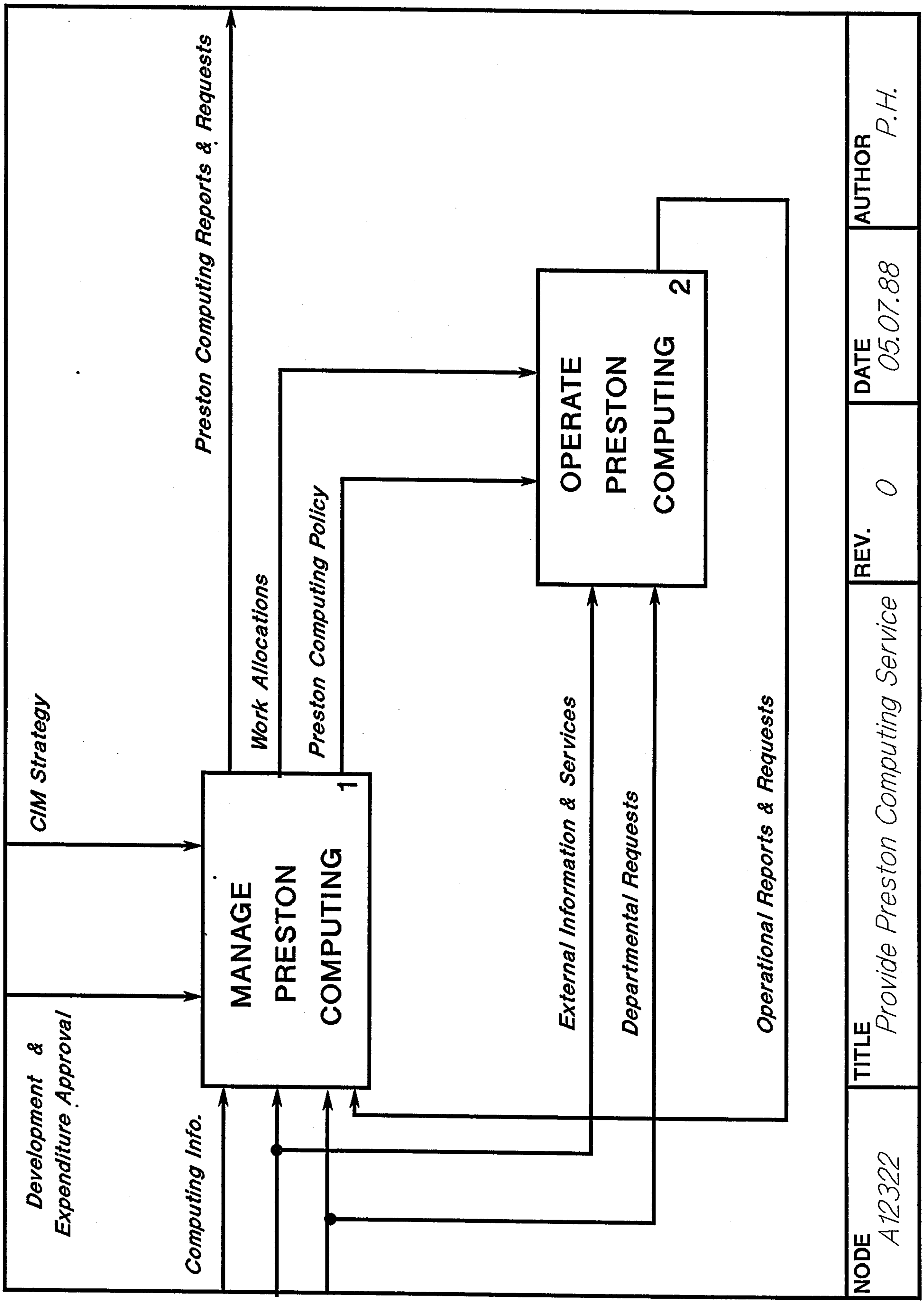
* See Appendix IV Part B For Computer Mechanisms

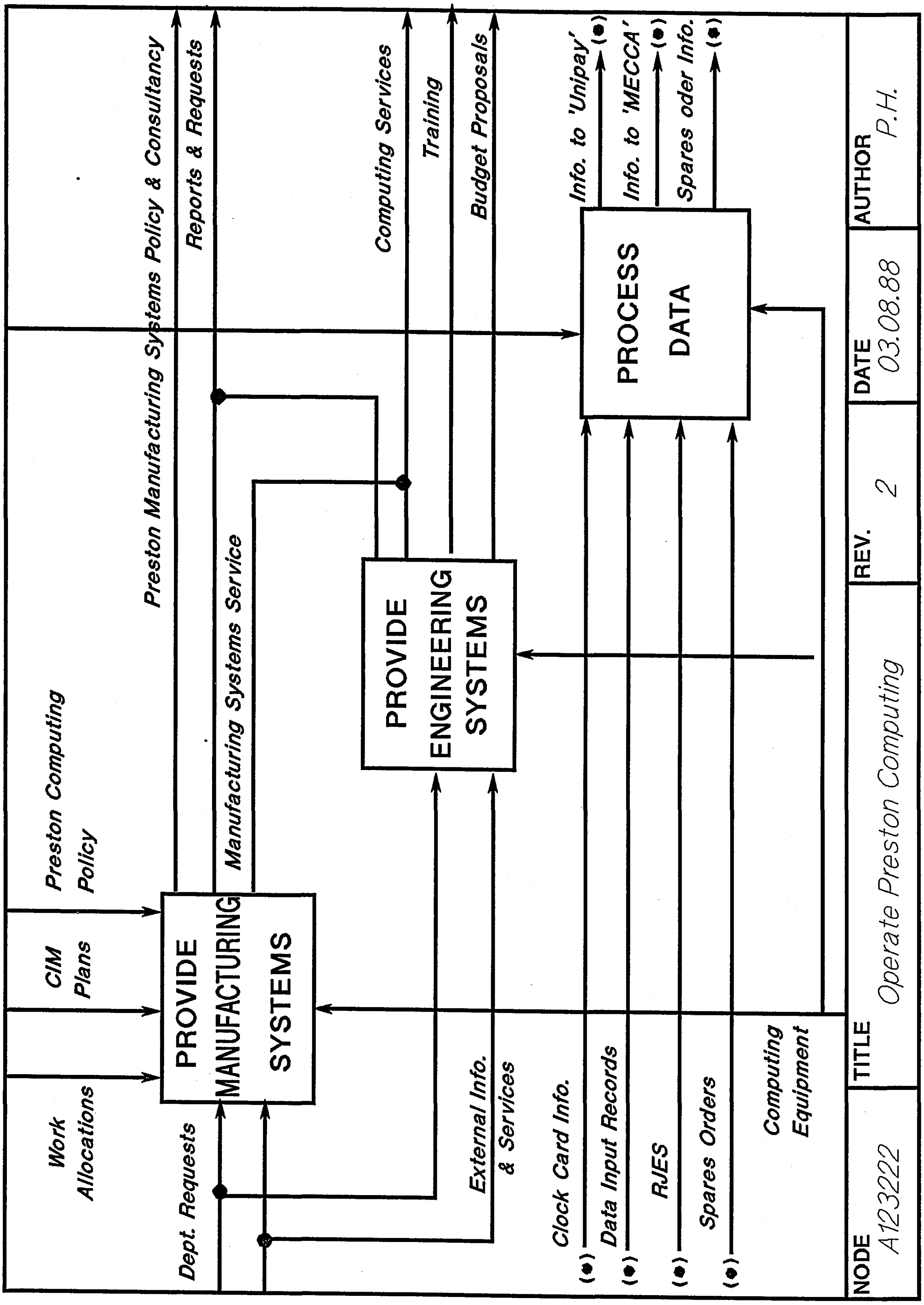

Company
Computing Policy
Expenditure &
Development Approval
Dept. Requests
(●)
Preston Info.
MANAGE T.P. COMPUTING
Reports & Requests
(●)
Personnel Information
(●)
Computing Policy
Work Allocations
OPERATE T.P. COMPUTING
Computing Services
Operational Reports &
Expenditure Requests
User Requests
DEVELOP COMPUTING AT T.P.
External Info.
& Services
Vendor Information
Developed Systems
Evaluation Report
NODE
A12321
TITLE
Provide Manchester (T.P.) Computing
REV.
2
DATE
03.08.88
AUTHOR
P.H.


Work Allocations
& Company Policy
PROVIDE CADD SERVICE
Computing Service
Orders for Materials
Facility Reviews
Evaluation Report
PROCESS DATA
RJE Entries
Departmental Requests
SUPPORT OFFICE SYSTEMS
Expenditure Requests
Hardware Specification
Back-up/ Archiving
Training
OP Reports
SUPPORT MAIN-FRAMES
Maintenance Documentation
IBM AS400
PCs, DWP
IBM / VAX
NODE
A123212
TITLE
Operate T.P. Computing
REV.
0
DATE
10.08.88
AUTHOR
P.H.


Development &
Expenditure Approval
User'Requests
(•)
Software
DEVELOP COMPUTING SOFTWARE
1
User Communications
Developed System
Evaluation Reports
Hardware Requirements
Vendor Information
(•)
Supplied Hardware & Support
(•)
EVALUATE COMPUTING HARDWARE
2
Expenditure Requests
(•)
NODE
A123213
TITLE
Develop Computing at TP Site
REV.
1.02
DATE
15.08.88
AUTHOR
P.H.


2 Year Plans
Development & Expenditure Approval
Dept. Requests
GENERATE & MAINTAIN SOFTWARE 1
User Communications
Development Requests (•)
System Spec. Software & Documentation
Hardware Reuirements (•)
Project Plan
Proprietary Software
EVALUATE SOFTWARE 2
Evaluation Reports
Expenditure Justification (•)
Software Documentation
(•) System Sign-off
Operating Systems
Database Mgmt. Systems
Compilers
IMPLEMENT SOFTWARE 3
User Guides & Training (•)
Developed Systems
NODE A1232131
TITLE Develop Software
REV. 0
DATE 03.08.88
AUTHOR P.H.


Development &
Expenditure Approval
CIM Strategy
Computing Info.
MANAGE PRESTON COMPUTING
1
Preston Computing Reports & Requests
Work Allocations
Preston Computing Policy
OPERATE PRESTON COMPUTING
2
External Information & Services
Departmental Requests
Operational Reports & Requests
NODE
A12322
TITLE
Provide Preston Computing Service
REV.
0
DATE
05.07.88
AUTHOR
P.H.

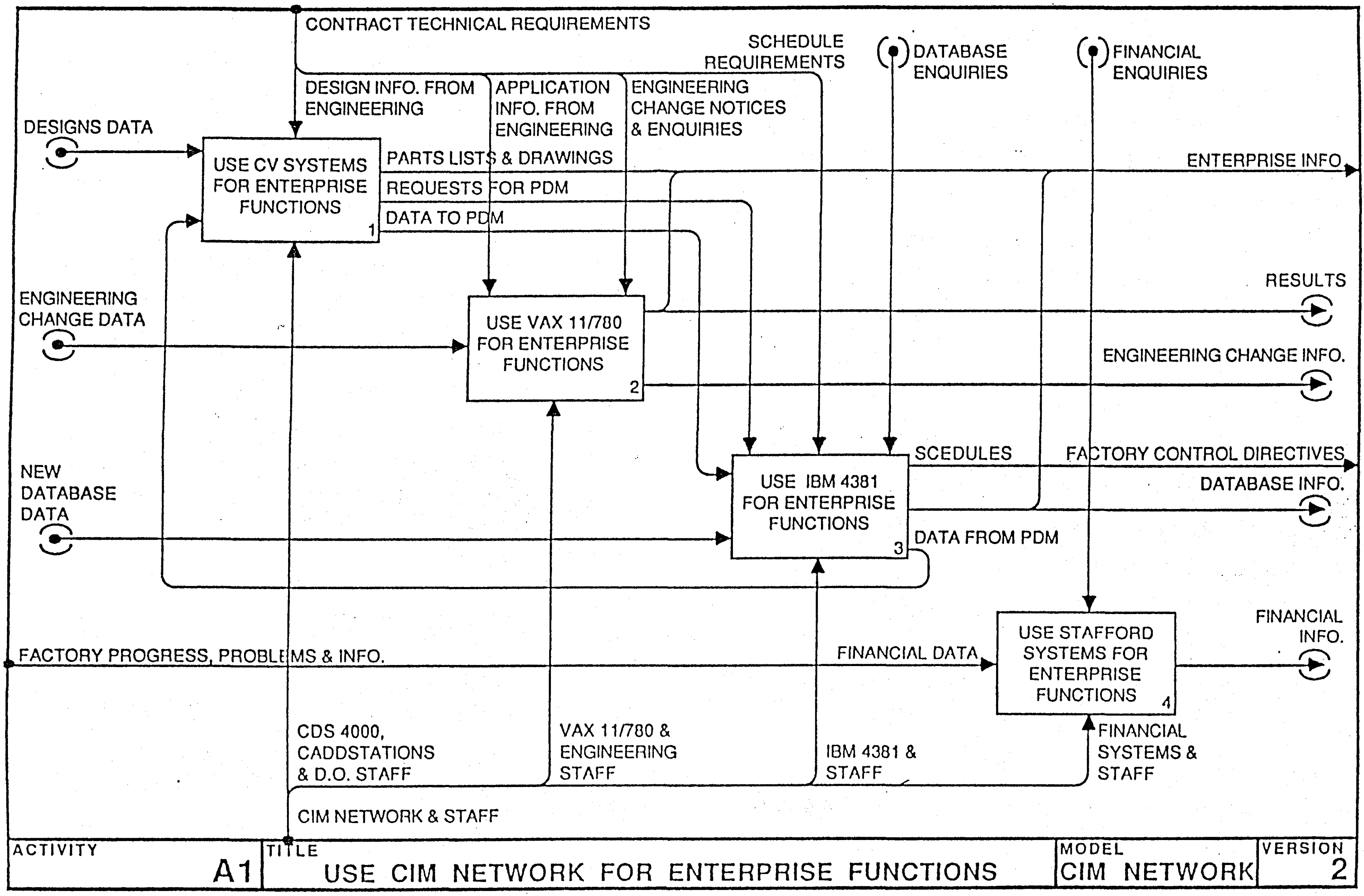

Work Allocations
CIM Plans
Preston Computing Policy
PROVIDE MANUFACTURING SYSTEMS
Dept. Requests
Preston Manufacturing Systems Policy & Consultancy
Reports & Requests
Manufacturing Systems Service
PROVIDE ENGINEERING SYSTEMS
External Info. & Services
Computing Services
Training
Budget Proposals
PROCESS DATA
Info. to 'Unipay'
Info. to 'MECCA'
Spares oder Info.
Clock Card Info.
Data Input Records
RJES
Spares Orders
Computing Equipment
NODE
A123222
TITLE
Operate Preston Computing
REV.
2
DATE
03.08.88
AUTHOR
P.H.

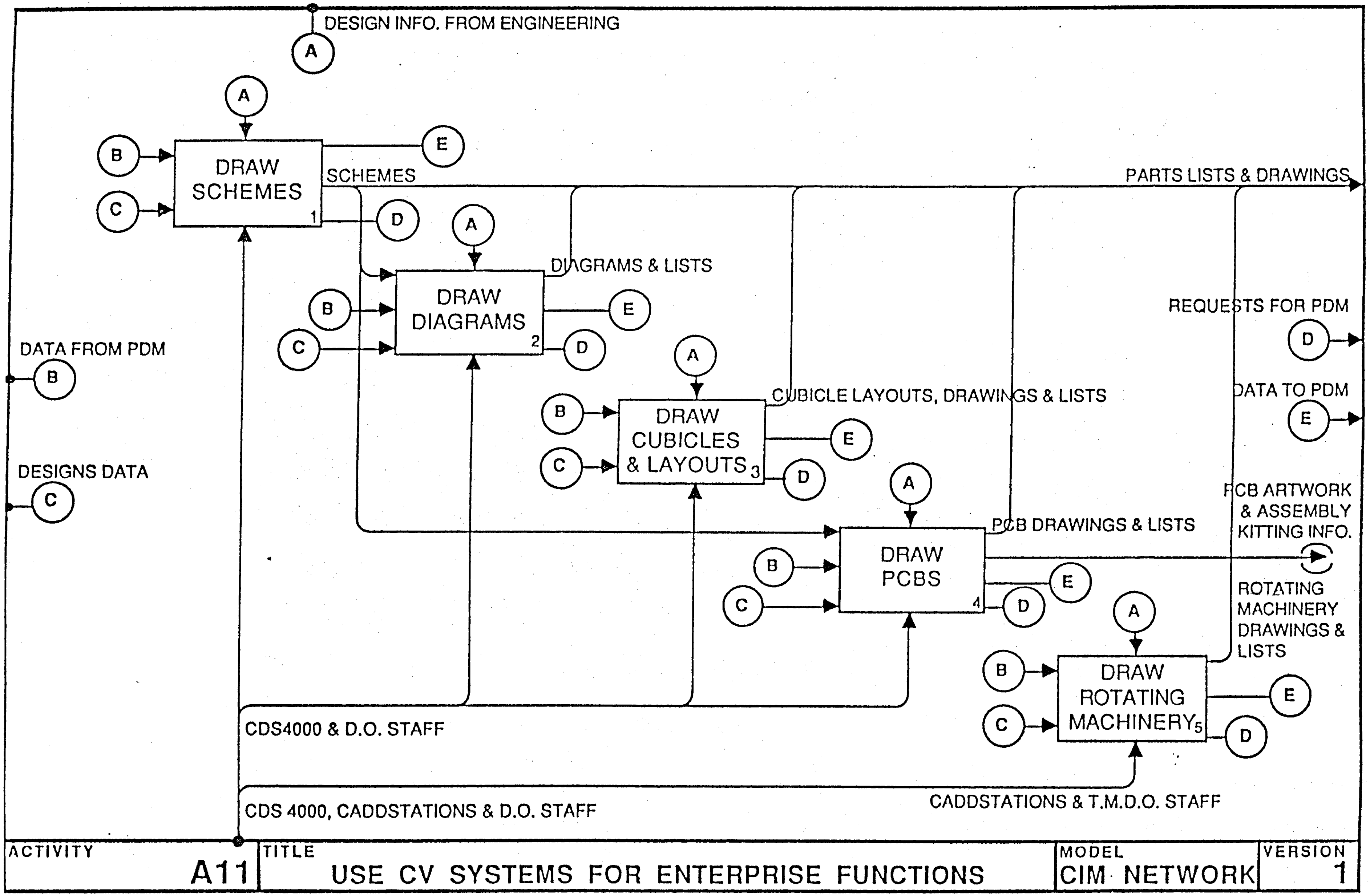
Appendix IV: Part B.

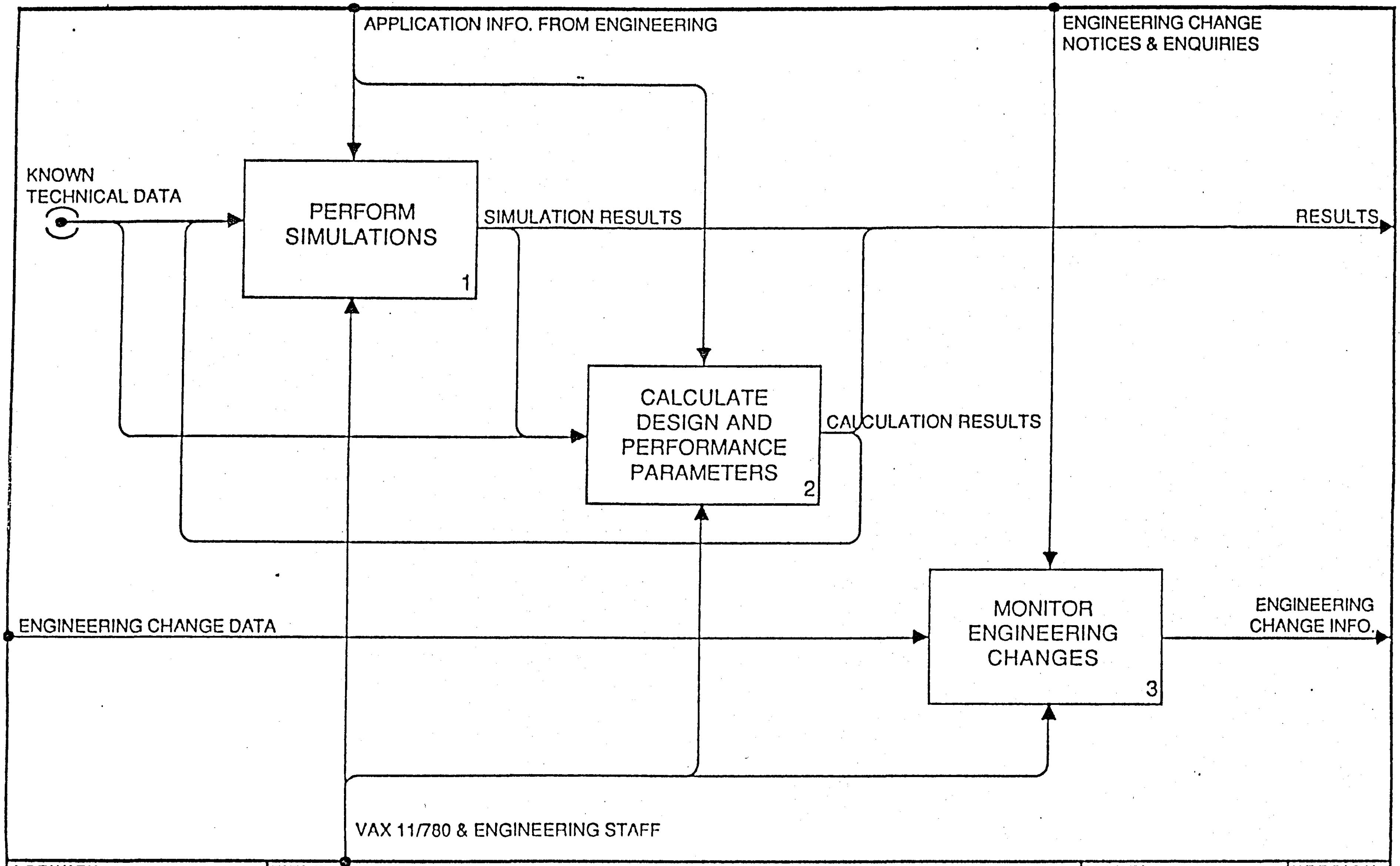
A Selection of Nodes from the IDEFo Physical Model *

---

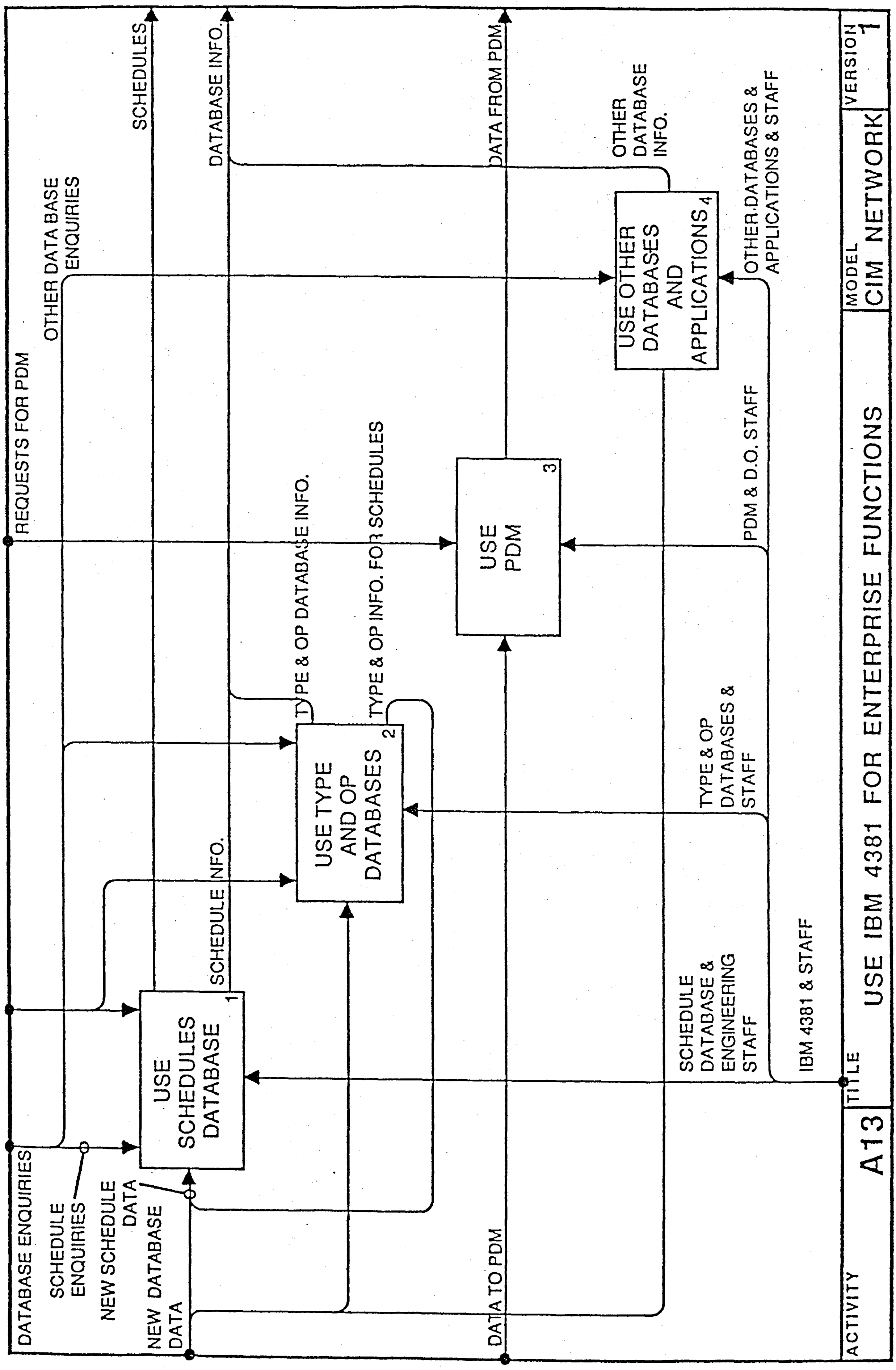
* *These diagrams may not be reproduced; nor may information be used from them without the permission of the author and ultimately of the sponsoring organisation.*

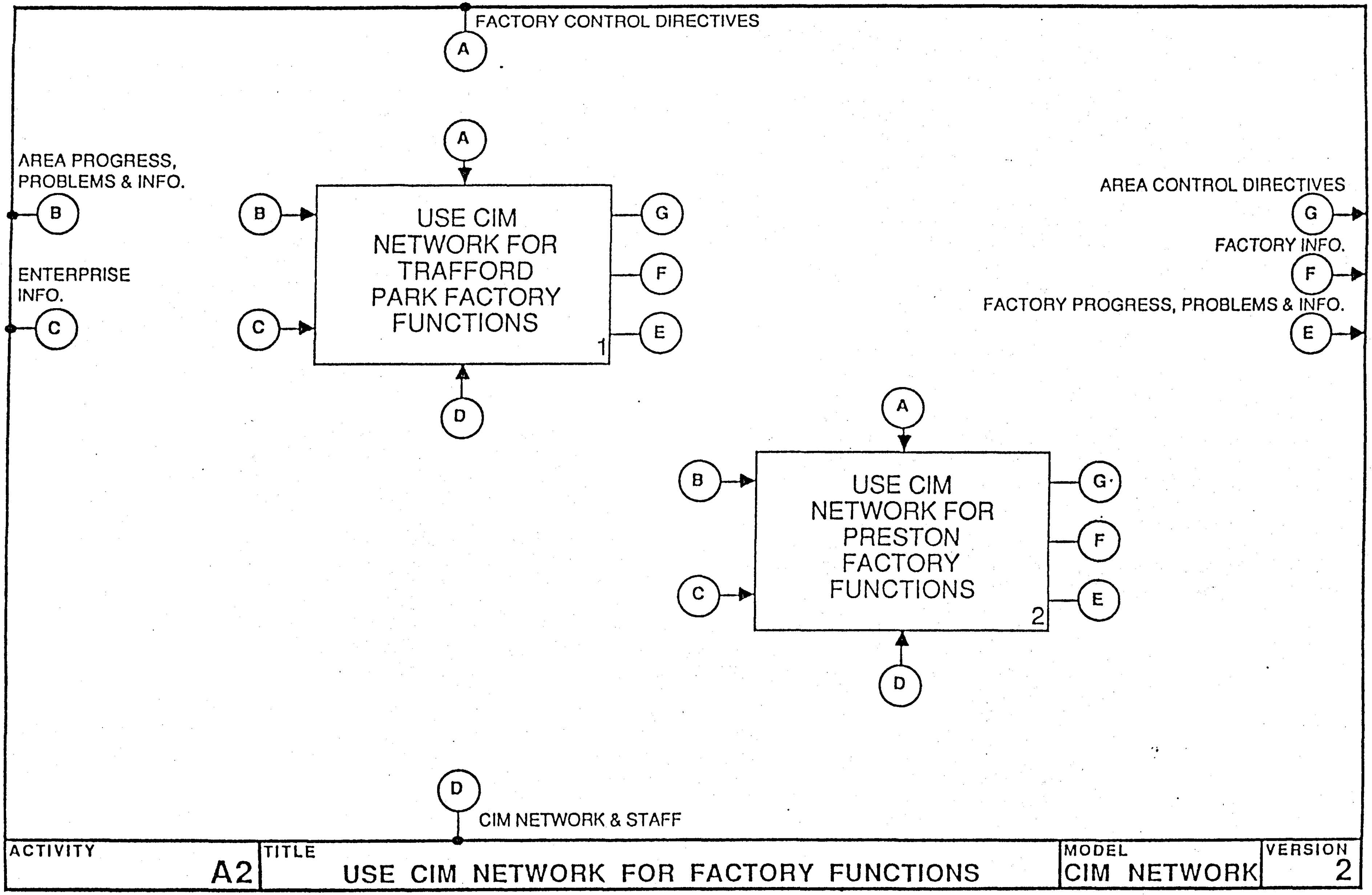

CONTRACT TECHNICAL REQUIREMENTS
SCHEDULE REQUIREMENTS
DATABASE ENQUIRIES
FINANCIAL ENQUIRIES
DESIGN INFO. FROM ENGINEERING
APPLICATION INFO. FROM ENGINEERING
ENGINEERING CHANGE NOTICES & ENQUIRIES
DESIGNS DATA
USE CV SYSTEMS FOR ENTERPRISE FUNCTIONS
1
PARTS LISTS & DRAWINGS
REQUESTS FOR PDM
DATA TO PDM
ENTERPRISE INFO.
ENGINEERING CHANGE DATA
USE VAX 11/780 FOR ENTERPRISE FUNCTIONS
2
RESULTS
ENGINEERING CHANGE INFO.
NEW DATABASE DATA
USE IBM 4381 FOR ENTERPRISE FUNCTIONS
3
SCEDULES
FACTORY CONTROL DIRECTIVES
DATABASE INFO.
DATA FROM PDM
FACTORY PROGRESS, PROBLEMS & INFO.
FINANCIAL DATA
USE STAFFORD SYSTEMS FOR ENTERPRISE FUNCTIONS
4
FINANCIAL INFO.
CDS 4000, CADDSTATIONS & D.O. STAFF
VAX 11/780 & ENGINEERING STAFF
IBM 4381 & STAFF
FINANCIAL SYSTEMS & STAFF
CIM NETWORK & STAFF
ACTIVITY
A1
TITLE
USE CIM NETWORK FOR ENTERPRISE FUNCTIONS
MODEL
CIM NETWORK
VERSION
2


DESIGN INFO. FROM ENGINEERING
A
DATA FROM PDM
B
DESIGNS DATA
C
DRAW SCHEMES
1
SCHEMES
DRAW DIAGRAMS
2
DIAGRAMS & LISTS
DRAW CUBICLES & LAYOUTS
3
CUBICLE LAYOUTS, DRAWINGS & LISTS
DRAW PCBS
4
PCB DRAWINGS & LISTS
DRAW ROTATING MACHINERY
5
PARTS LISTS & DRAWINGS
REQUESTS FOR PDM
D
DATA TO PDM
E
PCB ARTWORK & ASSEMBLY KITTING INFO.
ROTATING MACHINERY DRAWINGS & LISTS
CDS4000 & D.O. STAFF
CDS 4000, CADDSTATIONS & D.O. STAFF
CADDSTATIONS & T.M.D.O. STAFF
ACTIVITY
A11
TITLE
USE CV SYSTEMS FOR ENTERPRISE FUNCTIONS
MODEL
CIM NETWORK
VERSION
1

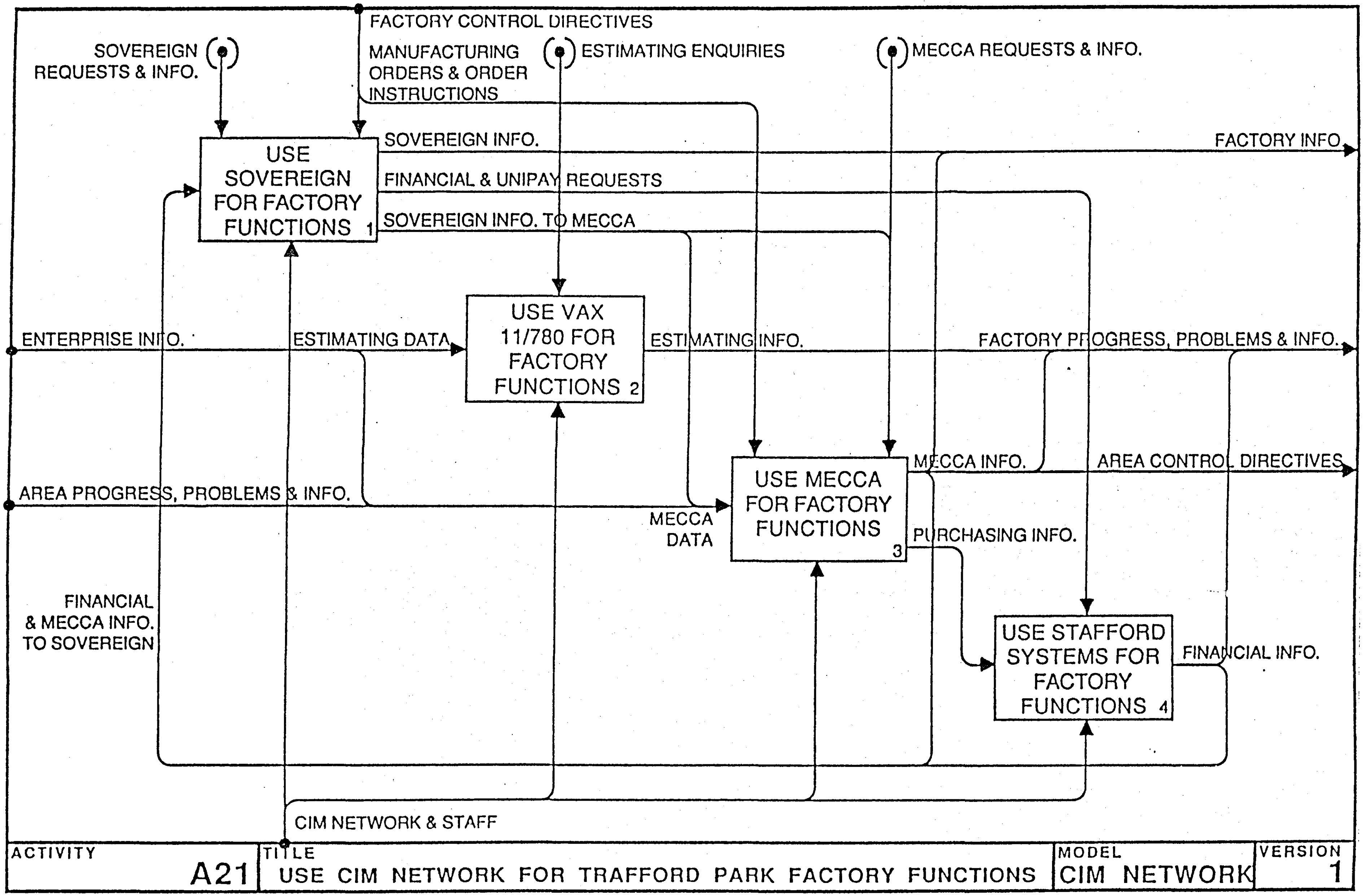


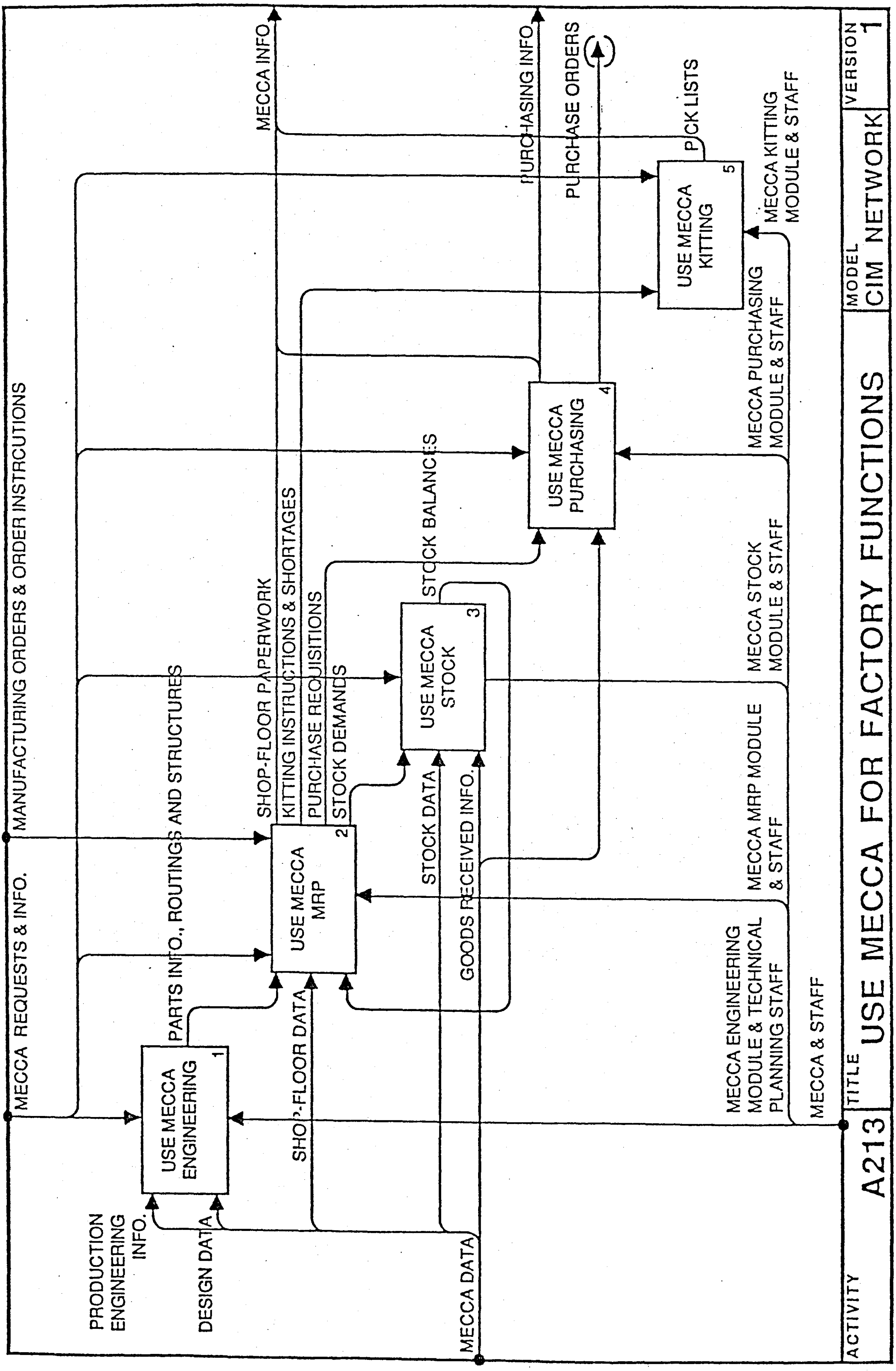
ACTIVITY	TITLE	MODEL	VERSION
A12	USE VAX 11/780 FOR ENTERPRISE FUNCTIONS	CIM NETWORK	1

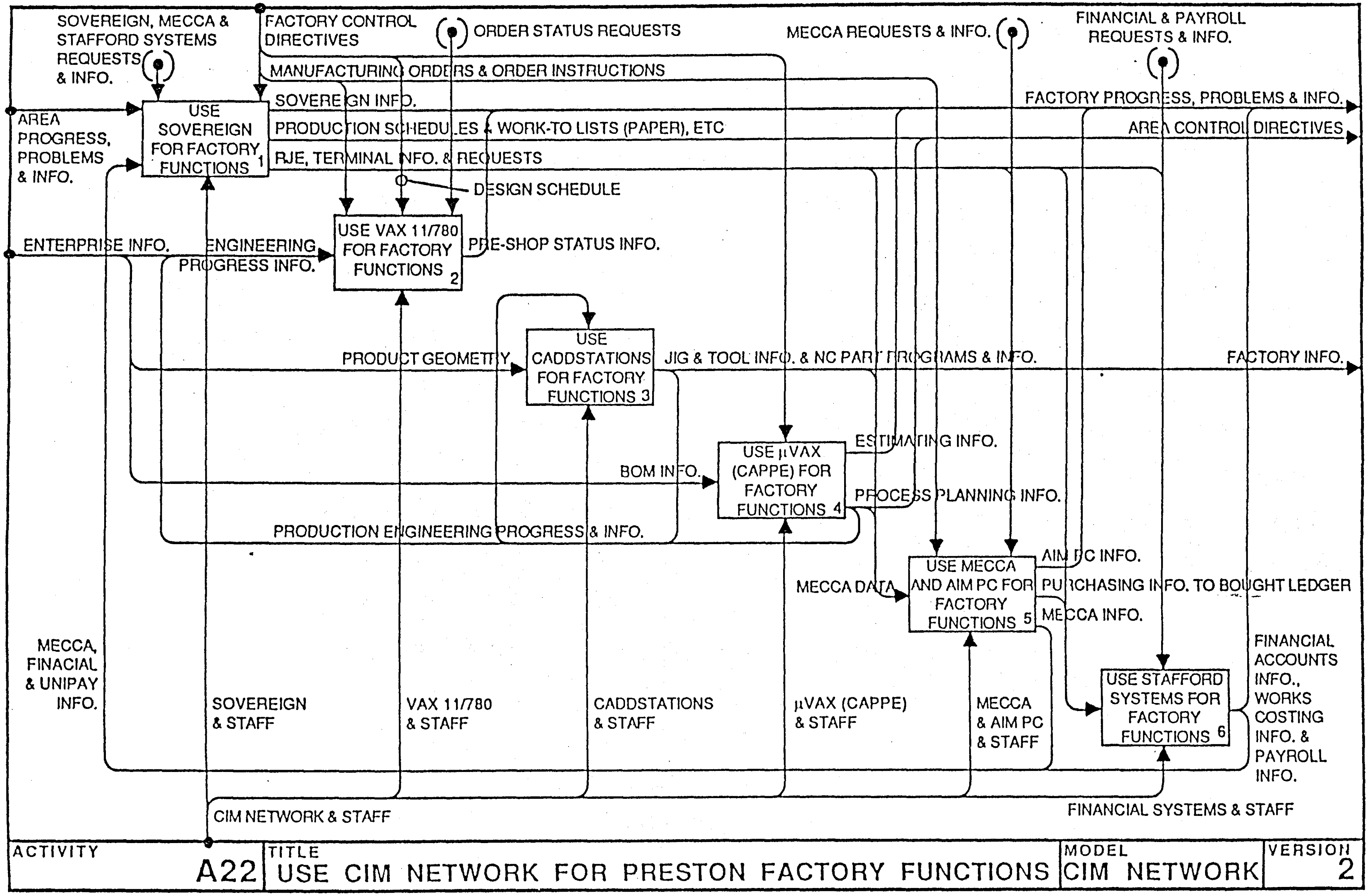

DATABASE ENQUIRIES
SCHEDULE ENQUIRIES
NEW SCHEDULE DATA
NEW DATABASE DATA
REQUESTS FOR PDM
OTHER DATA BASE ENQUIRIES
SCHEDULES
DATABASE INFO.
SCHEDULE INFO.
USE SCHEDULES DATABASE
1
USE TYPE AND OP DATABASES
2
TYPE & OP DATABASE INFO.
TYPE & OP INFO. FOR SCHEDULES
USE PDM
3
DATA TO PDM
DATA FROM PDM
USE OTHER DATABASES AND APPLICATIONS
4
OTHER DATABASE INFO.
SCHEDULE DATABASE & ENGINEERING STAFF
IBM 4381 & STAFF
TYPE & OP DATABASES & STAFF
PDM & D.O. STAFF
OTHER DATABASES & APPLICATIONS & STAFF
ACTIVITY
A13
TITLE
USE IBM 4381 FOR ENTERPRISE FUNCTIONS
MODEL
CIM NETWORK
VERSION
1

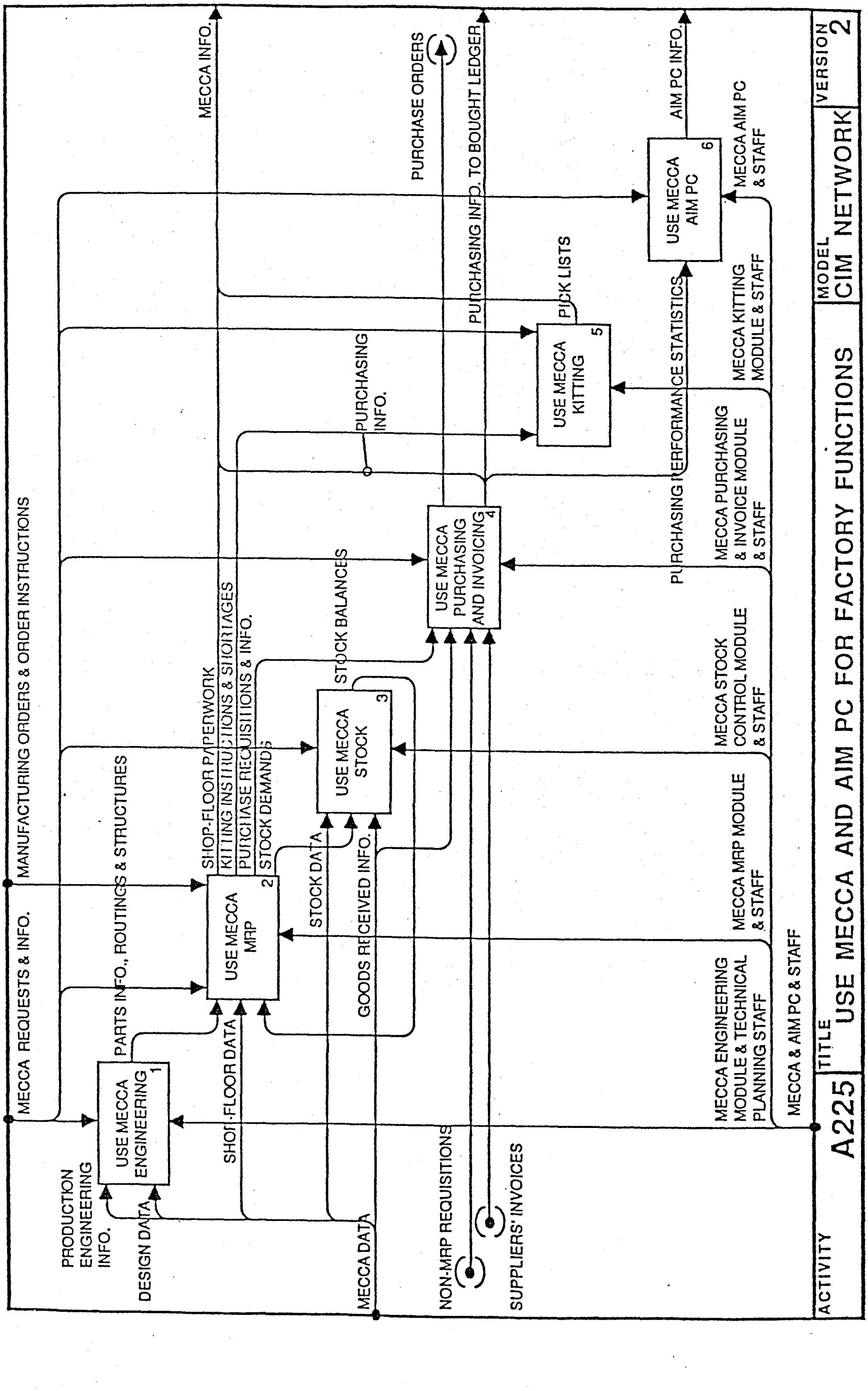


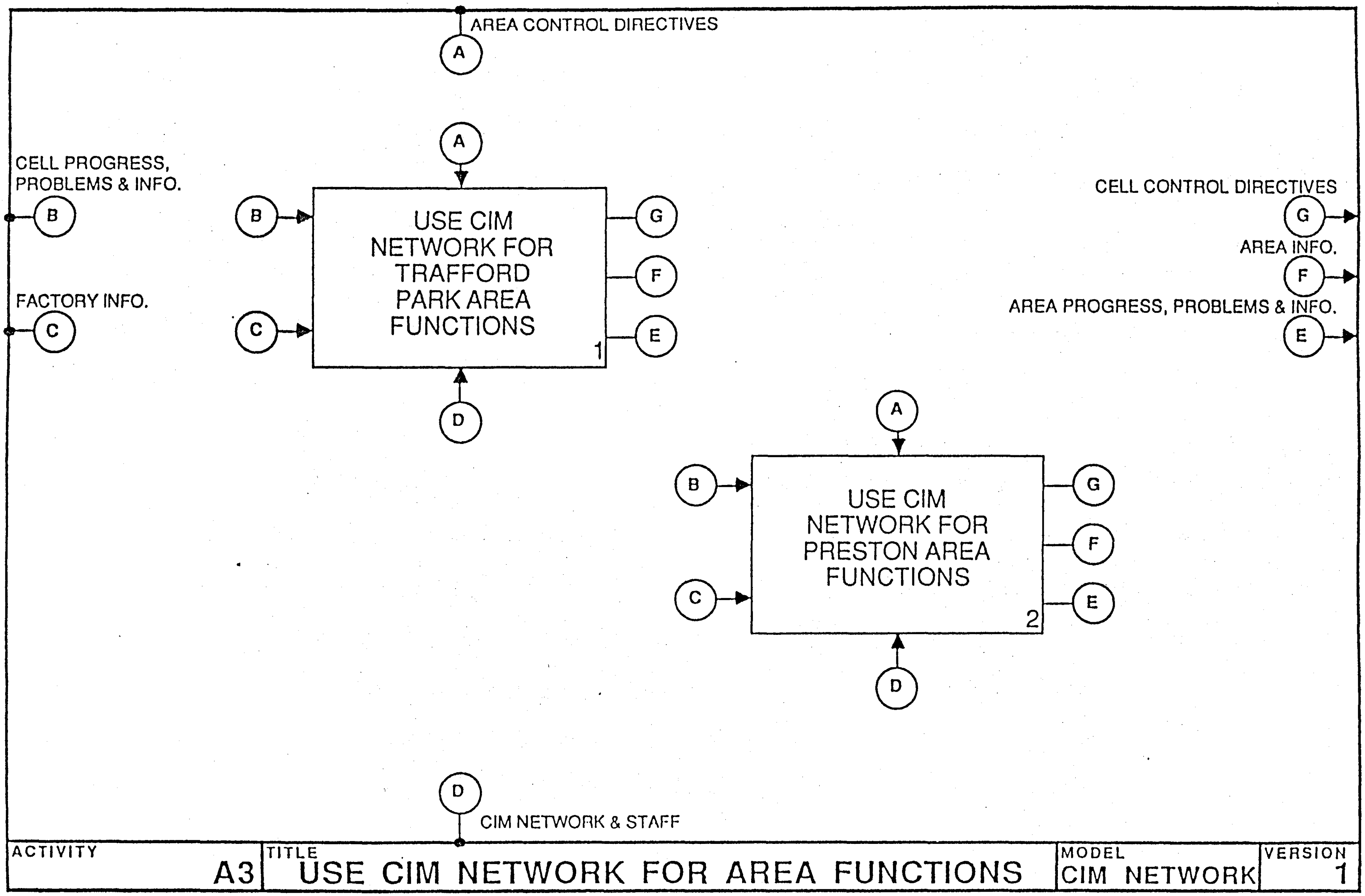
ACTIVITY	TITLE	MODEL	VERSION
A2	USE CIM NETWORK FOR FACTORY FUNCTIONS	CIM NETWORK	2

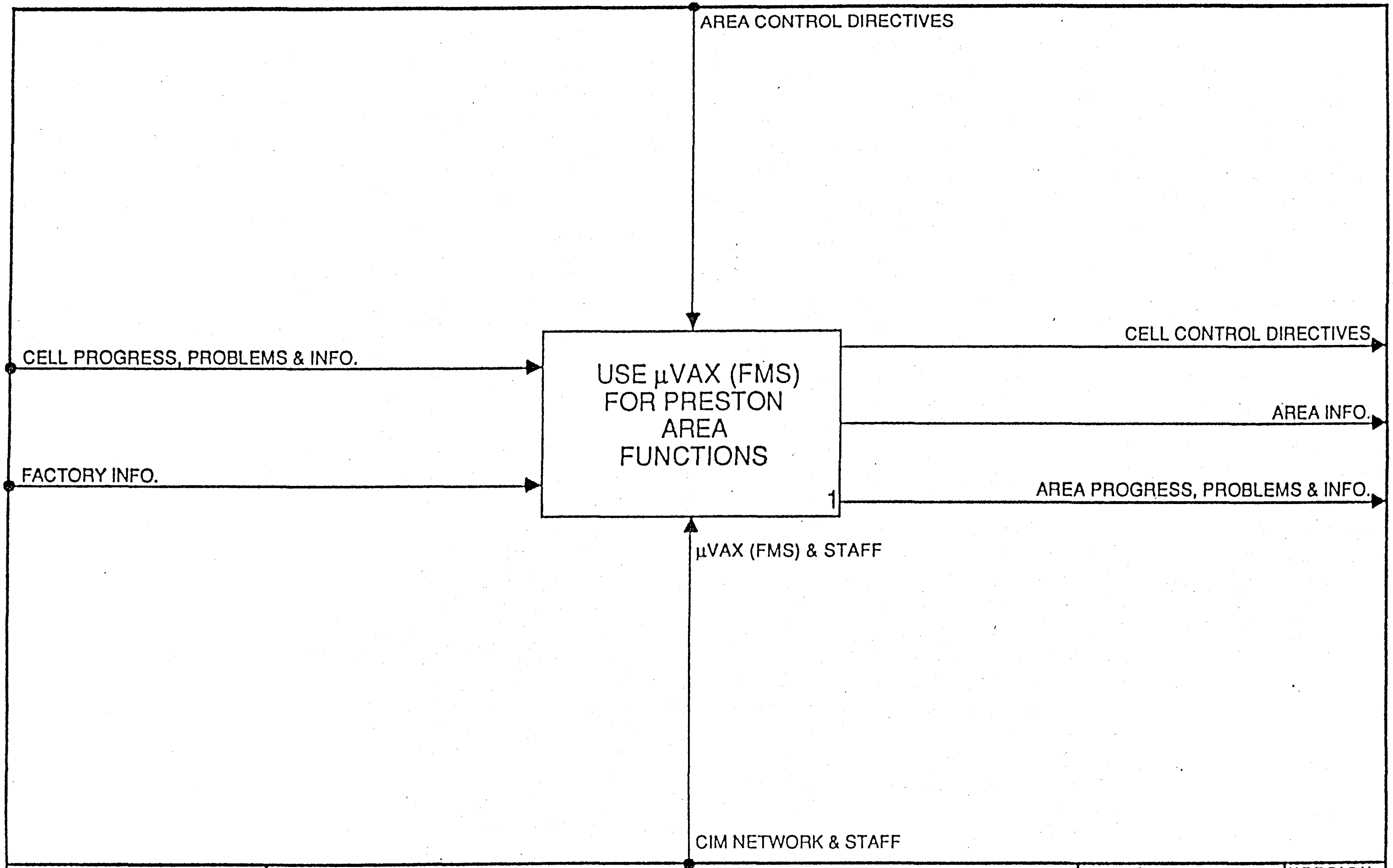

FACTORY CONTROL DIRECTIVES
SOVEREIGN REQUESTS & INFO.
MANUFACTURING ORDERS & ORDER INSTRUCTIONS
ESTIMATING ENQUIRIES
MECCA REQUESTS & INFO.
USE SOVEREIGN FOR FACTORY FUNCTIONS 1
SOVEREIGN INFO.
FACTORY INFO.
FINANCIAL & UNIPAY REQUESTS
SOVEREIGN INFO. TO MECCA
ENTERPRISE INFO.
ESTIMATING DATA
USE VAX 11/780 FOR FACTORY FUNCTIONS 2
ESTIMATING INFO.
FACTORY PROGRESS, PROBLEMS & INFO.
AREA PROGRESS, PROBLEMS & INFO.
MECCA DATA
USE MECCA FOR FACTORY FUNCTIONS 3
MECCA INFO.
AREA CONTROL DIRECTIVES
PURCHASING INFO.
FINANCIAL & MECCA INFO. TO SOVEREIGN
USE STAFFORD SYSTEMS FOR FACTORY FUNCTIONS 4
FINANCIAL INFO.
CIM NETWORK & STAFF
ACTIVITY
A21
TITLE
USE CIM NETWORK FOR TRAFFORD PARK FACTORY FUNCTIONS
MODEL
CIM NETWORK
VERSION
1


MECCA REQUESTS & INFO.
MANUFACTURING ORDERS & ORDER INSTRCUTIONS
PRODUCTION ENGINEERING INFO.
DESIGN DATA
MECCA DATA
USE MECCA ENGINEERING
1
PARTS INFO., ROUTINGS AND STRUCTURES
SHOP-FLOOR DATA
USE MECCA MRP
2
SHOP-FLOOR PAPERWORK
KITTING INSTRUCTIONS & SHORTAGES
PURCHASE REQUISITIONS
STOCK DEMANDS
STOCK DATA
GOODS RECEIVED INFO.
USE MECCA STOCK
3
STOCK BALANCES
USE MECCA PURCHASING
4
MECCA INFO
PURCHASING INFO
PURCHASE ORDERS
USE MECCA KITTING
5
PICK LISTS
MECCA KITTING MODULE & STAFF
MECCA PURCHASING MODULE & STAFF
MECCA STOCK MODULE & STAFF
MECCA MRP MODULE & STAFF
MECCA ENGINEERING MODULE & TECHNICAL PLANNING STAFF
MECCA & STAFF
ACTIVITY
A213
TITLE
USE MECCA FOR FACTORY FUNCTIONS
MODEL
CIM NETWORK
VERSION
1

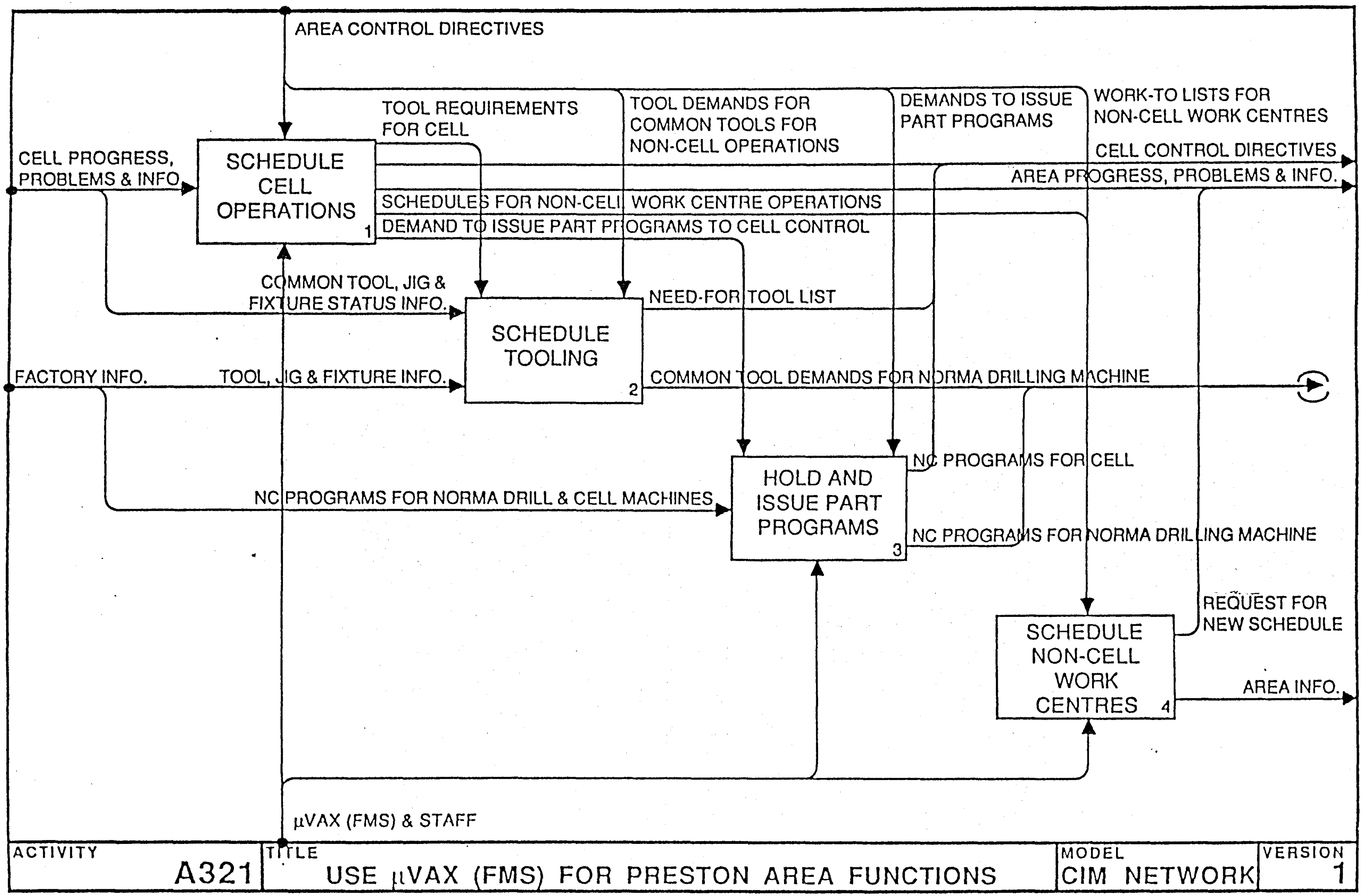

SOVEREIGN, MECCA & STAFFORD SYSTEMS REQUESTS & INFO.
FACTORY CONTROL DIRECTIVES
ORDER STATUS REQUESTS
MECCA REQUESTS & INFO.
FINANCIAL & PAYROLL REQUESTS & INFO.
MANUFACTURING ORDERS & ORDER INSTRUCTIONS
SOVEREIGN INFO.
FACTORY PROGRESS, PROBLEMS & INFO.
PRODUCTION SCHEDULES & WORK-TO LISTS (PAPER), ETC
AREA CONTROL DIRECTIVES
RJE, TERMINAL INFO. & REQUESTS
AREA PROGRESS, PROBLEMS & INFO.
USE SOVEREIGN FOR FACTORY FUNCTIONS 1
DESIGN SCHEDULE
ENTERPRISE INFO.
ENGINEERING PROGRESS INFO.
USE VAX 11/780 FOR FACTORY FUNCTIONS 2
PRE-SHOP STATUS INFO.
PRODUCT GEOMETRY
USE CADDSTATIONS FOR FACTORY FUNCTIONS 3
JIG & TOOL INFO. & NC PART PROGRAMS & INFO.
FACTORY INFO.
ESTIMATING INFO.
BOM INFO.
USE µVAX (CAPPE) FOR FACTORY FUNCTIONS 4
PROCESS PLANNING INFO.
PRODUCTION ENGINEERING PROGRESS & INFO.
AIM PC INFO.
MECCA DATA
USE MECCA AND AIM PC FOR FACTORY FUNCTIONS 5
PURCHASING INFO. TO BOUGHT LEDGER
MECCA INFO.
MECCA, FINACIAL & UNIPAY INFO.
FINANCIAL ACCOUNTS INFO., WORKS COSTING INFO. & PAYROLL INFO.
USE STAFFORD SYSTEMS FOR FACTORY FUNCTIONS 6
SOVEREIGN & STAFF
VAX 11/780 & STAFF
CADDSTATIONS & STAFF
µVAX (CAPPE) & STAFF
MECCA & AIM PC & STAFF
CIM NETWORK & STAFF
FINANCIAL SYSTEMS & STAFF
ACTIVITY
A22
TITLE
USE CIM NETWORK FOR PRESTON FACTORY FUNCTIONS
MODEL
CIM NETWORK
VERSION
2

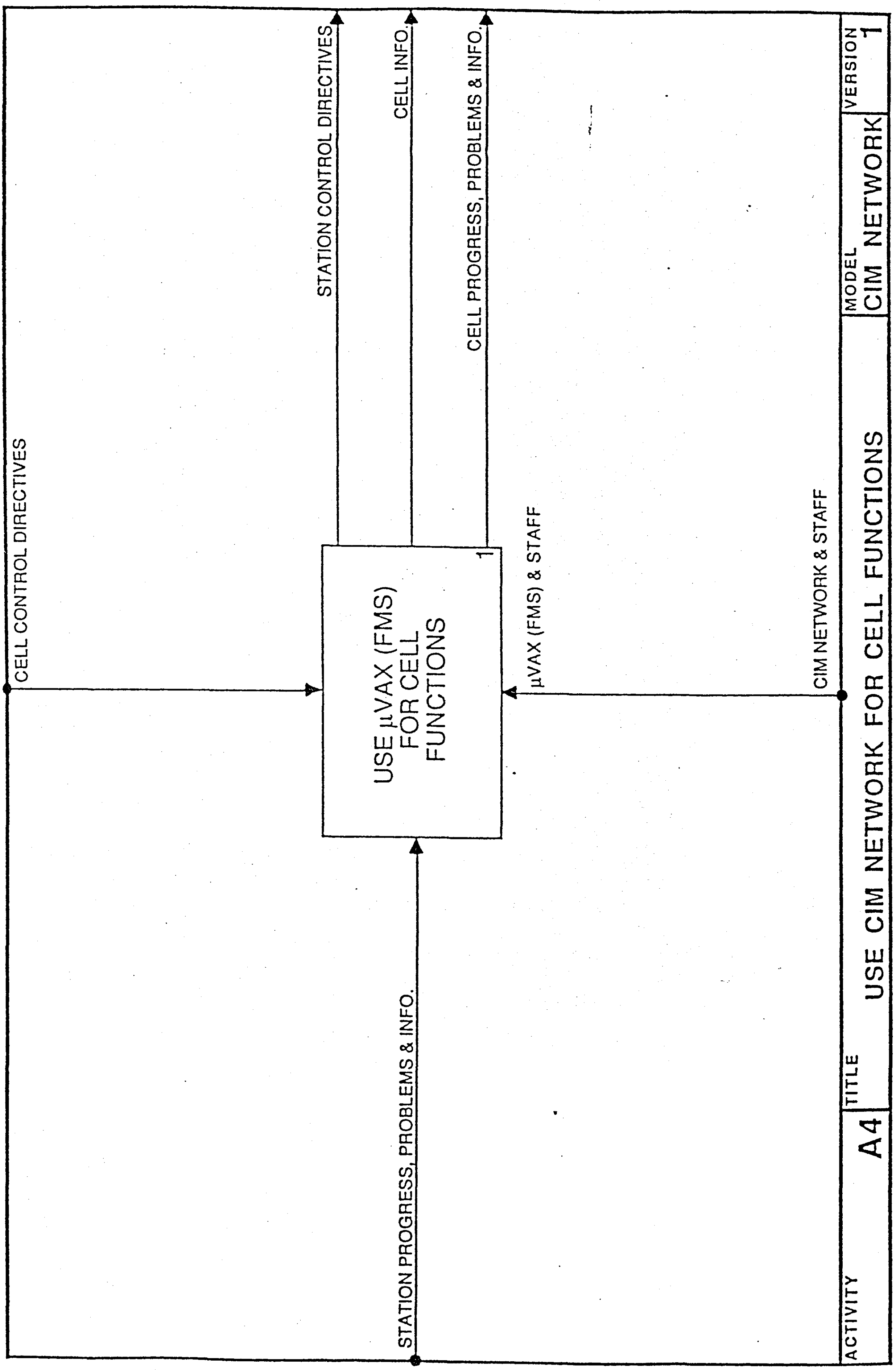

MECCA REQUESTS & INFO.
MANUFACTURING ORDERS & ORDER INSTRUCTIONS
PRODUCTION ENGINEERING INFO.
DESIGN DATA
MECCA DATA
NON-MRP REQUISITIONS
SUPPLIERS' INVOICES
USE MECCA ENGINEERING 1
PARTS INFO., ROUTINGS & STRUCTURES
SHOP-FLOOR DATA
USE MECCA MRP 2
SHOP-FLOOR PAPERWORK
KITTING INSTRUCTIONS & SHORTAGES
PURCHASE REQUISITIONS & INFO.
STOCK DEMANDS
STOCK DATA
GOODS RECEIVED INFO.
USE MECCA STOCK 3
STOCK BALANCES
USE MECCA PURCHASING AND INVOICING 4
PURCHASING INFO.
USE MECCA KITTING 5
PICK LISTS
MECCA INFO.
PURCHASE ORDERS
PURCHASING INFO. TO BOUGHT LEDGER
PURCHASING PERFORMANCE STATISTICS
USE MECCA AIM PC 6
AIM PC INFO.
MECCA ENGINEERING MODULE & TECHNICAL PLANNING STAFF
MECCA & AIM PC & STAFF
MECCA MRP MODULE & STAFF
MECCA STOCK CONTROL MODULE & STAFF
MECCA PURCHASING & INVOICE MODULE & STAFF
MECCA KITTING MODULE & STAFF
MECCA AIM PC & STAFF
ACTIVITY
A225
TITLE
USE MECCA AND AIM PC FOR FACTORY FUNCTIONS
MODEL
CIM NETWORK
VERSION
2

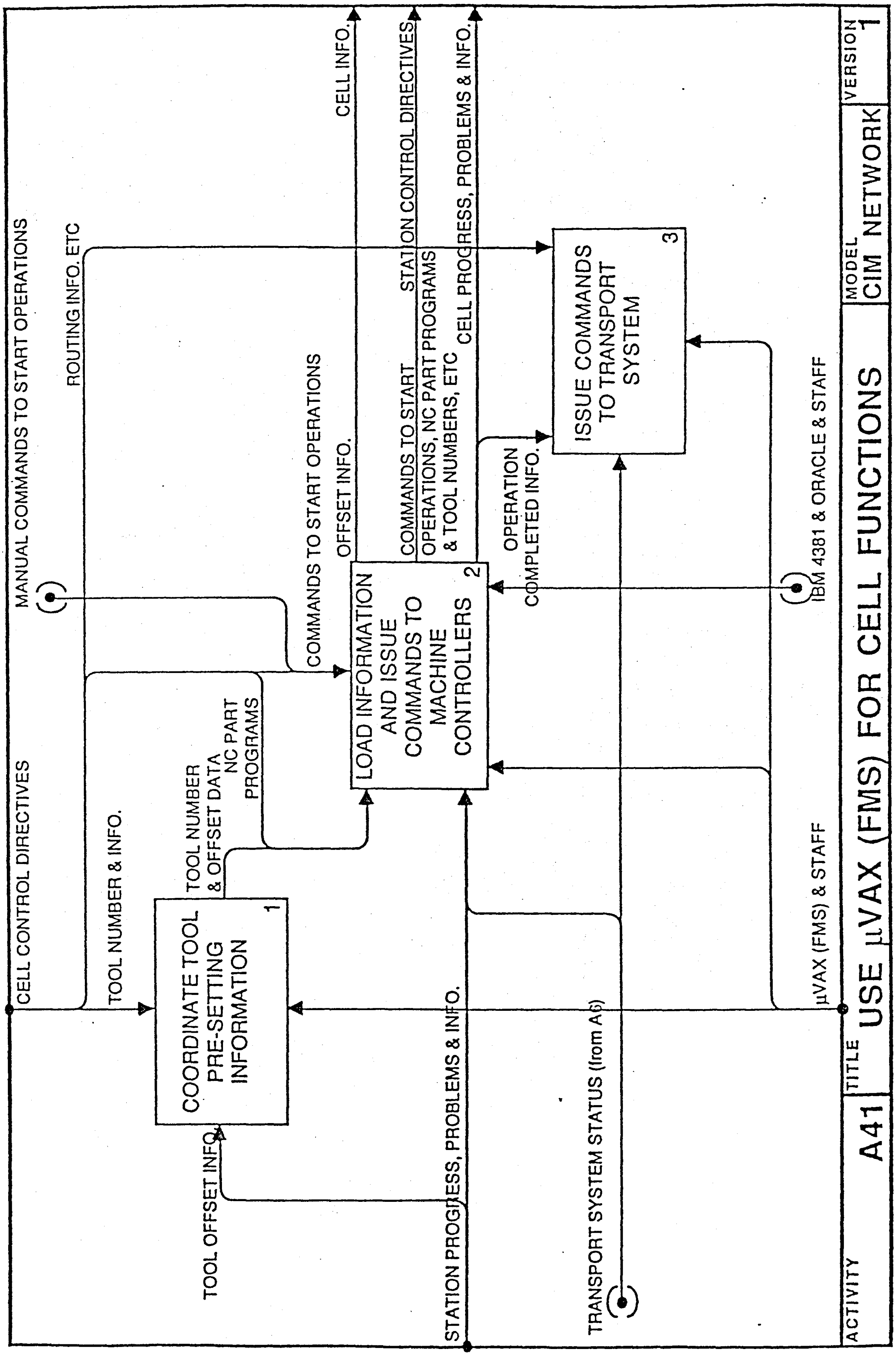

AREA CONTROL DIRECTIVES
A
CELL PROGRESS, PROBLEMS & INFO.
B
FACTORY INFO.
C
A
B
C
USE CIM NETWORK FOR TRAFFORD PARK AREA FUNCTIONS
1
D
G
F
E
CELL CONTROL DIRECTIVES
G
AREA INFO.
F
AREA PROGRESS, PROBLEMS & INFO.
E
A
B
C
USE CIM NETWORK FOR PRESTON AREA FUNCTIONS
2
D
G
F
E
D
CIM NETWORK & STAFF
ACTIVITY
A3
TITLE
USE CIM NETWORK FOR AREA FUNCTIONS
MODEL
CIM NETWORK
VERSION
1

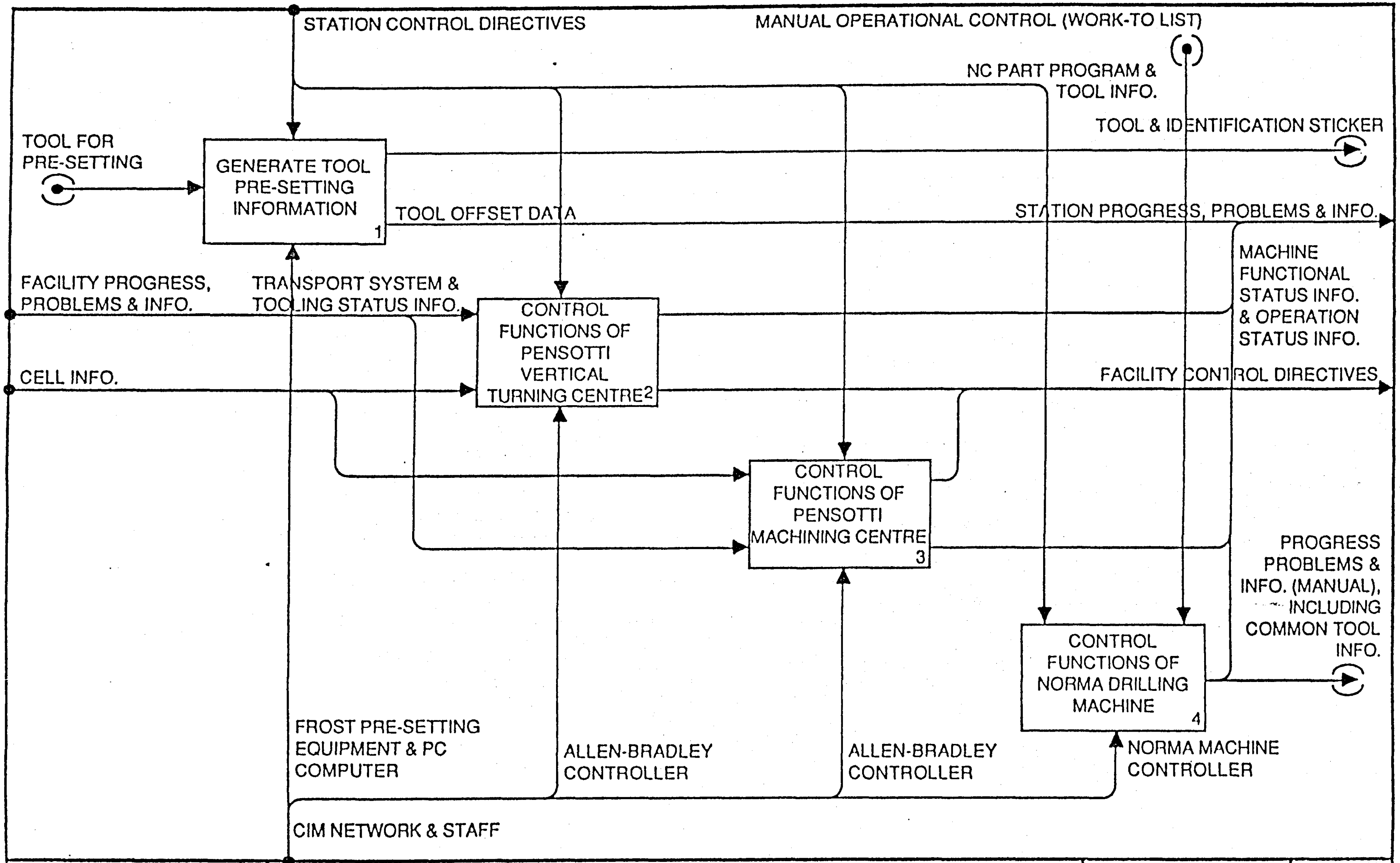


ACTIVITY	TITLE	MODEL	VERSION
A32	USE CIM NETWORK FOR PRESTON AREA FUNCTIONS	CIM NETWORK	1

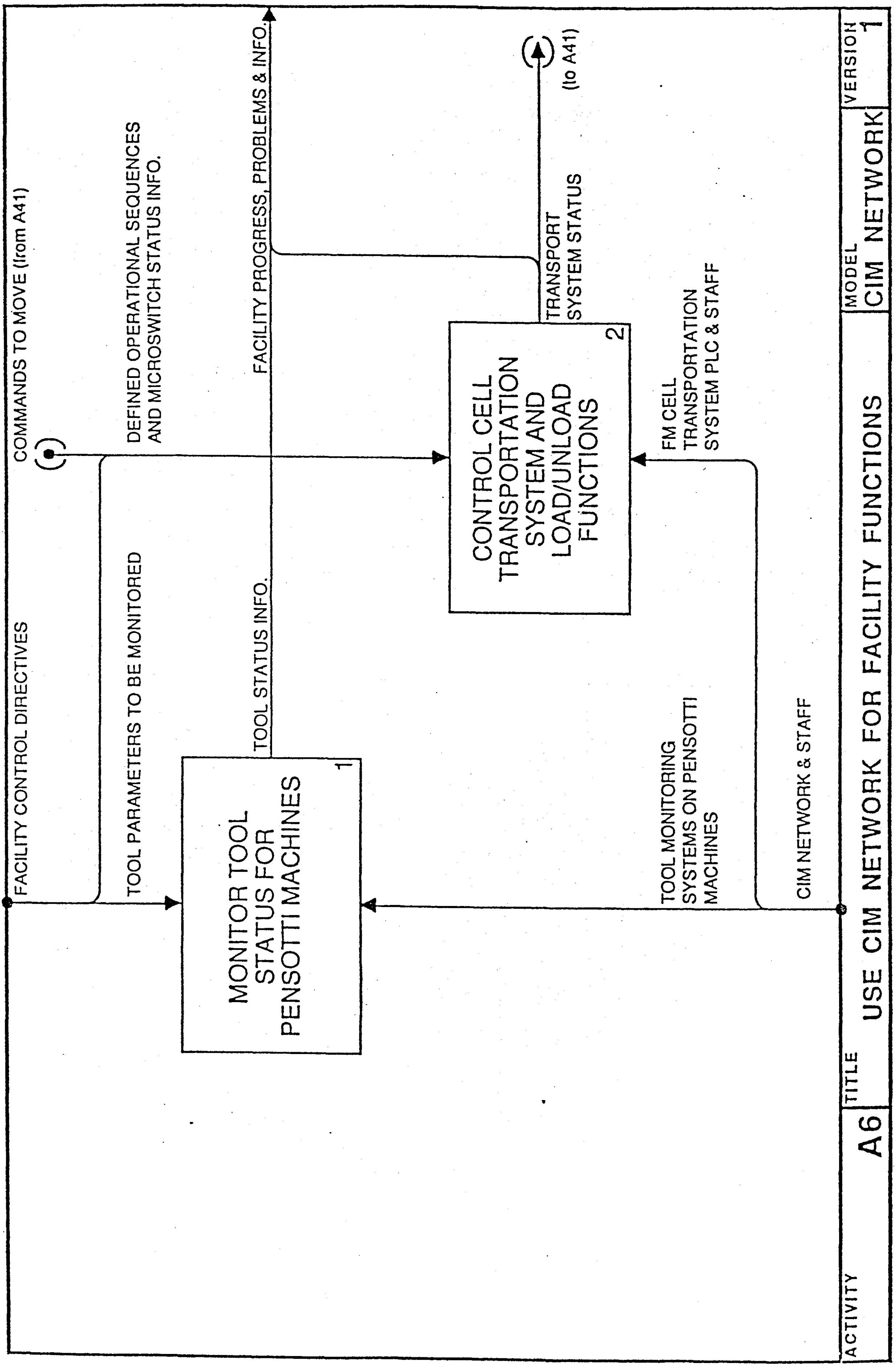

AREA CONTROL DIRECTIVES
CELL PROGRESS, PROBLEMS & INFO.
SCHEDULE CELL OPERATIONS
1
TOOL REQUIREMENTS FOR CELL
TOOL DEMANDS FOR COMMON TOOLS FOR NON-CELL OPERATIONS
DEMANDS TO ISSUE PART PROGRAMS
WORK-TO LISTS FOR NON-CELL WORK CENTRES
CELL CONTROL DIRECTIVES
AREA PROGRESS, PROBLEMS & INFO.
SCHEDULES FOR NON-CELL WORK CENTRE OPERATIONS
DEMAND TO ISSUE PART PROGRAMS TO CELL CONTROL
COMMON TOOL, JIG & FIXTURE STATUS INFO.
SCHEDULE TOOLING
2
NEED-FOR TOOL LIST
FACTORY INFO.
TOOL, JIG & FIXTURE INFO.
COMMON TOOL DEMANDS FOR NORMA DRILLING MACHINE
HOLD AND ISSUE PART PROGRAMS
3
NC PROGRAMS FOR CELL
NC PROGRAMS FOR NORMA DRILL & CELL MACHINES
NC PROGRAMS FOR NORMA DRILLING MACHINE
REQUEST FOR NEW SCHEDULE
SCHEDULE NON-CELL WORK CENTRES
4
AREA INFO.
μVAX (FMS) & STAFF
ACTIVITY
A321
TITLE
USE μVAX (FMS) FOR PRESTON AREA FUNCTIONS
MODEL
CIM NETWORK
VERSION
1


CELL CONTROL DIRECTIVES
STATION CONTROL DIRECTIVES
CELL INFO.
CELL PROGRESS, PROBLEMS & INFO.
USE µVAX (FMS) FOR CELL FUNCTIONS
1
µVAX (FMS) & STAFF
CIM NETWORK & STAFF
STATION PROGRESS, PROBLEMS & INFO.
ACTIVITY
A4
TITLE
USE CIM NETWORK FOR CELL FUNCTIONS
MODEL
CIM NETWORK
VERSION
1


CELL CONTROL DIRECTIVES
MANUAL COMMANDS TO START OPERATIONS
ROUTING INFO. ETC
TOOL NUMBER & INFO.
TOOL NUMBER & OFFSET DATA
NC PART PROGRAMS
COMMANDS TO START OPERATIONS
OFFSET INFO.
CELL INFO.
COMMANDS TO START OPERATIONS, NC PART PROGRAMS & TOOL NUMBERS, ETC
STATION CONTROL DIRECTIVES
CELL PROGRESS, PROBLEMS & INFO.
OPERATION COMPLETED INFO.
COORDINATE TOOL PRE-SETTING INFORMATION
1
LOAD INFORMATION AND ISSUE COMMANDS TO MACHINE CONTROLLERS
2
ISSUE COMMANDS TO TRANSPORT SYSTEM
3
TOOL OFFSET INFO.
STATION PROGRESS, PROBLEMS & INFO.
TRANSPORT SYSTEM STATUS (from A6)
μVAX (FMS) & STAFF
IBM 4381 & ORACLE & STAFF
ACTIVITY
A41
TITLE
USE μVAX (FMS) FOR CELL FUNCTIONS
MODEL
CIM NETWORK
VERSION
1



ACTIVITY	TITLE	MODEL	VERSION
A5	USE CIM NETWORK FOR STATION FUNCTIONS	CIM NETWORK	1


FACILITY CONTROL DIRECTIVES
COMMANDS TO MOVE (from A41)
TOOL PARAMETERS TO BE MONITORED
DEFINED OPERATIONAL SEQUENCES AND MICROSWITCH STATUS INFO.
TOOL STATUS INFO.
FACILITY PROGRESS, PROBLEMS & INFO.
MONITOR TOOL STATUS FOR PENSOTTI MACHINES
1
CONTROL CELL TRANSPORTATION SYSTEM AND LOAD/UNLOAD FUNCTIONS
2
TRANSPORT SYSTEM STATUS
(to A41)
TOOL MONITORING SYSTEMS ON PENSOTTI MACHINES
FM CELL TRANSPORTATION SYSTEM PLC & STAFF
CIM NETWORK & STAFF
ACTIVITY
A6
TITLE
USE CIM NETWORK FOR FACILITY FUNCTIONS
MODEL
CIM NETWORK
VERSION
1

Appendix V

A Top-Level Identification Check-list for Managers

(Taken from company document)

1. Overview

The purpose of this check-list is to provided an aid to managers or functional heads, with little understanding of expert systems, to identify for themselves possible areas where expert systems may be applied to their area of the organisation. It is not intended as a rigourous and definitive method of evaluation, but rather as a guide-line which will provide a useful starting point for further discussions with staff in the computer department.

2. Guide-lines on Answering the Questions

The questions asked are intended to help the manager identify areas where expert systems have been successfully applied in the past. The questions are as broad in scope as the manager wishes and may be address a particular task, undertaken by himself or others, or concern the functioning of the department as a whole.

The questions look at the distribution of expertise and the nature of problems in the company. 'Expertise' in this context refers to any person who has a good understanding of a particular field or is adept in a specific practice. Clearly, many experts are needed in the department, but some are in greater demand or are more scarce than others and it is here, particularly, where expert systems may be of benefit.

You are encouraged to cite specific examples in response to the questions, where this is possible. Do not worry if examples are repeated for different questions. You are also advised to read the 'Notes' which accompany the check-list and explain the reasoning behind each of the questions. Further information on expert systems is available from the Computer Department

Part A: Identifying Expertise

Are there areas in your department (or area of responsibility):-

1. Where skilled staff (i.e. 'experts') are in short supply or have not enough time to see all those who require his or her skills ?

2. Where there are difficulties in training staff through the nature of the task itself of because no one is available to train them ?

3. Where routine problem solving and other tasks tie-up skilled staff preventing them from doing their specialised work ?

4. Where staff will be leaving soon leaving a 'skills-gap' ?

5. Where too many departmental operations depend upon a single person ?

6. Where the response to requests and queries from skilled staff is not quick enough ?

7. Where specialised staff must understand processes that are outside their field ?

8. Where 'expertise' is necessary in a specialised area, but required infrequently ?

9. Where skilled staff performance has a high level of variance ?

10. Where there are a number of experts who do the job differently ?

11. Where it would be useful to archive skills and develop a 'knowledge-bank' of experiences.

Part B: Identifying Appropriate Problems

Are there tasks or activities in your department: -

1. Where judgements are made rather than precise and absolute decisions ?

2. Where it is difficult to find solutions using conventional programming methods ?

3. Where no exact solution is possible within 'reasonable' resource constraints ?

4. Where it is difficult to explain procedures and working practices ?

5. Where there are bottle-necks in working practices ?

6. Which require consistent and high quality evaluation and monitoring ?

7. Where it would be desirable to reduce the uncertainty of a decision-making process ?

8. Where it would be desirable to provide a standard approach to solving fairly unstructured problems?

Part C: Information Structure and Requirements ?

Are there areas in your department where :-

1. Information is partial, contradictory or unreliable ?
2. Large amounts of information are required in order to make simple decisions ?
3. Information is of no use because it is presented in the wrong form or is inaccessible ?
4. Information gathering takes up too much of the expert's time ?
5. Insufficient information is available to make the correct decisions ?
6. It is important to distribute information rapidly ?
7. Information, which could be utilized with effect, is not being used ?
8. It would be desirable to interface with a number of information systems ?
9. It is necessary to disseminate large amounts of information ?

Part D: Notes to the Check-List

A1. By representing the skills and knowledge of an expert, in an expert system, it becomes possible to share this knowledge at different levels of enquiry among other company personnel

A2. Expert systems make effective training systems. They can provide advisory, explanation and query facilities; allow evaluation and self-assessment; and can provide a 'front-end' high level interface to more complex computing facilities, thus improving their accessibility to non-specialist personnel.

A3. Experts are a costly resource and yet much of their time is spent in routine decision-making and paperwork. These tasks, with the aid of an expert system, could be handed down to less qualified personnel so that the expert can concentrate on skills commensurate with his or her skills.

A4. A company often fails to acknowledge the importance of a person's skills until that person has left the organisation. An engineer, for instance, may have many years of accumulated knowledge and experience in a particular field which will be very difficult to replace. Expert systems can help by archiving this knowledge and so retaining it within the company once the engineer has left.

A5. An over-reliance upon a single individual is dangerous for the above reason, but may also result in a bottle-neck of expertise inaccessible to the rest of the organisation. Expert systems should be considered here because they offer a potential to record and distribute this knowledge.

A6. There are a number of reasons why an expert system is unable to respond to requests or queries in time; from work overload to unavailability of the right information required to make a decision. Expert Systems can help to improve the productivity of decision making through the use of rapid searching techniques, standard decision making and accurate interpretation of information.

A7. Specialised staff have an excellent understanding of a specific field but may be less aware of external factors which nevertheless impose, or have an effect, upon their work. Expert systems provide useful access to these secondary levels of expertise in a format which is both understandable and usable.

A8. Occasional reference to expertise may be costly and wasteful of human resources. Expert systems may provide an effective platform by which to record this 'occasional' knowledge.

A9. Experts, as human beings, vary in mood and temperament; and the environment in which they work is equally unpredictable. Expert systems have been used to verify, check for accuracy, relevance and consistency of decisions.

A10. Different experts in the same field may have different strategies for solving problems and yet arrive at the same solution. This can be confusing and difficult to verify, and so it is important that a single approach is used. An expert system can help to enforce standards or conventions based upon an agreed approach or compromise.

A11. A knowledge-bank is a repository for the expertise of skilled people in the company. It can be used to archive knowledge, but also as a means of pooling together different types of expertise in order to solve a higher level of problem.

B1. Human expertise is characterised by its use of 'rules-of-thumb' as well as facts. Rules-of-thumb' or heuristics are informal judgements based on a feel for what the solution is, gained through years of experience. Often such judgements are made in areas where there is uncertainty or lack of information.

B2. Conventional programming techniques make use of algorithms which define precisely what the computer is to do and how it is to·do the computation. It requires that both the process method and data are known and quantifiable. By contrast, human judgements make qualitative decisions on the basis of experience and know-how and therefore require a different programming approach.

B3. Since conventional programs make use of precise data, it is possible to arrive at optimal solutions- scientific and engineering calculations for example arrive at correct and one-time solutions. However, human decisions are often made on the basis of attaining a satisfactory solution, rather than being optimal, based upon a number of parameter constraints.

B4. Introducing a new computer system may require re-training, tuition and explanation of new concepts and procedures. It may take up to two years before the trainee becomes fully useful to the company depending upon his or her ability and familiarity with the system. An expert system can allow variable rates of learning, and through interactive and self-help facilities, can accelerate the learning process itself.

B5. Bottle-necks in work practices may be as a result of erroneous, unavailable, untimely or excessive amounts of information, or a lack of expertise to interpret and process it. Expert systems have been used in all of these areas.

B6. Expert systems can be used for on-line evaluation and monitoring using a mixture of judgemental and probablistic decisions with factual and real-time information.

B7. Expert systems do not require perfect inputs and outputs in order to be able to make decisions. By use of various techniques, the technology is able to accommodate uncertainty in the reasoning process, and equally, provide answers with *levels* of correctness.

B8. The more unstructured a problem, the greater the likelihood of error and variance in the way the problem is resolved. Expert systems have been used to add consistency and uniformity to the decision-making process.

C1. By defining the necessary relationships between facts and data, and with the ability to handle uncertainty given incomplete data, expert systems are able to elicit valid information for contradictory or unreliable data through various techniques.

C2. By defining rule constraints in an expert system connected to a database, it is possible to undertake rapid searches using the principle of 'heuristic reduction' whereby the expert system directs the search to that part of the database which satisfy top-level constraints.

C3. A system which takes more than five minutes to diagnose a fault is of little use in a real-time environment. Similarly, a printout of data has little value to a senior manager. It is thus important that information is targeted and presented to the right person at the right time.·Expert systems have been used as interfaces and filters so that this is achievable.

C4. Effective decision-making requires availability of information from which to make a decision. However, a disproportionate amount of time should not be spent in gathering this information. expert systems have proved themselves useful in locating presenting information in an appropriate format for use.

C5. A measure of expertise is the ability to recall what was done on previous occasions or judge intuitively what is required and be able to enact this knowledge in practice.

C6. Expert systems have been shown to be able to improve the speed and accuracy by which information may be distributed and processed.

C7. Expert systems have been used to improve the 'transparency' of information systems through interfacing and help facilities.

C8. The particular structure of expert systems allows the mechanism by which problems are solved to be separated from the knowledge source. Consequently, the knowledge source may originate form a number of different physical locations- databases, CAD, etc. This allows the expert system to rest upon a number of information systems and extract information and data when and where appropriate. It also means that it is possible to maintain and update these information systems without affecting the integrity of existing information.

C9. The adoption of heuristic search techniques allows expert systems to retrieve, breakdown and distribute relevant information from a larger source such as a database.

Appendix VI

A Listing and Classification of Potential Expert System Applications in the Client Organisation

Contents:

1. Classification of Potential Applications by Organisational Function

This section defines functional areas of the business where expert systems may be applied. For each application specific to a function, an indication of project length, resource needs, costs and impact is given.

Key to Classification Listings

Organisational Function- The division of applications by function corresponds to the functional decomposition represented in the IDEFo analysis. The applications suggested in each function have been chosen on the basis of a top-level evaluation; a more rigorous feasibility assessment proceeds this work.
Detail Number- The number refers to a description of the proposed system in the applications listings , Part 2.
Project Length- this is expressed in terms of development man years of effort. It does not include the involvement of the expert.
Resource Needs- this is covered in greater detail at later stages and will include an assessment of project requirements, development plans, implementation and post-implementation needs. However, as an indication of resource needs, four key criteria have been chosen:-

		KEY
a) **Hardware requirements:**	IBM Compatible p.c	**P.C.**
(Delivery Vehicle)	Workstation	**WS.**
	A.I. specific workstation	**A.I.WS**
	Mainframe	**MF.**
b) **Software requirements:**	A.I. Language	**L.**
(Primary Software)	Shell	**S.**
	Toolkit	**T.**
	Environment	**E.**
	Application Specific Software	**A.S.S.**
c) **Integration Requirements:**	Stand Alone	**SA**
	Embedded / Linked	**E**
	Fully Integrated	**F**

d) **Expert Involvement:** It is difficult to quantify precisely the level of commitment required by the expert, but it is possible to give it an estimate rating, ranging from no need, to intensive involvement by the expert.

None ....Moderate................................Intensive Involvement

0 1 2 3 4 5

Costs: Again costs are difficult to quantify. However, it is possible to make a relative cost estimate based upon resource needs and comparison with other similar projects. The following cost ratings are used:

Low Cost...............................Moderate..................................High Costs

0 1 2 3 4 5

Impact: Cost value alone is insufficient in justifying software however because it has a different meaning according to the purpose and need for the system in the organisation. For example, a system may be proposed which has significant strategic and long-term implications and yet may yield a high cost value. Similarly, a system developed as a demonstrator will yield a very high cost value with no visible economic benefit; and yet, it is an important part of the learning and development process. Thus, the following measures of impact and significance are used:

Demo	Systems	Operations	Tactical	Strategic
(for demonstration only	(unit efficiency)	(factory efficiency)	(Operations effectiveness)	(Business effectiveness)

Organisational Functions

Applications have been classified according to the structure defined by the IDEFo Analysis. The coding of functions is as follows:-

A. Front-End Services & Senior Management

1a	Traction Upper Management (Traction)
2a	Tendering (TPL)
3a	Sales/ Export (TPL)
4a	Tendering (TPL)
5a	Contracts (TPL)
6a	Commercial/ Spares (Traction)

B. Engineering

1b	Engineering Senior Management
2b	Engineering Planning
3b	Applications Engineering
4b	Systems Engineering
5b	Control Drawing Office (D.O.)
6b	Equipment D. O.
7b	Machine Engineering Department (Preston)
8b	Machines D. O. (Preston)

C. Preston Manufacturing

1c	Preston Senior Manufacturing Management
2c	Purchasing
3c	Manufacturing Services
4c	Pre Shop
5c	Industrial Engineering/ planning
6e	Manufacturing Operations
7c	Works Services

D. Trafford Park Manufacturing

1d	Trafford Park Senior Manufacturing Management
2d	Coordinate Manufacture- PP & PU
3d	P-R Purchasing
4d	Materials Control
5d	Industrial Engineering/ planning
6d	Manufacturing Operations
7d	Works Services

E. Support Services

1e	Estimating - Preston
2c	Estimating - Trafford Park
3c	Finance
4c	Quality Assurance
5c	Standards
6e	Personnel
7e	Computing- Trafford Park
8e	Computing- Preston.

F. All Areas

Categorisation of Application by Organisational Function

Page 1.

Organisational Function / Application	Appl'n Number	Project Length (Man Yrs.)	Resource Needs - hardware / software / integration / expert	Cost	Impact
1a. Upper Management					
Co. Finanial Management	56	1	P.C. / S / SA / 4	2	T. / St.
Documentation Reduction	82	3/4	MF. / L. / F / 2	5	O.
Intelligent Front-end	83	2 / 3	MF / L or ASS / F / 3	4	S / O
Project Layout	42	1	P.C. / S. / SA. / 3	2	T
Knowledge-Based Administration	62	1/2 - 5	P.C. / S & L. / E / 1	1-4	O
Industry Regulations	65	< 1	P.C. / S. / SA. / 1	1	O / T
Report Processing	90	2 - 3	MF / L / F / 2	5	O
Tuition / Training / Updates	110	1 - 2	P.C. / E -ASS/ F /4	3	T
Sales Forecasting	112	1 - 2	P.C. / ASS / E / 5	2 - 3	St.
Sales Advisor	116	2	P.C. / S / E / 3	3	St.
Tender Decisions	129	1	P.C. / S and T/ E / 4	2	St.
Commitment to Contract	131	< 1	P.C. / S / SA / 5	1	St.
Financial Pack Collator	132	3	MF. / T. / F ./ 3	1 - 2	Sy
Customer Feedback	133	5 - 6	MF / T / F / 5	5	St.
Srategic Planning	134	7 - 10	MF / E / F / 5	5	St.
2 a. Tendering (TPL)					
Codes of Practice	10	< 1	P C / S / SA / 0	0	O
Customer Order Engng.	16	4 - 7	WS / E / F / 3 - 5	5	T
Tendering Protocol	40	1 - 2	P.C. / S / SA / 5	1	Sy.
Clarifying Legislation	54	< 1	P.C. / S / SA / 0	0	O
Knowledge- based Administration	62	1/2 - 5	P C / S & L / E or F / 1	1 - 4	O
Bid Estimation Support	75	6	P.C. & MF / ASS / F/4	4 -5	T
Documentation Reduction	82	3 - 4	MF / L / F / 2	3	O

Organisational Function / Application	Appl'n Number	Project Length	Resource Needs - Man-years of work - Hardware / Software	Cost	Impact
Tender Proposals	98	7- 8	MF / L or E. / F / 5	5	T
Tender Decisions	129	1	P.C. / S & T / E / 4	2	Sy / O
Customer Specification	130	4 - 12	WS / E / E/ 3 -4	4	T
3a. Sales / Export (TPL)					
Sales Configurer	38	2	P.C. / L / E / 5	3	T
Clarifying Legislation	54	< 1	P.C. / S / S.A. / 1	0	Sy
Sales Forecasting	112	2 - 3	P.C. / S or T / E / 5	3	O
Promotion	124	< 1	P.C. / S / SA. / 1	1	Sy
Client Specification	130	2	P.C. / E / E / 3	2 - 3	T
4a. Contracts (TPL)					
Contract Handover	33	1	P.C. / S / SA / 3	1 - 2	O
Project Layout	42	1	P.C. / L / E / 4	3	O
Client Maintenance	67	2	WS / T / E / 4	2 - 3	St
Evaluate subcontractors	92	3 - 4	P.C. / S. / E / 5	4	T
Engng. Plannning	113	4	WS. / E / F / 4	5	T
Engineering Costing	114	2	P.C. / S / E / 3	2	Sy
Sales Advisor	116	2	P.C. / L / E / 2	5	T
Contract Capital	131	1	P.C. / S / SA. / 3	1 - 2	T
5a. Spares / Commercial					
Spares Configurer	123	1	P.C. / S / SA / 5	2	Sy
Parts Numbering	44	2	P.C. / S / SA / 1	1	Sy
Sales Advisor	116	1 -2	P.C. / S / E / 3	4	Sy
Customer Feedback	133	4	MF / T / F / 4	4	T

Business Function / Application	Appl'n Number	Project Length	Resource Needs	Cost	Impact
1b. Engng. Senior Management					
Contract Handover	33	1	P.C. / S / SA / 3	1 - 2	O
Make or Buy Advisor	26	2	P.C. / S / SA / 5	3	O
Expense Approval	135	< 1	P.C. / S / SA / 1	1	Sy
Capital Proposal	43	< 1	P.C. / S / SA / 2 - 3	2	T
Financial Management	56	1	P.C. / S / SA / 4	2	O
Sales Advisor	116	2	P.C. / S / E / 3	3	O
Supplier Vetting	106	2	P.C. / L / E / 2	4	T
Substitution	115	1 -2	PC-MF / L /SA-F /3	3	O
2b. Eng'ng Planning					
Planning	113	2	WS / E / F / 5	5	T
Expedition Control	98	2	MF / ASS or T / F / 4	5	T
Engineering (3b,4b,7b)					
Codes of Practice	10	1	P.C. / S / SA / 0	0	Sy
Intelligent Interfacing	83	2	MF / L / E / 5	5	Sy
Training Aid	110 / 66	2	MF / L / F / 5	4	Sy
PC Help Desk	71	1	PC / S / SA / 4	1	D/Sy
Rejection Analysis	15	1-2	P.C. / S & L / E / 2	3	Sy
Order Engineering	16	3	WS / L or ASS / F /3	3 - 5	O
Design Specification	23	1 -2	P.C. / S / E / 1	2	Sy
Capital Proposal Aid	43	< 1	P.C. / S / SA or E / 1 -2	2	T
Prototyping Diognostics	136	1	P.C. / S / AI / 3	4	Sy
Equipment Layout	41	1 - 2	P.C. / T / E / 4	3	O

Business Function / Application	Appl'n Number	Project Length	Resource Needs	Cost	Impact
Parts Numbering	44	2	P.C. / S / SA / 1	1	Sy
Electronic Connector	77	1	P.C. / L / SA / 1-2	2	Sy
Equipment Design Aid	88	3 - 4	AIWS / E / F/ 3	5	O
R . & D. Costing	137	1	P.C. / S / E / 5	1	O
Intelligent Simulation	138	3	WS / L / F / 2 -3	4 -5	Sy
Design for Manufacture	101	8 -10	WS & MF / L / F / 5	5+	St
PLC Selection	103	1	P.C. / S / SA / 4	1	Sy
Supplier Vetting	106	2	P.C. / S & L / E / 3	2- 3	T
CAD Support	109	5 - 6	WS / L / F / 5	5	T
Tape Selection	104	< 1	P.C. / S / SA / 3	1	Sy
Circuit Modelling	120	3	WS / L or T / E / 2	3	Sy
PCB Design	19	1- 2	PC / L / E / 2	2 -3	O
Wiring Configurer	38	1	PC / S / SA / 3 - 4	2	Sy
Draw'g Office (5,6,8b)					
Drawing Conventions	139	1	PC / S / SA / 0	1	Sy
Codes of Practice	10	1	P.C. / S / SA / 1	1	Sy
Training	12	1 -2	PC / T / E / 4	3	O
Senior Manufacturing Management (1c, 1d)					
Contract Handover	33	1	P.C. / S / SA / 3	1 - 2	O
Investment Advisor	43	1 - 2	P.C. / S / SA / 2 -4	2	T
Financial Management	56	1	P.C. / S / SA / 5	1 - 2	T
Intelligent Interface	83	2	MF. / S & L / E or F / 2	3	Sy/O
Evaluate Subcontracts	92	1	P.C. / S / SA / 5	1	Sy

Business Function / Application	Appl'n Number	Project Length	Resource Needs	Cost	Impact
Maintenance M'ment	100	4	WS / E / F / 4	5	T
Strategic Planning	134	6	MF / E / F/ 5	5	St.
Expense Approval	135	1	P.C. / S / SA / 2	1	Sy
R & D Costing	137	1	P.C. / S / E / 4	3	O
Training Advisor	121	1 -2	PC / S or L / SA /3	2-3	T
Clarifying Legislation	46 / 54	< 1	P.C. / S / SA / 1	1	Sy
Purchasing (2c, 3d)					
Trend / Dbase Analysis	13	1 - 2	P.C / L / E / 3	2	Sy/O
Reject ion Analysis	15	2	P.C. / L / E / 3	1	Sy
Purchasing Advisor	26	1 - 4	P.C. / L & S / F / 4	2 - 4	O / T
Clarifying Legislation	46 / 54	1	P.C / S / SA / 1	1	Sy
Order Entry	86	1	P.C. / T / E / 1	2	O
Supplier Vetting	106	1	P.C. / S / E / 3	3	T
Price Forecasting	107	1 - 2	P.C. / S & L / SA /4	2	O / T
Manufacturing Services (3c , 4c, 2d, 4d)					
Release Notes	14	< 1	P.C. / S / SA / 4	1	Sy
Rejection Analysis	15	1 - 2	P.C / S & L / E / 2	3	Sy
Shipping Planner	24	< 1	P.C. / S / SA / 3	1	Sy
Transport Scheduling	140	1	P.C. / T / SA / 2	2	Sy
Capacity Planning	25	1	P.C. / S / SA / 5	1	T
Materials Control	32	2	MF / L / F / 3	5	O
Job Shop Scheduling	36	2- 3	MF / L / F / 4	4	O
Capital Proposals	43	< 1	P.C. / S / SA / 3	1	T

Business Function / Application	Appl'n Number	Project Length	Resource Needs	Cost	Impact
Machine Monitoring	74 / 60 /51/ 27	1 -2	P.C. / S / E / 2	4	O
Equipment Calibration	79	1	P.C. / S / SA / 3	1	Sy
Stock Control	85	1	P.C. / S & L / E /1	2	Sy
Warehouse Control	105	2 - 3	WS / T or E / F / 2	3 - 4	O
Welding Processes	127/29 /50 /70	1 -2	P.C. / S / SA or E/4	2	Sy
R. & D. Costing	137	1	P.C. / S / E / 5	2	T
Industrial Engineering / Planning (5c , 5d)					
NC Programming	17	1	P.C. / T / F / 5	3	Sy
Machine Layouts	21	1	WS / ASS/ F/ 4	5	O
Process Planning	20 / 53 / 97	6 -7	AIWS / E / F/ 5	5+	T
Design Spec.	23	2	P.C. / S / E / 3	3	Sy
Tool Selection	31 / 89	1	WS / S / SA / 5	4	Sy
Tool Design	34	1	P.C. / S / SA / 5	2	Sy
Lubricant Selection	28	< 1	P.C. / S / SA / 4	1	D / Sy
NC Error Diagnosis	35	1	P.C. / L / E / 4	2 -3	Sy
Joint Advisor	30	< 1	P.C. / S / SA / 2	1	D/Sy
Connector Assembly	47	1- 2	WS / L / E / F / 3	4	Sy
Technical Specific'n.	48	2	WS / L / F / 1 - 2	3 -4	O
M/C Level Plann/ Sched.	36	3 -4	WS / T or E / F/3	5	O
Factory Level Plann/Sch.	78	5	WS / E / F / 4	5+	T
Strategic Plann./ Sched.	96	2	WS /E/ F / 5	3	St
Industry Regulations	65	1	P.C. / S / SA / 1	1	Sy
Prod. Engng. Aid	68	1 -3	P.C./ S /SA or E/ 3	1 -2	Sy

Business Function / Application	Appl'n Number	Project Length	Resource Needs	Cost	Impact
Jig / Fixture Design	111	1	P.C. / S / SA / 4	2	Sy
Process Simulation	138	3	WS / L / F / 3	3 - 5	Sy/O
Drawing Advisor	139	1	P.C. / S / SA / 1	1	Sy
Manufacturing Operations (6c ,6d)					
Defect Analysis	8/ 59	2	P.C. / S or L / E/3	1 - 2	Sy
Quality Control	11	1 -3	P.C/ S or T/ E / 4	2	Sy/O
Training in Quality	12	1	P.C./ S / SA / 5	1	O
Cutting Fluid Advisor	18	< 1	P.C. / S / SA / 4	1	D/Sy
Online M/C Diagnosis	49/108	1 - 2	PC-WS/ S / E/ 3-4	2 -3	Sy
Machine Monitoring	74 / 76	2	PC-WS /S -L/ E/4	2- 3	Sy
Machine Control	80 / 81	2	WS/ L / F / 5	3	Sy/O
PCB Test & Repair	73	2	WS / E / F / 5	3	O
T. & I. Selection	58	< 1	P.C. / S / SA / 1	1	Sy
Equipment Calibration	79	1	P.C. / S / SA / 3	1	Sy
Stock Control	85	1	P.C. / S / SA / 3	2	Sy
S / Floor Maintenance	87	1	P.C. / S / SA / 5	1	Sy
Quality Control	102	1 - 4	PC/S-E/SA-F/3	1-5	Sy-St
Inspection	125	3	WS-MF/ T/F/4	4	T
Development Costing	137	2	P.C / S / E / 5	1 - 2	T
Prototyping	136	1 - 2	WS/ L / E / 5	2 -3	O
Work Services (7c, 7d)					
Facility Layout T	21 / 99 /128	2	WS-MF/ S& L/F/ 5	4	
Capital Proposals	43	< 1	P.C. / S / SA / 3	1	Sy

Business Function / Application	Appl'n Number	Project Length	Resource Needs	Cost	Impact
Technical Specs.	48	1 - 2	P.C. / L / E /1 -2	2	Sy
Repair / Overhaul	122	1	P.C. / T / E / 4	2	Sy/St
Maintenance Design	22				
Maintenance Scheduling	64/78	2	WS / T / F / 3	3 -4	T
Maint 'nce Management	93/100	1	P.C. / S / E / 5	2	Sy
Maint'nce Stock Control	126	1	P.C. / S / SA / 3	1	Sy
Maintenance Simulation	138	3	WS / E / F / 4	4	T
Estimating (1e, 2e)					
Manufacturing Costs	52	2	WS / L / F / 4	5	O
Financial Management	56	1	P.C. / S / SA / 4	1	Sy
Bid Estimator	75	4	PC & MF/ ASS/ F/3	5	O
Intelligent Interfacing	83	2 - 3	MF / L / F / 3	2 -3	Sy
Subcontracting	92	1	P.C. / S / E / 5	1	T
Price Forecasting	107	2	P.C / T / E / 4	2	O/ T
Engineering Estimating	114	1	P.C. / S / E / 3	2	Sy
Finance (3e)					
Credit Advisor	45	1	P.C. / S / SA /4	1	Sy
Clarifying Legislation	54	1	P.C. / S / SA /2	1	Sy
Customer Finance	55	1	P.C. / S / SA / 3	1	St
Financial Management	56	1	P.C. / S / SA / 4	2	O
Payroll Procedures	72	1	MF / L / F / 3	2	Sy
Order Entry	86	1	P.C. / T / E / 1	2	O
Price Forecasting	107	1	P.C./ L / E / 5	1	T
Capital to Contract	131	1	P.C. / S / E / 4	1	T
Financial Pack Collator	132	1	MF / 1 / E / 2	2	Sy

Business Function / Application	Appl'n Number	Project Length	Resource Needs	Cost	Impact
Expense Approval	135	1	P.C. / S / SA / 2	1	Sy
Capital Evaluation	43	< 1	P.C. / S / SA or E/2	1	T
Quality Assurance (4e, 5e)					
Defect Analysis	8 / 13 / 59	2	P.C. / L / E / 3	2	Sy/O
Codes of Practice	10	2	WS/ S & L / E /2	4	O
Quality Education	12	1	P.C. / S / E / 4	2	O /St
Release Notes	14	1	MF / L / F / 5	2	Sy
Reject Analysis	15	1 -2	P.C. / L / E / 3	1 -2	Sy
Intelligent F / end	83	2	MF / S & L/F / 2	3	Sy/O
Tech. Specification	48	2	MF / L / F / 1 - 2	2	O
Industry Regulations (BS 5750)	65	1	PC / S / SA / 1	1	T
Personnel (6e.)					
Training	12/ 110 / 121	1 - 2	PC-WS/S/SA -E/3	2 - 3	O
Employment Law	46	< 1	P.C/ S or ASS/SA /2	1	Sy
Clarifying Legislation	54	< 1	P.C / S / SA / 1	1	O
Personnel Policy	63	< 1	P.C. / S / SA / 4	1	O
Industrial Regulations	65	1	P.C. / S / SA /2	2	O
Payroll Procedures	72	1	PC-MF/ S / E / 3	2	Sy
Recruitment	119	1	P.C. / S / SA / 5	1 - 2	O
Computing (7e, 8e)					
LAN Advisor	1	1	P.C. / S / SA / 4	1	Sy
S'ware Development	2	1	P.C. / ASS/SA / 4	2	O
Mainframe Diagnostics	3	1	P.C. / S / SA / 2 - 3	2	Sy

Business Function / Application	Appl'n Number	Project Length	Resource Needs	Cost	Impact
Mainframe Tuner	4	1	MF / ASS / F / 0	2 - 3	O
Log Analyser	5	1	MF / L / F / 5	3	Sy
PC Help-Desk	6 / 71	1	PC / S / SA / 4	1	Sy
Terminal Helpline	7	1	P.C./ S / SA / 4	1	Sy
Network Configurer	38	1	P.C. / S / SA / 3	1	Sy
Printer Troubleshooter	57	1	P.C. / S / SA / 5	1	Sy
Training Aid	66	2	MF / L / F / 5	2	Sy
Obsolete Hardware	69	1	P.C. / S / SA / 3- 4	1	O
Computer Upgrades	91	1	MF / L / E / 2	2	O
Programming Aid	118	2 -3	MF / L / F / 4	3	O
Capacity Planning	84	1 -2	MF / L or ASS/F/ 4	3	O
All Areas (Applications which are not functionally specific and may be applied to all areas of the business)					
Archiving	e.g. 9				
Codes of Practice	e.g. 10				
Training	e.g. 12 / 110				
Database Analysis	e.g. 13 / 95				
Legislation	e.g. 54				
Administration	e.g. 62				
Document Reduction	e.g. 82				
Intelligent Front- end	e.g. 83				
Report Processing	e.g. 90				
Procedure Manuals	e.g. 94				
Statistical Advisor	e.g. 117				
Diagnostics	e.g. 71/ 108				
Selection	e.g. 28 / 58				

2. Applications Portfolio

This section gives a brief outline of each potential application corresponding to an assigned number referred to in Part One.

Page 1

Appl'n No.	Application Details
1	**LAN Advisor** : This could help technicians to diagnose trouble situations in GEC's Local Area Network. There is a packaged system available in America for sale by Pacific Bell called IDEA; typical consultations last 2 -3 minutes or less. The system is rule-based and was developed using a shell.
2	**Software Development:** A useful system could be one which helps engineers and managers to schedule and staff software development projects. The system can predict labour requirements, cost, and time involved and forecasts the productivity of projects. The system could be used therefore to either model costs or estimate and schedule. A similar system has been developed by Level 5 called COCOMO1 on a rule-based shell called Insight 2.
3	**Mainframe Diagnosis:** An established application for rule-based systems is the diagnosis of problems and failures of company mainframe systems. Systems used by SD-Scicon and Cambridge Consultants reduce the average time taken to solve a typical problem from forty-five minutes to five minutes, thereby reducing computer downtime. There is a wide scope for application within GEC-traction because of the shortage of expertise and the fact that new people are being trained to acquire expertise (and therefore an expert system could help to archive existing knowledge for the benefit of new staff). The system can be extended into all areas and levels, from troubleshooting disk drives for the technician; analysis of tape-drive failures by analsing error messageslogged in the system event file; to a help facility for the user in solving minor problems.
4	**Mainframe Tuner:** There are a number of commercially available systems which monitor the performance and recommend refinements and tuning to mainframe systems. Use of such systems improves the productivity of computer operations controllers. Two such systems include Vax tuner by SD-Scicon and DASD Advisor for the IBM.
5	**Log Analyser:** it should also be possible to analyse computer maintenance logs to identify possible future problems. The system would increase system availability and improves the efficiency of service operations. The system would also be effective in storing and recording maintenance logs in a user friendly way.
6	**P.C. Advisor:** Many of the above capabilities apply equally to PC based systems. Presently at GEC-traction, there is a single engineer responsible for servicing and ensuring the efficient use of over a hundred systems, including numerous printer and cable configurations. A small rule-based configuration could help the support engineer to troubleshoot problems more effectively or allow the forthcoming assistant to undertake some of these tasks.
7	**Terminal Helpline** A substantial amount of the operations engineer ' s job is spent advising Users how to edit and perform other routine and specialised tasks, and perform basic screen related troubleshooting. These tasks could be carried out by the Users themselves by the use of an on-line and context sensitive advisory system. Such a system would free the ' expert ' and also speed the training of new operators. The system clearly would need to function on the mainframe and therefore would be written in the source language- PL1 in the case of the VAX.

Appl'n No.	Application Details
8	**PCB Defect Analysis:** Presently Quality Assurance undertake trend analysis on the basis of a ' snag report '- this report fails to describe in suitable detail the symptoms, causes and effects of defects. An expert system could help to formalise the reporting procedures (as a friendly data-base) and also help the analyst to identify trends and implications. It could also be used by members of the PCB room itself to diagnose faults in the first instance. The system could be a stand-alone rule-based shell.
9	**Archiving:** The quality assurance department is relatively small and yet quality as an issue is a company wide responsibility. Expert systems may help to store present expertise and diffuse it throughout the company.
10	**Codes of Practice:** These and Test and Inspection specifications represent a formal source on standards, specification s, detail , company policy , procedures and practice . Expert systems can provide a more effective alternative which allows for the rapid search of requirements.
11	**Quality Control:** Expert systems may be used in three primary areas: to decide upon sample test sizes and test procedures; fault location; and fault identification in assembly and materials control. A further use might be in the choice and monitoring of quality of goods from suppliers. An expert system could help to apply more qualitative factors in the decision-making process- for example quality, reliability, delivery performance and so on. This would help the user to decide when to change suppliers and would also reduce the need to expedite (this would be of particular use with castings).
12	**Training:** In order to improve awareness of quality, expert systems could be used for training. They can do more than simply convey information, they can also ask questions and assess users responses and therefore their understanding of the problem.
13	**Database Analysis:** Accessing data from the database and making meaningful conclusions from this data is a further area where expert systems might be used. Using heuristic search techniques, expert systems can very quickly identify the relevant part of the database to be accessed and then undertake an analysis on this data , for example . This is of particular use in quality assurance where there are substantial historical records of faults and defects which require interpretation .
14	**Release Notes:** A system which could make use of the experienced judgements of the QA manager on whether to release customer quality assurance documents.
15	**Rejection Analysis:** A system to analyse trends in the types of rejection and stages in the production process in order to identify design and process faults. This could also be of use to engineers in identifying generic design faults.

Appl'n No.	Application Details
16	**Customer Order Engineering:** The process involves converting electrical schematic drawings of a standard product into wiring drawings of customer specific product versions. To do the wiring of an electrical system, a program needs to know the complete list of components of the system, and other physical characteristics. Such a system is available in the USA called CORA: developed by Westinghouse, it uses the A.I. language GLISP(Marketed in the UK by Artificial Intelligence Limited) to provide the following:- a) Negotiation tool - generation of generic billing - early specification problem identification b) Engineering Front End - generic bill-of-materials providing catalogue numbers and other information - timely identification of problems and missing information, resulting in much reduced engineering turnaround time. - consistency of styles between engineers.
17	**Automatic NC Program Generating KBS:** A system which automatically generates the computer programs used by numerically controlled(NC) turning machines in the manufacturing industry . A particular system developed by Technology Applications (distributed in the UK by Artificial Intelligence Limited as AGES) uses Golden Common Lisp language on a PC to provide a rule based architecture operating from within frames. The system equals the performance of the experienced NC programmer to develop a program. This process can be broken down into three basic steps : geometry definition, manufacturing process identification, and automatic code generation
18	**Cutting Fluid Selection:** This simple rule-based system uses information on the machining operation, materials and problems and provides an analysis and recommendations on the type of cutting fluid necessary to perform a particular job.
19	**PCB Design:** A system developed by Carnegie Mellon(called ECAE) uses input data in the PCB design for manufacturability. It is possible using this system , to identify design problems from signal integrity and waveform analysis and recommend structural changes such as wire length and electrical properties. A likely role for this application specific software is as a front -end to existing automatic testing facilities at the Trafford Park site.
20	**Process Planner for Assembly:** This makes use of CAD layouts, parts listings and special instructions to develop a process plan. Such a system is in use by Carnegie called OPGEN. (see also No. 53 & 97)
21	**Facility Layout:** a rule-based model which uses simulation and kbs techniques to design an optimum layout. It can also assess the implications of relocating machines etc. Systems in this area include FADES, Simulation and CRAFT- see No. 99 & 128
22	**Maintenance Design:** a rule-based system which advises upon an optimum schedule for preventative maintenance and can also cater for crisis maintenance (see also No.'s 64, 78 , 93, 100 , 126, & 138 for other maintenance applications.

Appl'n No.	Application Details
23	**Design Specification Selector:** The purpose of these specifications is to convey the manufacturability and testability requirements into the design environment, at the right time and in the right form; and to gather expertise upon which these requirements are based from the engineers who have this expertise. Presently, Standards fail to do this because of a number of reasons:- a) lack of common meaning (Jargon!) b) often standards are out of date, c) not referenced enough d) difficult to use effectively e) essential detail is sometimes omitted f) time consuming process. With the capabilities of easy updating , effective explanation facilities and rapid search techniques for the speedy accession of relevant data, expert systems clearly have a potentially useful role in this area.
24	**Shipping Planner:** A system to advise on the distribution of finished goods according to regulations and the specific requirements of the client. Factors include routes, stops, diversions, due-dates, speeds, regulations, capacity, weight, volume, cost(milage, stop-overs, loading, waiting, double-up etc.) and packaging materials, design of casing etc.
25	**Capacity Planning Advisor:** The distribution of work between the two sites, Trafford Park and Strand Road, follows a number of complex decisions, some of which transcend physical and locational availability. A rule-based system could advise and highlight important issues in the distribution of work. Capacity planning is also necessary at a lower level and would be particularly useful in computing where software and hardware development projects and utilization require planning
26	**Make or Buy Decision Advisor:** This expert system will help purchasing personnel, engineers and managers maker decisions on whether a part assembly or a product should be built internally or be sourced outside of the company. Built into the system will be a set of basic questions necessary for reaching a logical business decision aided by a knowledge base augmented with past experience from various sources internal and external to the company. The system consists of a basic 'general ' site strategy product programme, site resource constraint related questions and detailed supplier sourcing knowledge. It would also access existing databases for information, such as product cost estimates in industrial engineering, capacity planning forecasts from planning and development costs from accounting. The rationale for developing an expert system in this area is that it would help to reduce the risk of making wrong decisions due to lack of information or of making an incomplete, rash decision because of negligence.
27	**Combine Testing:** This system works on trend analysis in which a history of vibrational information is used to predict failure. The system could also make use of frequencies, time signals, shock impulses, amplitudes and harmonics of shafts, rotors and other data representations. Incorporated into this system could be systems modelling, design modification, monitoring and predictive maintenance.(see 51, 60, & 74)

Appl'n No.	Application Details
28	**Selection of Lubricants:** A system to select a quantity of oil or grease, the necessary additives and mixture ratios according to particular applications such as metal cuttings.
29	**Selection of Welds:** An expert system front-end to a complex database which contains critical information on base metal types. The system gives expert advice on what materials should be used to weld base metals.
30	**Selection of Joints:** An advice giving system which suggests the type of joint suited to a particular application. The joint may be a weld, glue, adhesive and so on, and may require certain preparations and treatments etc.
31	**Selection of Tools:** The trend towards smaller batch sizes places demands upon the use and flexibility of tools, fixtures and incoming stock. There are a number of physical numeric and non-numeric considerations involved in the choice of tool and this knowledge may be incorporated into an expert system. The machinist may be able to tell when a cutting tool is dull simply by looking at it, but still may be unable to explain how he knows the tool is dull. However, upon further intuitive analysis by questioning, the machinist may be able to articulate the general shape of the wear scars on the top face for example, corner and flank face of the tool, and in addition relate these features to events in the last machining pass. Changes in chip colour, vibrations of the machine tool and the surface finish obtained on the component are the first signs that the tool is becoming worn and should be checked at the end of the machining pass. This suggests that there is a " trend analysis" active in the mind of the machinist. This analysis could take the form of simple rules which could be incorporated into the knowledge-base of an expert system as follows:- GIVEN THAT THE TOOL HAS BEEN USED FOR 'X' MINUTES AND THE SURFACE FINISH ON THE PART IS DETERIORATING AND THERE IS NO RUBMARK ON THE BAR THEN THE CUTTING SPEED CAN BE REDUCED BY 'Y'.
32	**Materials Change Control:** This is a meta-level control knowledge-based system which overlooks existing database operations and routes changes, decides upon changes, and informs those that need to know in the right format.
33	**Contract Handover:** The handover process from TPL to Traction appears to be ad-hoc and informal when infact a formal specification of requirements, terms and conditions would be expedient. A small p.c. based system would provide an instructive procedural consultation which guides the user through an arrangement which covers all pertinent issues.
34	**Machine Tool Design:** This provides recommendations to the industrial engineer on the type of machine tool hand. Issues include stiffness, machine access, ease of placement, set-up times, feedback shape and position and positioning accuracy. This system would make use of rules-of-thumb of the machinist together with machining physics- hard data and facts on cutting face values etc.

Appl'n No.	Application Details
35	**Error Detection in Parts Programming :** The ability to detect and diagnose errors in a manufacturing system is necessary for building autonomous controls for machining. The problems that an expert system will be able to diagnose in a machine shop include the identification of parts programming errors and the location of machine tool features. The former is based upon analysis of historical records of error types logged by the NC programmer. The likely reason for program changes are as follows:- - Dimension error (20 % of faults) - Preparation code error (2 %) - Function (8 %) - Set-up: shop (5 %) - Set-up: programming (4 %) - Techniques: programming (18 %) - Excess Stock (6 %) - Unknown Stock (1 %) - Engineering Revisions (4 %) -Routing changes (3 %) - Speeds & Feeds (9 %) - Link Error (1 %) - Tooling: Shop (6 %) - Tooling: prog. (7 %) - Defective tape (.1 %) - Damaged tape (. 5 %) - Machine control (1.8%) - Comment error (3.7%)
36	**Machine Level Planning & Scheduling:** At the machine level of the factory, planning and scheduling are required to determine the order of the different cutting operations of a particular part (for example, milling followed by drilling followed by tapping). In most cases, there are several possible ways to order the steps involved in producing a piece. The problem is generally to find the quickest sequence that minimises the number of steps in the fixture. However, other factors can be involved in making the decision ; for example, the machinist knows that as a component is shaped, it becomes increasingly more difficult to hold in a machine tool fixture. Therefore, the machinist may plan to do some of the most difficult cuts first, before the piece becomes awkward to clamp. In other words, it may be wise to concentrate on safe holding and safe machining rather than trying to minimise the set-up operations. This is just one example at the machine level of how a number of factors need to be considered in planning and scheduling in order to arrive at the most appropriate option and is an area where constraint driven knowledge-based systems can be of help.
36	**Shop-Level Planning & Scheduling:** This is required to determine how to allocate the available machines, materials and personnel to finish a variety of parts orders in the least amount of time or by the scheduled due-dates. Trading off constraints between minimum production time and satisfying contract requirements is an example of job-shop scheduling, and this is an area where Knowledge based systems are useful. A particular system called ISIS (intelligent scheduling and information system) determines various sequences of the operators. This is difficult to implement without the experience of the skilled industrial engineer who appreciates the order in which machines must be employed to achieve effective tolerances, surface finishes and so on. The knowledge of the engineer is encoded in this system as a 'constraints' that are used to focus the search for an acceptable machine operation sequence. The ISIS system addresses the problems of how to construct accurate, timely, realizable schedules and how to manage their use in the job shop. *(continued on page seven)*

Appl'n No.	Application Details
37	ISIS carries out a hierachical, constraint directed search through the space of possible schedules. The search moves through more and more detailed levels of analysis with specific types of constraints coming into play at each level. ISIS is capable of incrementally scheduling orders as they are received by the plant as well as reactively rescheduling orders that are affected by machine breakdowns and other dynamic changes in the shop. A similar system which is commercially available in the UK, called **Ikonman**, is a job-shop sccheduling system designed for a process control environment although it is being extended for use in manufacturing.
38	**PCB Wiring / Cable Configurer:** The configuration of PCBs, resistors, wiring, meshes etc., determines the efficient arrangements of wires connecting circuit boards. Based upon the experience of past contracts and requirements, the system can aid the productivity of the engineer unfamiliar with new configurations. The system could run on a p.c., but more complex systems would require a workstation.
39	**Commercial / Sales Configurer:** The experience in selling previous contracts can be used to great effect at the Tendering stage in making a first pass or top-level estimate of costs and duration through a rough configuration based upon the requirements of the customer., An analogy can be made to the very successful systems used by the sales departments of large computer companies such as IBM, Digital , ICL and Honeywell. These companies faced the problem of sales personnel requiring great expertise in order to configure and recommend the best arrangement of cables, disk drives, layouts and hardware, networks, financial packages and other arrangements with the customer. By using knowledge-based systems, these companies have been able to reduce the time taken to configure(at a top-level) the requirements of the customer; to explain effectively why certain decisions or approaches were made, based upon the explanation facilities of the system; to act as an effective training aid to new sales management; to archive existing skills; and these systems also help to direct updates of inventory entries- although this option is less applicable in the case of GEC-Traction Limited. Such systems have led to great improvements in the competitiveness of the company and has improved the " image " of these companies- a factor which should not be overlooked.
40	**Tendering Protocol:** The size and complexity of contracts at GEC requires that the Tender is validated from an engineering standpoint at an early stage. A knowledge based system could be developed which validated consistency and accuracy of the approach taken and the subsequent information generated , but this would be a large integrated system taking many man-years of effort. However, a much smaller system operating on a p.c. could be developed which would provide an appreciation to TPL sales of the important factors and constraints to consider in Traction Engineering when in consultation with the client. It could also be used as a checklist to ensure that all factors and issues- contractual, legal , technical, economic etc. - are not overlooked.

Appl'n No.	Application Details
41	**Equipment Layout:** The design of equipment cases is constraint driven and therefore amenable to the use of Knowledge-based systems. Prior to more detailed design, a case may have to satisfy fundamental issues of cost, safety, maintenance and construction requirements. A KBS can help the designer to specify a top-level configuration and layout prior to detailed sub-layout design. When a design is modified by a user, the system is also able to modify other sections of the design as needed to maintain consistency. In effect, it is an intelligent reconfigurer.
42	**Project Layout - TPL:** Recent turnkey developments such as Docklands involve a number of companies each providing some product or service towards the total project. The management of the interface between products and the design of layouts may be assisted by a top-level layout rule- based (i.e. constraint and parameter driven)knowledge-based system. This would help to reduce the number of possible combinations of manufacturers, suppliers, products(propulsion equipment, rolling stock, bodywork, construction etc.) and services to a limited number which meet the total requirements of a particular contract. Following this process, more detailed analyses based on existing simulation and project management techniques would be appropriate.
43	**Capital Acquisition Advisor :** The intention of this system is twofold:- a) To help engineers with no special expertise in finance and economic justification to be able to price and justify proposals and provide a standard format of presentation which is effective in highlighting the benefits and limitations of the proposal. b) To help the evaluator to assess proposals by covering both qualitative and quantitative issues as part of a general assessment. The system could also be used as an interactive system to help in the paperwork involved in the transfer or disposal of pieces of equipment, computers or machinery. It asks enough questions to determine which forms are needed, what must be filled in, whose approvals are required, and where the completed form should be sent.
44	**Parts Numbering:** A similar administrative advisor could be used in helping users to understand and apply the correct parts numbers, procedures and documentation to the correct operation.
45	**Letter of Credit Advisor**: Letters of Credit are used by a bank on behalf of an importer to the bank of the exporter in another country. When documents verifying the terms of the delivery have been issued, the exporter can collect payments. Different banks and companies have different requirements and formats for letters of credit, but they all share basic underlying rules governed by uniform customs and practices. An expert system advisor facilitates the detection of discrepancies between the letter of credit and accompanying documents, and recommends possible courses of action. This system would operate on a basic p.c. based shell.
46	**Employment Law**: this contains knowledge about U.K. employment law. The system helps determine if employees are covered under rules of the Wages Act of 1986 and the Sex Discrimination Act of 1986. The system was written on a shell and is available as a commercial, 'off- the- shelf' product by Expertech.

Appl'n No.	Application Details
47	**Electrical Connector Assembly**: A system could be developed which selects the correct tools and materials for assembling electrical connectors. In actual use, it can reduce specification research time from an average of 42 minutes to 10 minutes, with more uniform selections. This particular system was written in Quintus Prolog to run on a MicroVax.
48	**Technical Specifications- Works Services:** This provides information regarding the safety and regulatory related consequences of proposed equipment maintenance and operator actions in manufacturing plants. Each type of plant requires its own knowledge base. The system can also be used to help engineers to be able to select the correct specification for each application or process.
49	**FMS Diagnosis:** Preston Maufacturing: the installment and learning experiences of the FMS engineers have led to the development of case histories of faults. These could be used to good effect as a troubleshooter and maintenance assistant.
50	**Weld Selector:** (Preston Manufacturing): A number of packaged systems are available helping welding engineers to choose the proper weld electrodes. With its use, the time required in this process can be reduced significantly.
51	**Rotating Equipment Advisor:** This system would provide assistance in interpreting vibration patterns and measurements in rotating machinery. Applications at GEC include the monitoring of cutting for the FMS and Combine Testing at Preston
52	**Manufacturing Cost Estimator**: A rule-based system makes it possible to bring all existing data together in order to arrive at a cost estimate for propulsion equipment. Given a customer request for Traction equipment, a cost estimate is produced by considering the parts required, whether a similar contract / part has been built before, and the material and manufacturing costs.
53	**Manufacturing Process Planner:** An example of a system in the USA to aid in the planning process for the manufacture of approximately 20,000 parts that go into aeroplanes. The system develops a plan that identifies the operations that need to be performed on a piece of raw material to transform it into a finished item, as described in the engineering drawing or model. The plan includes specification of the equipment to be used on the shop-floor, any additional tooling required, and the sequential routing of the parts and associated materials through the factory. The system currently plans the manufacture of sheet metal extrusions and operates on a dedicated A.I. toolkit called KEE.
54	**Clarifying Legislation:** These systems provide friendly , usable and rapid searches and interpretation through lengthy documentation such as standards, codes of practise and legislation. Advice could be given in Personnel for example on various acts of Legislation such as Employment law, Dismissal, Sick Pay, Tax Regulations, Maternity leave, Pension schemes, and so on.

Appl'n No.	Application Details
55	**Customer Financial Planning:** TPL is involved in generating financial packages for clients. Typically, large amounts of information are analysed with such factors as new tax laws, interest rates, inflation, exchange rates and so on, in order assemble a plan that suits the client's needs. It is difficult to analyse and retain all this information and the human becomes overloaded. A KBS could be used to assist the planner with al the information in order to recommend a suitable plan. The output of such a consultation is a plan which will advise clients on what package to take.
56	**Business Financial Management:** GEC's financial condition is evaluated by considering such standard business documents as income statements and balance sheets. Using these documents, an expert system can perform a cash flow analysis, for example. Based on the results of this analysis, the expert system can make recommendations for improving cash flow. For example, it may suggest that the accounting methods used in dealing with company inventory be reconsidered. Such a system would take the form of a shell interfaced with Lotus 123 or Dbase III. Equally though, the system could be used to advise on budgetry spending , price fixing and profit evaluation.
57	**Printer Fault Troubleshooter:** A substantial portion of the support engineer's time is spent resolving printer problems. A help desk type approach would help the users to identify faults by themselves and also carry out basic maintenance without the need to consult the engineer. The same principle could be extended to p.c. diagnosis and terminal troubleshooting.
58	**T. & I. Spec. Selector** - To help engineers select the most appropriate specification for a particular design. A simple application with the potential to be integrated with CAD systems in future.
59	**Defect Analysis Advisor-** quality assurance have a large database of faults and defects. Use could be made of this data through an intelligent interface (see No. 13).
60	**Combine Test Analysis-** to make use of real time information of vibrational and other sensor information to test for faults in moving machinery- see No. 27 also.
62	**Knowledge-based Administration:** KBSs can be used in all areas of the company to make complex sets of rules and regulations available to large numbers of staff.
63	**Personnel Policy :** Personnel follow set rules in order to be as fair as possible to all employees. By encoding such rules, KBSs can ensure unbiased recommendations and so help achieve a consistent personnel policy.
64	**Maintenance Scheduling:** In order to make the most effective use of machinery, maintenance must be tightly scheduled. KBSs can assess the effects of increasing, reducing, delaying or bringing forward the out-time involved and can work out the optimum maintenance schedule.

Appl'n No.	Application Details
65	**Industry Regulations:** A time-consuming part of design is looking up regulations, for example, and other specifications. KBSs can indicate when regulations are applicable; which regulations must be obeyed; and how these are translated into action.
66	**Training-Aid for Computers:** An expert system that provides novice users with help in using the VAX/ VMS operating system. Systems are available which recognise both errors and inefficient operations.
67	**Client Maintenance:**To help clients of GEC-Traction to troubleshoot high prone faults in equipment and recommend consequent debugging and repair advice. This will speed up the process of diagnosis and will reduce the frequency of repair and maintenance calls.
68	**Aid for Production Engineers:** a system available to select the best process to make a particular component and to cost it. The user is asked to enter details about the component such as complexity, tolerances, length etc. Using a system of weights, the system then selects the best process out of a list of possible alternatives. The user can change any details previously entered to see what effect this has upon the selection process. This system differs from the existing process planning facilities, MICLASS and Supercapes , because it captures the expertise of the engineer in defining process requirements from an analysis of the drawing as well as matching these requirements to available on -site manufacturing processes. Thesefore an expert system in this area would be seen as a higher order of interface feeding down to Supercapes on the Microvax.
69	**Outdated Computer Hardware Diagnosis:** GEC make extensive use of a wide variety of different computer systems which are continually being upgraded and extended. Some of the company 's older hardware, although now considered obsolete by the manufacturer, remains an essential part of its integrated systems. There are now very few people within the organisation that know this hardware well and it would be useful to archive this expertise into rule-based systems to provide an on-going fault-diagnosis and advisory capability.
70	**Welding Analysis:** This company makes a large number of special order one-off products. It employs a wide variety of welding processes and an expert system would be of use to provide guidance as to which welding process would be most suitable for a given product design.
71	**P.C. Helpdesk:** GEC has a large number of users of personal computers. An expert system would be of use to help diagnose faults that occur on p.c.s and communications equipment. The aims of the system would be to increase awareness and understanding of p.c. problems to lower grade staff; archive rare problem occurances; and reduce downtime.
72	**Payroll Procedures:** A proposed system designed to clarify rules covering overtime payments, shift working, bonuses, holiday pay, etc. for manual staff.

Appl'n No.	Application Details
73	**Automatic Testing and Repair of PCB's:** a knowledge-based system which automates the test procedures used by inspectors to manually test a circuit board. Schematics of a board design are captured on-line in a hierarchy of modular diagrams, allowing users to choose the level of detail appropriate for the current task. The KBS understands the relations between all diagrams in a design and can switch automatically between them as appropriate. Schematics can be annotated interactively at runtime with any kind of information, in particular, test procedures, signals, and user comments, which is then shared by all users of the system. This annotation of the diagrams allows junior engineers to learn the knowledge of more senior engineers. In addition, the KBS could derive new test procedures from a schematic. (A similar system was developed by Intelligent Applications Limited called ' Synergist '.)
74	**Machine Monitoring:** A knowledge - based system which monitors and interprets the health of machinery that has large vibrating parts. It can unterpret either taped data or live data taken during operations. Based on the data taken from the monitoring system, the KBS is able to predict which machine parts will fail in the future; or act upon this information as a controlling function. A system called Violet supplied by Intelligent Applications Limited offer such capabilities.
75	**Bid Estimation Support System:** This is a proposed system providing intelligent support to cost estimators in producting ' bid estimates ' for complex manufacturing components. The system will:- a) Speed up the time taken to produce an estimate b) Share the experience of the best estimator among less skilled staff c) Allow for the integration of estimating with other business functions d) Improve the consistency and accuracy of estimates. The system uses an intelligent prompting mechanism to arrive at a given estimate. The system works on the principle that each project contains a hierarchy of a potential structure. For example, a 'system' will contain control gear , motors , pantograph and so on. For each of these components, a number of sub-components are potentially present, and so on down to the actual materials and methods used. The system will prompt the estimator with potential structures from which he or she will choose the component present. From there, an estimate can be made up of any combination of actual prices; unit costs; and detailed item costs- these are accessed from existing databases. Therefore, at any time, the estimator has the ability to feedback chosen estimates to improve and refine the estimation process. This particular example was developed by Telecomputing Limited using an Amstrad personal computer and IBM mainframe operating under VME. The mainframe and personal computer programs were written in ' TOP- ONE ' prolog, made up of 800 IF- THEN type rules using both forward and backward reasoning .

Appl'n No.	Application Details
76	**On-Line Shop-Floor Monitoring:** This system carries out on-line state monitoring, fault diagnosis, fault prioritisation in concurrent multiple fault situations, fault alert, provides assistance in recovering from faults and gives explanations of diagnoses, effectively providing a fault- level central warning panel. The system was developed by Cambridge Consultants and is called MUSE: available as application specific software, it is intended for real- time and process environments. Within GEC- Traction, such a system may be useful for the flexible manufacturing cell and combine test facility, both at the Preston works.
77	**Electrical Connector System:** GEC's hardware specifications for electrical connector assembly are found in many pages of documentation. A knowledge based system would help to reduce the search time required to locate the required specification. A similar system has been developed by A.I. Limited using Quintus Prolog
78	**Equipment Calibration:** This system could provide the engineer with recommendations on the types of calibrations and tests for calibrations for different tools and processes. It would also prescribe accuracy of tests and requirements and may refer to relevent codes of practice.
79	**Production and Maintenance Scheduling:** GEC Traction is faced with the situation of matching requirements to resources and thereby scheduling a plan flexible enough to allow them to assess the impact of changes. Such knowledge based techniques go beyond the capabilities of MECCA scheduling because it allows one to simulate different scheduling scenarios, provide explanations of reasoning and allocate changing priorities over a period (see notes elsewhere on scheduling).
80	**Flexible manufacturing Cell Scheduling:** The FMC at Preston offers the prospect of much increased productivity, but it does so by increasing the complexity of operations and the difficulty of production scheduling which often means that human decisions fail quality and timing goals. The FMC requires production scheduling that can handle the changes demanded by flexibility : determining the sequence of motor frames to be machined on the FMC requires that the due dates of each lot are met while taking into account several related problems, such as minimising machine idle times, queues at machines before and after the FMC and the status of work in progress. Knowledge- based systems offer potential to solve such complex, ill- structured problems because these tasks require a certain amount of reasoning capability and expert knowledge and because no direct algorithmic solution is feasible.
81	**Simulation and Control of the FMC:** To be effective, simulation tools must be responsive to the needs of the designer, allowing alternatives to be quickly and easily evaluated. A knowledge-based simulation and control system would allow the rapid reconfiguration (and performance evaluation)of models of the system under design, together with the re-use of the model for control of the real system. Since the basic

Appl'n No.	Application Details
	(Continued from page 13) structure of the cell has already been determined, the design task for a particular cell is very much a configuration problem, determining the quantities of the standard units required to meet a particular set of production requirements. There are three stages in this configuration process:- a) Enquiry- to determine production requirements. This part of the system establishes the component types to produce and their interrelationships (assemblies) ; their target production capacity (assembly quantities) ; their machine utilizations; system manning / operating schedule and expected operator utilizations. Part of these requirements could be gained through an interface with the MECCA job- shop scheduling system. b) Production Engineering Analysis- this involves the process planning and design of fixtures and includes estimated component machining times, estimated component loading and unloading times and machine , fixture and tooling quantity estimates. c) Simulation - to assess performance and arrive at a final configuration. This part of the system provides achievable machine and operator utilisations, production rates and outputs, final machine and fixture quantities and manning schedule. Knowledge- based scheduling systems are only just becomming commercially available; the most notable examples are MUSE (by Cambridge Consultants) , Ikonman (by YARD Software) and Schedulemaster (by ICL)
82	**Documentation Reduction:** The volume of data received by management is, at times, excessive and of limited or no use whatsoever. A front-end to computer systems could reduce the presentation of data to that information which is of importance to the manager. It would alsopresent information in the most useful format and filter that which is unneccesary or necessary according to different situations and circumstances.
83	**Intelligent Front- End:** There are a number of systems available which simplify the use and interpretation of databases. Prowindows (ICL) for example, is a graphical front- end to Quintus prolog which allows the use of windows, menus, graphics and explanation and help facilities. Similarly, a system called Nemesis (ICL) is a front-end, menu-driven system for giving non- technical people access to many data systems.
84	**Capacity Planning:** In building a capacity plan, the expert goes through the process of collecting and assessing the information concerning the existing and proposed demands made on the shop- floor, computer system etc., together with their expected impact. ICL have developed a capacity planning system for computer systems -it will be commercially available in June 1989 - that operates on a rule-based p.c. shell ; however, the same principles may be applied top shop- floor applications, in management and administration.
85	**Stock Control:** A system to assist controllers in inventory allocation and relocation; and to provide advice on the use of documentation and the transfer of materials.
86	**Order entry:** A system to advise on order entry. The system automatically handles complex pricing discounts, volume purchases, back- order release, credit approval, carrier selection and other tasks now requiring human expertise and time.

Appl'n No.	Application Details
87	**Production Line Maintenance Advisor:** On the shop-floor of the Preston site particularly, there is no formal recording of faults and consequent action following breakdowns. An expert system would provide an efficient means of logging faults whilst at the same time capturing and archiving the presently largely unwritten knowledge held by the experienced engineers. It would also allow maintenance staff to diagnose a wider range of problems without the need to consult engineers. The prospects of developing such systems will lead to reduced down-time and maintenance costs.
88	**Knowledge- based Aid for Equipment Design:** In GEC-Traction , testability has a major impact upon the maintainability and the in-service cost of electronic equipment. Testability is not something that can be retrospectively added to a design: it must be built in at all stages in the design process. Knowledge-based systems may be of use here in identifying the relevant constraints and parameters which must be considered in the design of " testable" components or systems. The prospects of using KBS techniques to improve testability in the computer aided design of electronic equipment will lead to improvements in the specification and design of electronic equipment
89	**Cutting Tool Expert:** Selecting the right cutting tool is a known problem in GEC-Traction The expertise to get it right is not widely available. Selection of the wrong cutting tool can easily damage the tool or the workpiece and the costs in terms of time and money may be substantial. The benefits of developing an expert system in this area would be a record of how experts select cutting tools which could be integrated with proposed tool management systems. A similar system was developed by SD (Camberley) using a small rule- based shell.
90	**Report Processing:** This knowledge based system is intended to assist in the design of report processing proograms to verify all kinds of information from assuring the correct form has been used to verifying the proper format of documents.
91	**Computer Enhancements:** Systems could be developed which add elements of expertise, advice, and help facilities to data processing applications. This could offer a useable interface to systems without the user necessarily being aware of the technology. Furthermore, the knowledge-based system could be used to select the most appropriate processing method and delivering the most efficient results.
92	**Evaluate Subcontracting Bids:** A system which will help in the decision-making process of whether to subcontract work or not. The system would review price, delivery, past performance, quality and all the factors that can affect the finished product or component. A similar system was developed on a P.C. rule-based shell called GURU.
93	**Maintenance Management:** A knowledge-based system which allows management departments to organise and plan all maintenance related activities, i.e. preventative maintenance, schedules, equipment records, work orders, spare parts etc. Such a system was developed with reported savings of 15-20 %.

Appl'n No.	Application Details
94	**Procedure Manuals:** Procedure manuals, whether for policy, pricing, maintenance, or instruction, typically contain a series of rules that depend on other rules. A particular rule may infer to a number of look-up tables or codes-of-practice, or may require you top go through a series of calculations based on whether the rule applies to a situation. Shells offer the capabilities to automate the search while speeding up and simplifying the process.
95	**Database Queries:** There are potential benefits in making the information held in data processing facilities more amenable, allowing staff and employees to access complete data systems. Data retrieval is performed by composing English-like commands, but with a relatively formal syntax. The problem arises when people require only occasional access to information systems. The use of computer systems can be forgotten ,and, no matter how easily learnt, it is frustrating trying to recall the required commands. Infact, it is often easier and quicker to plough through the manual rather thanembark on a trial and error session. A solution has been developed by ICL (called QueryMaster). British Gas use QueryMaster for extracting information from their VME databases. For trained or experienced users familiar with the system , Querymaster provides a useful and productive data-extraction service. For occasional users with only ad-hoc enquiries, a different level of entry into Querymaster provides a front-end in order to illuminate the database structure, its layout and responses , and is able to initiate database searches.
96	**Strategic Scheduling:** The common problems of a large engineering company like GEC Traction are long lead times and poor delivery performance, together with high inventory levels. These problems are often caused or exacerbated by poor resource management, especially in reaction to unplanned events. Production control and scheduling systems at the factory level could provide a solution but very few products are available which address the dynamic nature of the domain; the requirement to measure schedule merit by several, usually conflicting criteria; and the need for such a system to coexist with manufacturing management systems. A recent system, arising as an offshoot of the Alvey project, is Ikonman developed by Yard software: this is an intelligent Knowledge based system which schedules event driven activities at the strategic, tactical and operational levels.
97	Process Planning : The functions of process planning are to determine the sequence of machine tools, clamping devices, cutting tools, and so on from the product information represented by the product drawings and/or product model . There are particular areas of process planning where the knowledge and expertise of planners are referred to and utilised and may possibly be captured in a knowledge based system. These are:- - recognition of rough shapes of parts - determination odf raw material - determination of preference, relations among machining surfaces - recognition of important surfaces which have an essential function in the part - selection of a machining reference surface which has to be fixed on the tables of machine tools - selection of a suitable machining sequence

Appl'n No.	Application Details
98	**Project Planning Assistant :** This knowledge based system has two components which can be integrated. These are bid-proposals and project expedition. a) **Bid / Tender Proposals:** The problem for GEC-Traction , as with all large engineering companies is the speedy assessment of a prospective client's requirements from which to identify time, cost, effort and risk data which can be used to bid for the contract to produce the equipment. The salient aspects of this process are: i) identifying the features of the locomotive that satisfy the client's needs, ii) producing project plans to convey the time scale of the project and identify activities which constitute the project. With plans produced, " reasonable " time and cost estimates for the project can be given to the customer for assessment. To produce reasonable bids then is to assess fully the complexity of the propulsion equipment and additions which the client requires and then to match accurately requirements to product. A system, which is not yet commercially available, has been developed for bid proposals called PIPPA. It is intended to assist staff in performing these tasks by inferring module necessity and sophistication through assessment of customer specification, and by producing project plans, mainly in the form of project networks, so that the time, cost, effort and risk parameters can be analysed. Through dialogue between the system and the operator, the PIPPA system attempts to identify which parts of the loco equipment are necessary to satisfy the specification, and to what sophistication these must be present. b) **Expedition Control:** Expedition is the process of reducing the total duration of a project towards a particular target due-date. The necessity for expedition arises when a project activity is in danger of running grossly overtime, and ,clearly, this trend must be curbed to avoid project penalties or loss of company credibility. Expedition invariably incurs some increase in direct expenditure for an activity, as money is spent to save time, for example, by the deployment of additional resources. The aim of the expedition process therefore, is to meet the target due-date, but with a minimal increase in total project costs. This requires a great amount of knowledge and experience which can be captured in a knowledge based system. A particular system, Xpert, takes input information on the project to be expedited-regarding schedules, personnel involvement and so on; together with a target completion date in order to produce a revised schedule project network and information about which activities to expedite, by how much, and using what means. The system also uses project management knowledge to decide upon the choice of expedition strategy, applying overtime, adding resources, authorising shift work or an incentive bonus scheme, or applying increased control , for example.
99	**Facilities Layout:** The location of new plant and relocation of old according to new contract requirements and machine configurations, is a highly skilled job. A knowledge-based system may be of use in defining constraints and parameters for the evaluation of plant. The example rules below show how the expertise of the works services manager may be encoded :-

Appl'n No.	Application Details
	IF WEIGHT OF MACHINE TO BE PLACED IS W ***AND LOAN LIMITATION OF CANDIDATE SITE IS L*** ***AND LL IS (L - W) / L*** ***AND LL < 0.1*** ***THEN FLAG THIS ASSIGNMENT AS UNDESIRABLE*** *Examples of how the constraints set down by the manger can be used to configure an optimum layout. In all, about 60 rules are required to produce a full sized layout system.* ***IF SINGLE BEST LAYOUT DESIRED*** ***AND QUANTITY OF MACHINE TYPE IS Q*** ***AND AREA SIZE IS A*** ***AND MACHINE SIZE IS MEDIUM*** ***AND PATH DISTANCE IS MEDIUM*** ***THEN SELECT LAYOUT XYZ***
100	**Maintenance engineering Management:** There are three key areas in maintenance management: a database of past failures; the use of this database in finding improved design and maintenance methods; and effective action to introduce and sustain the improved methods. KBSs have a part in all these areas. Generally, there is limited visibility of machine downtime, of capacity reduction, and of product rejection rates resulting from poor maintenance. The traditional method of coping with limited visibility has been to resort to periodic preventative maintenance which generally leads to over-maintenance and wasted resources in other than critical failure areas. Such blanket maintenance policies can be replaced by the flexible management of resources and of equipment downtime. Flexible management requires a picture of the current state of resources, together with a means of alerting managers when adverse trends indicate that some action is required. It is this area particularly where knowledge-based systems may be of use. Such a system requires that the resource state picture is kept up-to-date and so the system is on-line. The system would be used in three ways: in order to meet maintenance deadlines; to react rapidly to unforseen demands; and to optimise the use of fixed resources. The system would make use of numeric data on MECCA, but also the skills and experience of engineers, machinists and planners and would at all stages, be able to offer advice and justify its reasoning.
101	**Design for Manufacture:** This is a design aid for engineers which identifies and highlights the practical manufacturing constraints which must be considered in the design application. This would improve the liason between engineering and planning and also reduce the number of defects and " departures from drawings ". A KBS could be used to observe a combination of the following constraints according to the requirements of the manufacturing / assembly process:- Functionality- temperature, wear, interchangeable jigs/ fixtures, resetting, stress, expansion, creep, corrosion protected, noise, re-usable, maintenance, Manufacturability-usability, assemblable ,transportable, automatable ,testable. Materials- small bulk, extensive use of materials, characteristics, reuse Marketability- cost conscious, quality assured, storeable.

Appl'n No.	Application Details
102	**Quality Control:** In terms of controlling quality, there are a number of knowledge - based systems, ranging from on-line quality monitoring, to pc. based advisory systems which allow the factory worker to to do his or her own quality control tasks formerly carried out by skilled technicians. Besides the obvious time and cost savings there would also be a strong likelihood that the worker would be more quality conscious and more responsive to quality as an issue in manufacturing. There are three particular areas where KBS may be considered in GEC-Traction : helping to decide how large a sample of test to apply ;for locating faults and defects; and resolving faults and repair work . KBSs could also help to represent the cost of ' non-quality' in financial terms, say through the effects of poor supplier materials, poor workmanship, bad design and so on. KBSs could also make effective use of historical data on faults and defects to recommend trends and likely failure areas.
103	**PLC Selection:** The range of programmable logic controllers is great because they need to be configured precisely according to the specific needs of an application. An expert system would help the engineer to select the most appropriate configuration of options based on technical specification and cost.
104	**Insulation Tape Selection:** The labs at Preston have a large library on the types of insulation tapes appropriate fro various motors and working environments. A simple rule-based shell could help assist in the selection process.
105	**Warehouse Control:** The MECCA system at GEC-Traction is used to book-in items from suppliers or other sites according to basic similarities- for example, whether they are free issues, or bar materials, electrical components, contactors and so on. When goods are received, the materials number is checked on MECCA to see if there are existing bins already allocated to this number. If not, then new locations are created preferably in a region where similar items are placed. Ideally, when a consignment of items is delivered to stores, it should be placed in such a way that the next items to leave are easily accessible. It is possible to use knowledge-based techniques to consider predetermined parameters such as size and weight, frequency and use and distance from workcentres, and predict patterns of future use so that optimal positioning is attained. A KBS advising on store loading could reduce picking-up times and costs; it could also reduce waste by correctly positioning fragile items and ensuring that older stock is used before new stock.
106	**Supplier Vetting:** Knowledge-based systems offer potential as an aid to purchasing staff to assist in the choice of supplier. This process is a balanced evaluation which requires experience and knowledge . Because of the pressures to find suppliers in a short period of time, the choice is often based upon that supplier who offers the lowest price among a considered few who the purchaser happens to know or have on record. Clearly, price is not the sole criteria , and a rule-based expert system shell would be useful in alerting members of purchasing to other more qualitative factors, which are as equally important, such as reliability, quality, delivery performance and so on.

Appl'n No.	Application Details
107	**Price Forecasting:** Raw materials entering the company vary in price , over an inventory cycle, according to a number of factors most notably the availability/ supply and rate of inflation. Knowledge based systems are excellent at trend analysis whereby forecasts and predictions are made based upon a heuristic analysis of past trends. Typical applications could be in estimating to provide more accurate indices for updating costs: or in purchasing to provide staff with the ability to foresee market shortages and to increase stocks where necessary - particularly with the more expensive commodities such as copper and gold .
108	**FMS Fault Diagnosis:** An on-line diagnostic system would be a useful tool for the FMS operations team. However, this would require a complex array of sensors and monitoring equipment and would be a costly addition to the FMS at Preston. Less ambitious, but still potentially of great benefit, is an off-line diagnosic system which uses a historical knowledge-base of faults, together with users' reports of symptoms to recommend the most likely fault and consequently suggest any corrective action that is required. Already, a modest history of faults has been recorded during trials and testing and trends of faults are arising which are readily representable in a rule-based system. The system could also include categories of faults within each sub-section of the cell so that more focused diagnoses could be made.
109	**Intelligent Design & CAD Support:** Conventional CAD systems reduce the amount of routine work of a designer. They facilitate the design of technical products, for example, because they allow a swift access to data of already known constructions and standard components, and the dimensioning of components aided by calculation algorithms. The real ' design' know-how however, has to be acquired as a long process by the design engineer himself. The know-how is thus specific, bound to the person and difficult to transfer. Knowledge- based techniques offer the opportunity to store and process appropriate knowledge designers require. It is possible to complement existing CAD systems in this way by considering precise knowledge such as design guidelines, and the designers own experiences and heuristics. With existing CAD systems, the selection and dimensioning of the component is a task of the system user. There are a range of design constraints which can be incorporated into a knowledge-base aid for designers: these include:- Precision- fixture tolerances - workpiece tolerances Cost - construction - production - assembly Function - reference to measurement - handling Positioning - operable - without collision Access to workpiece - number of elements - size of elements Fastening - simplicity - safety - standard

continued on page 21

Appl'n No.	Application Details
109 (cont.)	Often the demand for lower costs and good access to the workpiece stand in opposition, as do many other variables which are balanced in a decision made by the engineer based upon experiences in the field over many years. Thus KBSs provide the part of design know-how which can not be represented in conventional CAD systems. This includes three types of knowledge:- a) specialised knowledge about construction elements, materials and design guidelines, b) experience, knowledge and heuristics, like reliable approaches or test clamping methods etc. c) information about the design goals in addition to direct design knowledge. Two notable commercial systsems are available in the UK and these are discussed briefly: a) Power Computing- **Concept Modeller**: This system enables designers to capture their expertise so that future designs, optimised to specified criteria, are rapidly produced more quickly with less effort and with less knowledge of CAD operations. It achieves this by organising the modelling process using object orientated programming techniques , and provides the user with facilities to characterise the model and the parts that make up the model with unlimited properties such as height, weight, construction, materials etc. The user can define rules which will control the operation of the parts from which the model is constructed and how those parts should interact with other parts within the model. The design of motors, control gear, cases etc can be divided into two aspects; the routine work which occurs despite each new design; and more specialised situations resulting from particular client requirements . In both cases, Design make use of design codes of practice, standards and departmental procedures and have, in a less formal context, rules-of-thumb, know-how and other specific expertise. The Concept Modeller is intended to automate these formal and informal types of knowledge. It is designed to be used by 'end-users' with no special training in computing or CAD, and the 3-D modelling capabilities are integrated with a easy-to-use, self-help interface. The main benefits of this system are that it :- i) Speeds the design process(reduces design effort, prevents redesign due to error, allows more timely response to requests for proposals) ii) Reduces design costs (reduces manpower costs, prevents impossible or unrealistic designs from a manufacturing point of view, from being quoted. iii) Allows integration of manufacturing constraints into the design process (can impose constraints of standard parts, shapes and machining operations) iv) Produces consistent designs (prevents two engineers from designing the same product in two different ways- costs are less when designs are consistent!) v) Allows " what-if " type analyses (where many designs can be tried to examine the effects of design changes on cost, and other design effects : only those parameters that change get recalculated, so the effect of changes can be found easily)

Appl'n No.	Application Details
109 (cont.)	vi) Allows integration (of many separate analysis programmes into a single structure. The design of some products may require the use of several analysis programmes written in different languages. The Concept Modeller is able to manage these (simulation, finite element analysis, & general computational systems) including a direct interface with the Computervision CAD stations and Product Data Manager) . vii) Provides documentation for the entire design (because the system performs the design , all of the information that went into producing the design is captured). **b) ICAD:** The ICAD system is intended for the same purpose as the Concept Modeller, that of representing the design knowledge of the engineer in rules, with the same benefits as described above. Engineers use the ICAD design language to define all important engineering and manufacturing attributes about a product design. The resulting Design Knowledge Base contains rules for determining physical and geometric properties, design rules, engineering standards, rules for selecting purchased parts, configuration rules, and manufacturing constraints). The ICAD system allows engineers to evaluate alternative designs, to automate engineering tasks, to capture design intent, and to enforce standards. It can be used to build models to improve the design of tooling and fixtures, to generate and evaluate process plans, to reduce manufacturing-required design changes by providing design groups with manufacturing engineering constraints and to evaluate new machines, fixtures and manufacturing processes.
110	**Tuition & Training:** Knowledge based systems that perform tuition and training help to diagnose, correct and improve the technical competence of staff or trainees. However such an approach can be used for updating mechanical engineers on new processes for instance, or providing an interactive system for potential clients who have received a Tender from Traction to understand more and question the proposal . A second form of instructive system is intelligent help. This is for users who generally understand procedures but may be uncertain in some areas, or may need reminding after some period away from the system. Such systems are context sensitive in that as the user becomes more proficient in using the CAD station, or in programming in Oracle, for example, then less prompts are required and the nature of the consultation may change from being tutorial based to a highly sophisticated task-based advisor. In all areas of Traction business where there is a long learning curve before the person may be considered to be useful, the use of instructive knowledge- based systems can help to accelerate the training process and also free the human expert to perform other tasks.

Appl'n No.	Application Details
111	**Jigs & Fixture Specification:** The design and manufacture of Jigs and Fixtures is perceived as a bottleneck in the manufacturing process. There are numerous parameters to consider such a shape, complexity and process, whether a standard modular or new design is required, the tool material (and location in stores) and machining process. As with jigs, there is also concern over the waste in the use of materials for tools and fixtures. A final problem is understanding the constraints in which the jig or fixture will operate ; factors include the number of faces of the piece to be machined, quantity , size and weight of components to be produced, holding requirements, and machines availability (based upon routings on MECCA). An expert system would be of use in capturing the rules-of-thumb, kmow-how and experiences of the manufacturing planning engineer. The system could become more sophisticated by providing a direct interface with MECCA routings and part information and integration with the CAD system.
112	**Sales Forecasting:** Based upon past experience, this system is intended to help unit managers process , and senior management to evaluate, sales forecasts. An enhancement to this system could be developed to aid P.P. & P.U. to interpret sales information and generate an output profile with Works . A similar process take place with spares orders.
113	**Engineering Planning:** This function is effectively undertaken by a single engineer with many years experience in the organisation. An expert system could be used to assist and archive the skills of the engineer in four particular areas:- a) establish time-scales for the engineering activity at the enquiry stage, b) liaise with project planning to produce overall project plans c) establish workload programmes for contracts to achieve project plans d) monitor future resource requirements
114	**Engineering Costing:** One of the main problems in generating a price estimate is the assessment of engineering costs. This is based on substantial knowledge gained over years of experience on patterns of costing and trends in machine estimating. This expertise could be captured in a rule-base.
115	**Engineering Substitution:** engineers are a scarce resource and it is important that they are utilised effectively. However a substantial portion of the engineer's time is spent in information gathering and undertaking standard and procedural checks. By substituting these tasks to less skilled staff, or improving the efficiency of the information gathering process, the engineer can perform those tasks which are commensurate with his skills and qualifications. Expert systems play a useful role in both instances.

Appl'n No.	Application Details
116	**Sales Advisor:** With the exception of spares, selling is not a well defined activity within Traction, which makes it difficult to define customer requirements precisely. This makes the effective communication of requirements between TPL Commercial departments and Traction Contracts critical. Knowledge based systems can be used to effectively amplify customer requirements through explanation, help and advanced interface facilities. Knowledge-based systems can also be used to advise on procedures and information requirements between TPL and Traction.
117	**Statistical Advisor:** Middle management receive large amounts of data on performance, costings, quality and attendance. Various statistical techniques and computations are performed and management often act upon these figures. A system could be developed which undertook the correct computations based upon the spcified needs of the user and interpret this information and recommend actions.
118	**Programming Productivity:** Upon analysis of computer programming, it is evident that most new programs contain parts from previously written files, libraries and subroutines. Access to a knowledge-base would advise the programmer, upon receiving directions of intent, what has already been written and how particular files could be modified and included in the latest program. This would improve the productivity and quality of new programs.
119	**Recruitment Advisor:** Recruitment is a process of identifying desirable attributes required of a person in order to perform a particular job description and matching these against the performance of people during an interview. Numerous techniques and procedures together with the expertise of the interviewer combine to evaluate a candidate. An expert system could provide guidelines on the attributes sought for particular jobs and offer advice on conditions of employment, legislation and so on. This would help to standardise recruitment and improve the selection process.
120	**Modelling Electrical Circuits:** During the design layout process, a knowledge-based simulation system could be used to assess the effects of current, voltage, loading etc. upon the new design without the need to perform destructive tests. (Also see No. 19 and 88).
121	**Training Advisor:** A system to advise on the training requirements of new employees, select appropriate training courses, and schedule inductions, in-house training and other developments. The system could also incorporate a monitoring and assessment procedure which included appraisals and pursuit of professional examinations.

Appl'n No.	Application Details
122	**Repair & Overhaul Management:** The various machines which return to the Preston site for repair and overhaul require different test and maintenance procedures, skills and materials. This system would schedule a plan for maintenance which would include deadlines, testing and spares.The system may also be used to design a maintenance plan for the client.
123	**Spare Parts Advisor:** Based upon experiences of similar past requirements, this system would select and configure the arrangement of spares which would be required by a particular client based upon type of engine and working conditions.
124	**Pre-sales Promotion:** Expert systems are an effective medium for promoting products and services. An expert system could be useful in promoting the capabilities and attributes of GEC-Traction and TPL in addition to providing technical advice on contract conditions or specification of equipment. The use of intelligent user-interfaces, explanation facilities and menu-driven functions provides an informative communication medium whilst promoting a " leading edge " company image.
125	**Knowledge-based Inspection:** Inspection is a critical stage in the manufacturing process at GEC-Traction as in all engineering firms. The increases in product accuracy and tighter limits on geometric tolerancing to meet tighter design specifications require increasingly more effective inspection planning and execution. Knowledge based techniques could be used to provide a generative (feature based) inspection task planning system which would help the engineer in the following areas:- a) interpreting engineering drawings to arrive at inspection criteria b) Decision-making regarding the inspection procedure given available inspection facilities. c) executing the inspection plan. This would also help to formalise inspection procedures: presently, inspection is an informal exchange of information between operator and inspector. The inspector keeps no records of how much machines are out, for example, and the operator is given no incentive to maintain his own machine. However, a knowledge based system would help the operator to detect symptoms of faults (based upon limited records of faults) which would suggest various maintenance actions. This proactive approach is more cost effective than preventative maintenance or the present state of " crisis maintenance " and allows the operator to become more involved directly in the maintenance of the machine.
126	**Maintenance Stock Control:** The spares store within maintenance , re-order stock on a purely random basis with no apparent analysis of needs. An expert system could help to advise on the re-order types and quantities based upon more rigorous criteria than presently adopted and more in line with company stock control practices.

Appl'n No.	Application Details
127	**Welder Specification:** There are a number of areas where expert systems would be of use in welding . These are:- a) Welder selection. A system could be used to ensure that the right welder with the correct qualifications was available to do the required job. The system would also monitor when the welder should be tested next. b) Procedure Generators. The operator is prompted for information about the joint to be welded, and, using rule-based inferencing techniques, the system produces a suitable procedure. For example, the input information required from the user may include material thickness, material composition, joint type, and joint position; and the output could include welding process, pre- and post-heat, consumables and welding parameters. c) Process, Consumable & Equipment Selection. several expert systems have been designed to specifically deal with the problem of selecting the optimum process, consumable or equipment for a given welding application. Similar inputs are fed in as with the procedure generator, but the system may also incorporate process economic and cost effectiveness, risk evaluation, equipment diagnostics and defects analysis. d) Costing the Welding Process. This would help to arrive at some cost in pence per metre or per minute of weld by considering a number of factors in addition to materials and labour costs. These include:- i) productivity ii) quality of weld iii) distortion costs (through rectification) iv) grinding-off costs v) post-weld heat treatment costs vi) preparation heat treatment costs vii) depreciation on welding equipment viii) other overheads
128	**Facilities Layout:** When MECCA schedules parts to machinery,it fails to assess the practical implications of facility layout and plant requirements. For instance, whether there is sufficient compressed air or gas supplies at this location, or the weight of the crane to move the piece, or the width of the passages. An expert system would be a useful enhancement to MECCA by recognising the constraints one needs to consider in the scheduling of work to machines, and also the location and relocation of plant. This could be assimilated in cost terms so that a decision to move a machine for a new contract for example, could be quantified with a particular cost-benefit- there would be a cut- off rate at which it would be uneconomic to schedule work to certain machines or the converse. The system could be built quickly using rule-based inferencing techniques. (See also Applicatio numbers 21 and 99).

Appl'n No.	Application Details
129	**Tender Decisions:** The decision to go for a contract is highly complex and involves technical, economic, business, organisational, strategic and political issues. For instance, the Docklands contract was necessary in order to develop company expertise as much as it was to make money. A knowledge-based advisor would help to highlight such strategic and long-term issues, and provide a suggestion of whether to place a tender based upon a detailed, weighted analysis of the client and conditions in question.
130	**Customer Specification:** The response and dissemination of customer specifications and contractual requirements by Traction has been criticised on a number of occasions. This is attributable to the enormous complexity and volume of information processing required in order to assess requirements and implications before a response is drafted. This task has two aspects: the first is attending to new work specific to a particular contract; however a second is routine work which may have been done in the past and requires slight modification to suit the needs of the new contract. An expert system would be useful in the latter to represent, once-and-for-all, these routine decisions and forms of expertise, so that the response to customer specifications would be quicker and the quality of the constructed tender would be improved.
131	**Capital Commitment to Contract:** During the evaluative process of assessing the potential benefits of a contract with a view to drafting a tender, the commitment of capital to the contract is an important strategic and economic issue. An expert system could help capture the skills of commercial management in TPL whose expertise is used to decide how much capital ought to be tied-up . This is balanced against the expected returns accruing from the contract from which a recommendation to proceed is based. Such a system could also be extended to minimise the capital commitment to contract before and throughout the contract's lifetime.
132	**Financial Pack Collator:** The financial pack dictates events in GEC-Traction.It provides a commentry on contract status, sales in months, orders, profit, direct wages & salaries, costs of materials and so on. Although the format of the pack is quite acceptable, some of the procedures used to put it together are laborious and slow. The process involves collating voluminous amounts of data and making routine-decisions based on well-known procedures and accounting practices. This application therefore, lends itself very well to an embedded expert system: more specifically, a rule-based shell such a s Crystal which is able to interpret data from software such as Lotus 123 and DbaseIII, as well as mainframe database systems, in order to provide " smart" information as it is required in the financial pack. The system can also be extended to provide recommendations on the basis of the financial pack and the way in which this is interpretted by senior management.

Appl'n No.	Application Details
133	**Customer Feedback:** It has been pointed out that there is little feedback on the performance of Traction machines in service. This is a waste of a valuable source of ideas and comment. Used correctly, client feedback could have ramifications upon design, manufacture and the way in which projects are approached in future. Expert systems would be a useful mchanism by which to generate customer feedback by facilitating an interactive consultation during the first few years of the machine in service.
134	**Strategic Planning:** One of the notable conclusions arising from the IDEFo study was that long range planning, in terms of business development, product development and general process development, was seen by many to be ill-structured and ill-considered. This is made worse by the nature of the business which is based on the fulfillment of large contracts tailored to the needs of individual customers, and the short range appraisal of the company's performance based primarily on accounting measures, like the financial pack. It is worthwhile considering whether knowledge-based systems may be of use in corporate planning support for the appraisal of product markets, economic forecasting, long-run financial evaluation, and technological and manufacturing systems evaluation. The purpose of such a system would be to inform and advise senior decision-makers of the relative strengths and weaknesses within the company, and opportunities and threats originating outside.
135	**MRP Listings/ Exception Listings:** Both require the expertise of an experienced employee in order to confirm and filter relevant information fromlengthy listings- this process can take up to two days and is an evident bottleneckin the re-ordering of stock , in the case of the MRP listings, or the rectification of anomalies in planning in the case of exception lists. A useful process would be to capture this expertise in the form of rules to provide an intelligent filter between MECCA and the user.
136	**Expense Approval :** A simple shell- based application which would help finance to appraise and approve expense forms submitted by the various departments within the company. It would make use of existing procedures, but also offer explanations and may be used to automatically generate response forms which could recommend acceptance or refusal or request modifications.
137	**Prototype Development:** Prior to full scale manufacture and assembly, new designs of electrical equipment are first hand built under laboratory conditions. During this process, tests are made concurrently which may reveal certain problems and performance characteristics. From these symptoms, the experienced engineer is able to diagnose faults and undertake specific debugging procedures. In addition , he will recommend special manufacturing requirements as as result of these tests- it is these rules- of thumb which may be captured in a rule-based system.

Appl'n No.	Application Details
138	**Research & Development Costing :** Research and Development in the organisation should be viewed as an investment and therefore as a strategic issue. The decision to follow a particular route of research is influenced by a number of issues: cost , state of the market , exclusivity of the technology , potential benefits of the technology, design requirements , competition, client requirements and many more. An expert advisor using weighted averages could measure scales of emphasis upon each factor to generate a research agenda or development plan as a stimulus for further management discussion.
139	**Simulation Front - End :** Simulation software is used in a number of areas in GEC from the simulation of loading upon engines and electrical circuits to the design of equipment modules . The outcome of such simulations require expert interpretation and application for effective use. An intelligent interface could automatically interpret results to provide the User with advice ; or alternatively provide sophisticated help and query facilites.
140	**Drawing Conventions Advisor:** This would have a dual role of ensuring that designers and draughtsmen adopt proper industry and company standards and procedures; and of helping Technical Planning , whose role it is to translate drawings into a plan for manufacture, to interpret the drawings correctly and with ease. The system would take the form of a rule- based shell operating on a personal computer with a graphics system to communicate drawing symbols and conventions.
141	**Transport Scheduling:** The purpose of this system is to monitor the progress of company pool cars and goods vehicles for effective scheduling and utilization. It could also include an advisor on when to pay tax, insurance, etc ; how to apply company procedures on the use of vehicles; and how to cost milage, depreciation and so on for different types of vehicle and different users in the organisation.

Applications Portfolio [*]

Update: Issue No.4.

March 1988

---

[*] Peter Holden, Cranfield Institute of Technology

Appl'n No.	Business Function	Application Details
142	5a	**Train Logging System:** This is a fault logging system which offers different levels of use according to the requirements of the user. The system was developed by Cambridge Consultants on a tool called MUSE. MUSE makes use of signal processing technology to identify variances which indicate characteristic faults in the operation of the train. This information is presented to the train driver, via an intelligent interface, as a series of three lights: the first light is continually lit and indicates to the driver that no fault is present; the second tells the driver that there is a minor fault which should be reported (although he need not know what the fault is) at the end of the shift; the third light tells the driver that there is a serious fault and that he should stop at the next station if possible and report the fault. The second level of interface is presumably at the depot where service or maintenance engineers remove the fault log (for a system used by London Underground, this is in the form of a 31/2 inch disk) and undertake diagnostics and repair, using MUSE at a different level of enquiry. At a third level of enquiry, central control in the case of the London Underground, MUSE is used to perform trend analysis and performance monitoring over an extended period.
143	4d, 7e , 8e	**Defining MECCA Parts Parameters:** The process of loading parts onto MECCA involves a hierarchy of procedural decision-making. The parts parameters are then used for engineering, MRP and stock computations. Specifically dealing with the MRP module of MECCA, plannersdecide whether the part is a manufactured good or a bought out item: if the former then whether it is booked to contract or to stock and so on. Although this information is checked again by MRP planners, it would save time and resources if this process could be verified at the time of input through the use of rules. A rule-based system operating interactively with the user on the MECCA system would apply MRP constraints, presently used informally and represented only in the minds of the planners, on parts headers , structures and routings for example. It would formalise and archive this decision-making process and also promote consistency and accuracy throughout the company as well as improving the rate at which this task is caried out.
144	7c, 7d, 6e	**Safety Support & Auditing:** In a number of areas in GEC Traction, but particularly personnel and works services, there is considerable expertise on subjects such as emergency response procedures, materials handling, operations hazards and so on. This knowledge could be captured and used as an advisory system for non- specialised staff. Expert systems could also be used for safety auditing, since such an undertak-

Appl'n No.	Business Function	Application Details
144 (cont.)		-ing requires a qualitative and judgemental approach in the analysis as well as secondary tasks such as drawing up an interviewing schedule and providing on - line and context sensitive help facilities.
145	1a , 3a	**Strategic Marketing Planning:** A product of the Alvey scheme and now pursued by A.I. limited, is an expert system for strategic marketing planning. The product is an interactive system that supports a marketing planner by enabling the market and products to be represented in a logical model and interpreted, leading to a better understanding of the business and an appropriate course of action. (Contact is Hugh Wilson A.I. Limited).
146	3b , 4b	**Reliability Testing :** Reliability is a central issue in the design of electronic equipment for example. An expert system has been used successfully in companies in which the engineer outlines a design and the system specifies the reliability requirements at the component or system level. Coversely, the engineer specifies reliability requirements and the system suggests components or configurations which achieve these requirements.
147	6e , 6d	**Coating Selection :** This is a system intended to assist in the selection of paints and surface coatings for frames, cases and electrical equipment housing. The selection of the ' correct ' coating is often difficult because of the numerous factors that should be considered if the ideal coating is to be selected and correctly applied. Such factors to be considered are base material, finish of surface to be coated, the surface hardness required, the structural condition of the bas material, the operating temperature of the application and other environmental conditions.

Appendix VII

A Decomposition of Expert Systems Applications by Functional Domain & Business Impact

Key to Figures in Appendix VII

a) Hardware Needs:-

P.C.	IBM Compatible Personal Computer
WS	Workstation
AI.WS.	Artificial Intelligence Workstation
MF.	Mainframe Computer

b) Software Needs:-

S.	Expert System Shell
L.	Artificial Intelligence Language
T.	Toolkit
E.	Environment
A.S.S.	Application Specific Software

c) Integration Needs:-

SA.	Stand-Alone
L/E.	Linked or Embedded
FI.	Fully Integrated.

d) Company Function Classification

i) Front-end Services : -	Strategic Management Tendering Sales & Exports Contract and Project Management Commercial and Spares Ordering
ii) Engineering:-	Engineering Planning Applications Engineering Systems Engineering Machine Engineering
iii) Pre-Shop Activities:-	Purchasing Estimating Industrial Engineering Scheduling and Planning
iv) Operations:-	Manufacturing Services Manufacturing Operations Works Services Materials Control
v) Services Support:-	Finance Computing Quality Assurance and Standards Personnel

Figure A: A Classification of Expert Systems According to Perceived Business Impact

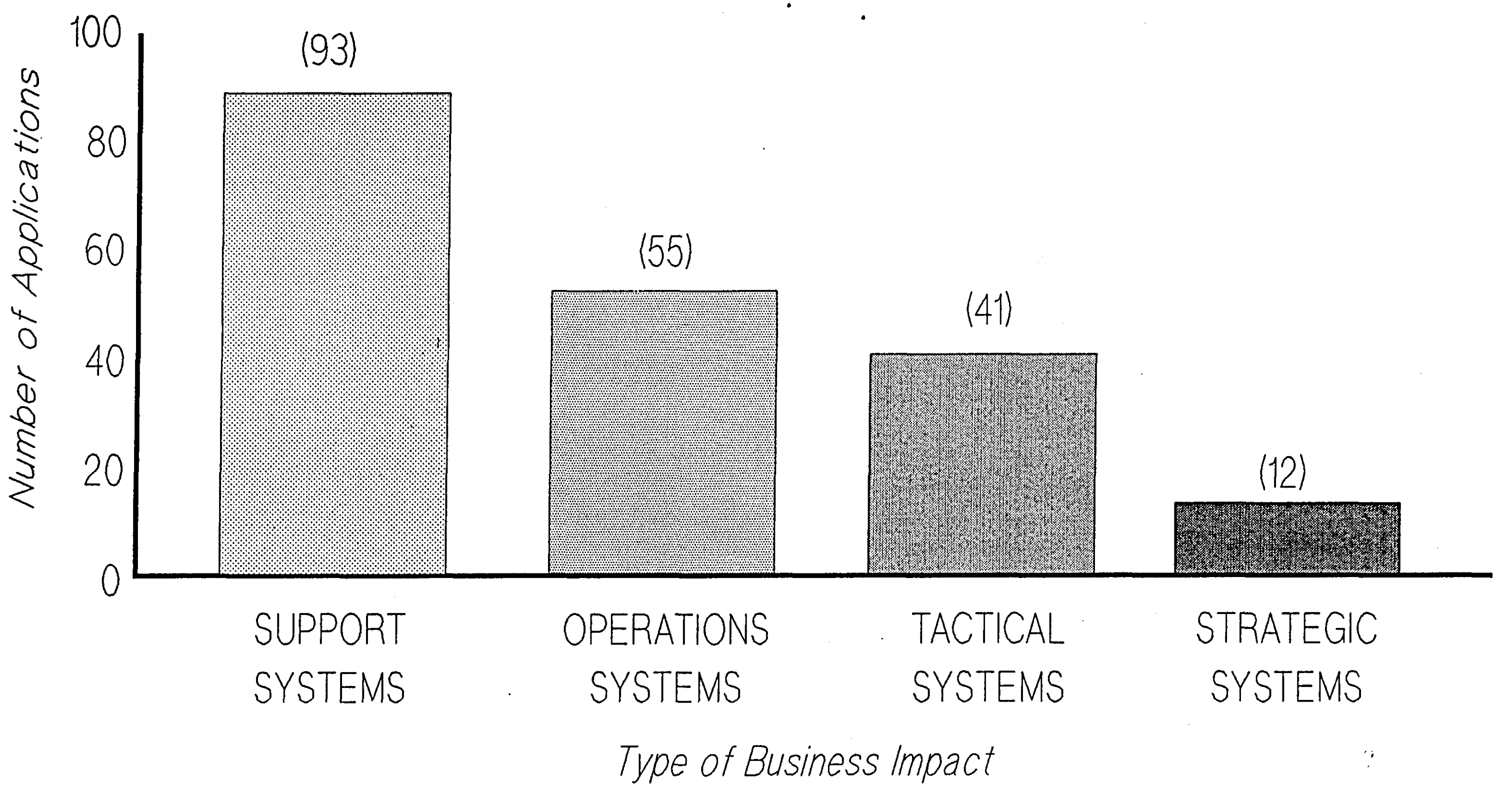


Figure B: A Breakdown of 'System' Type Expert Systems By Company Function

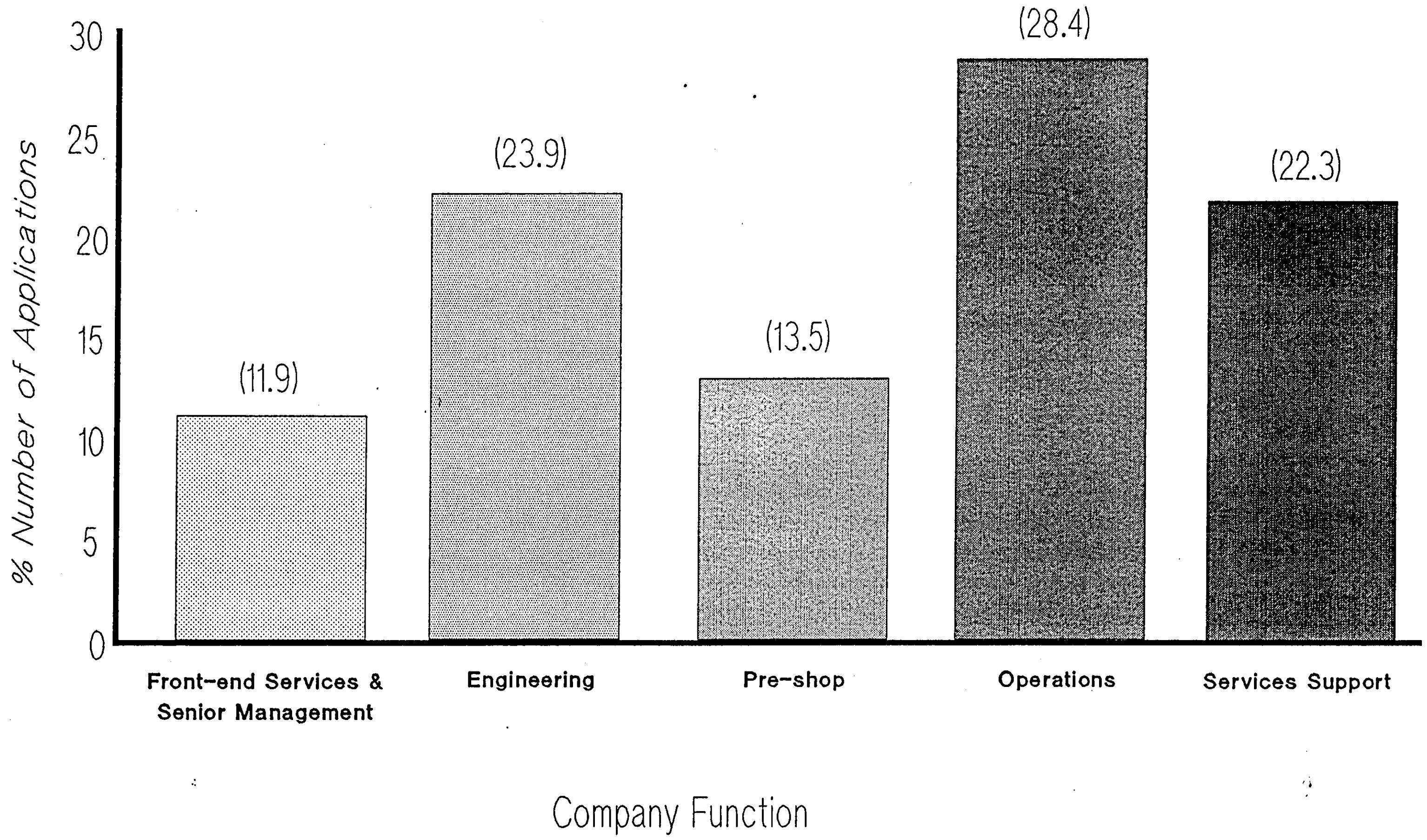


Figure C: An Analysis of 'System' Type Development Requirements

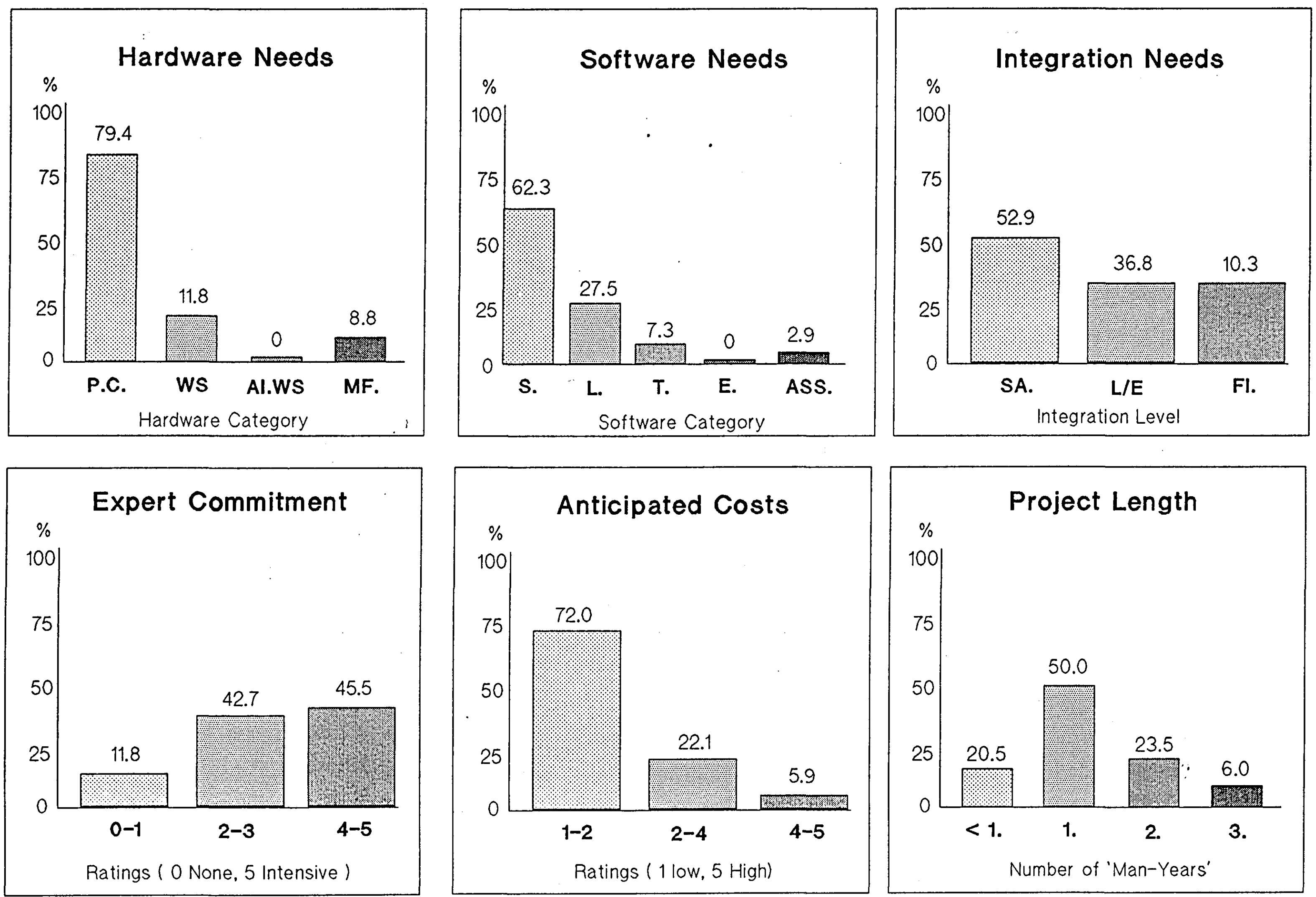


Figure D: A Breakdown of 'Operations' Type Expert Systems By Company Function

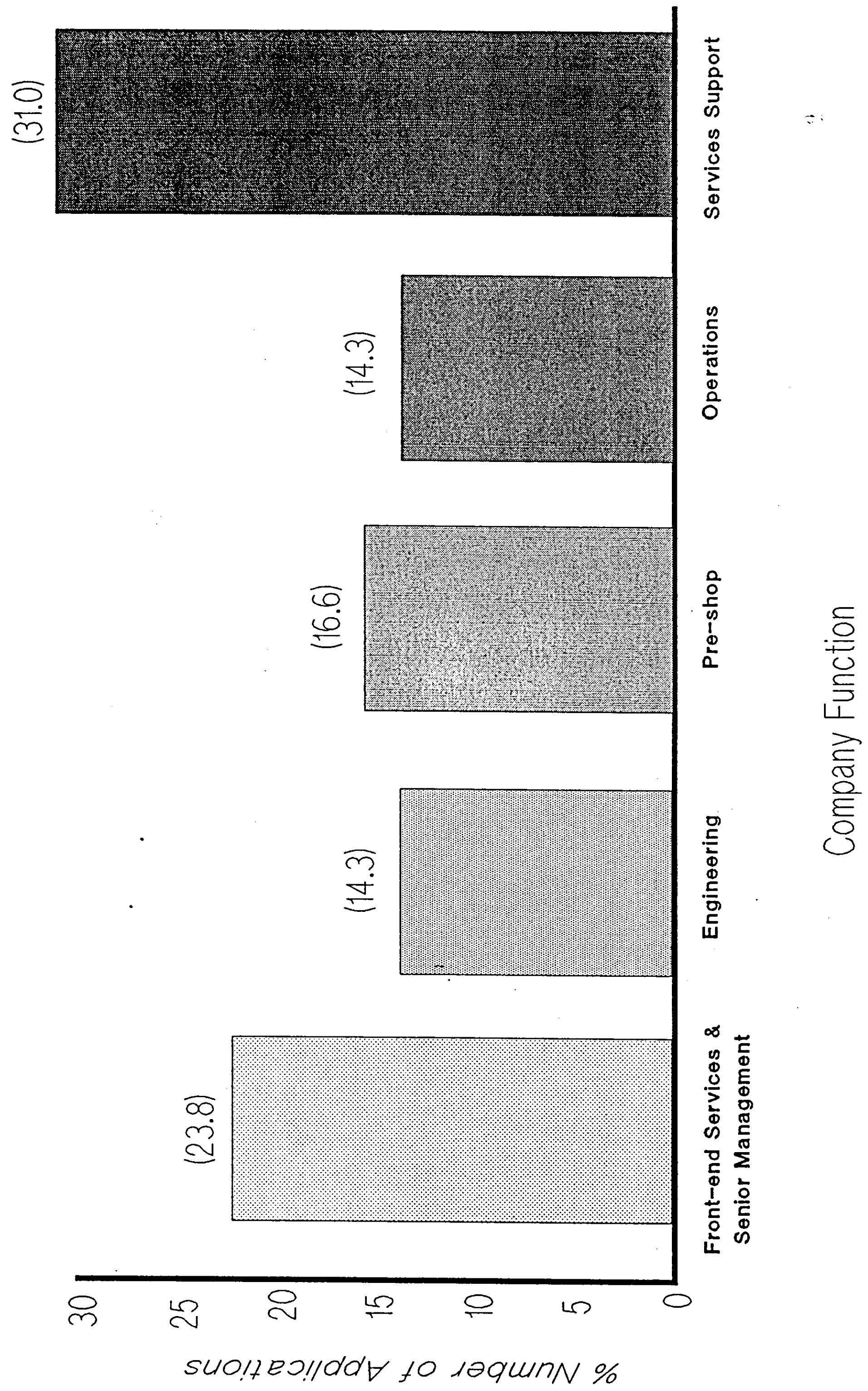


Figure E: An Analysis of 'Operations' Type Development Requirements

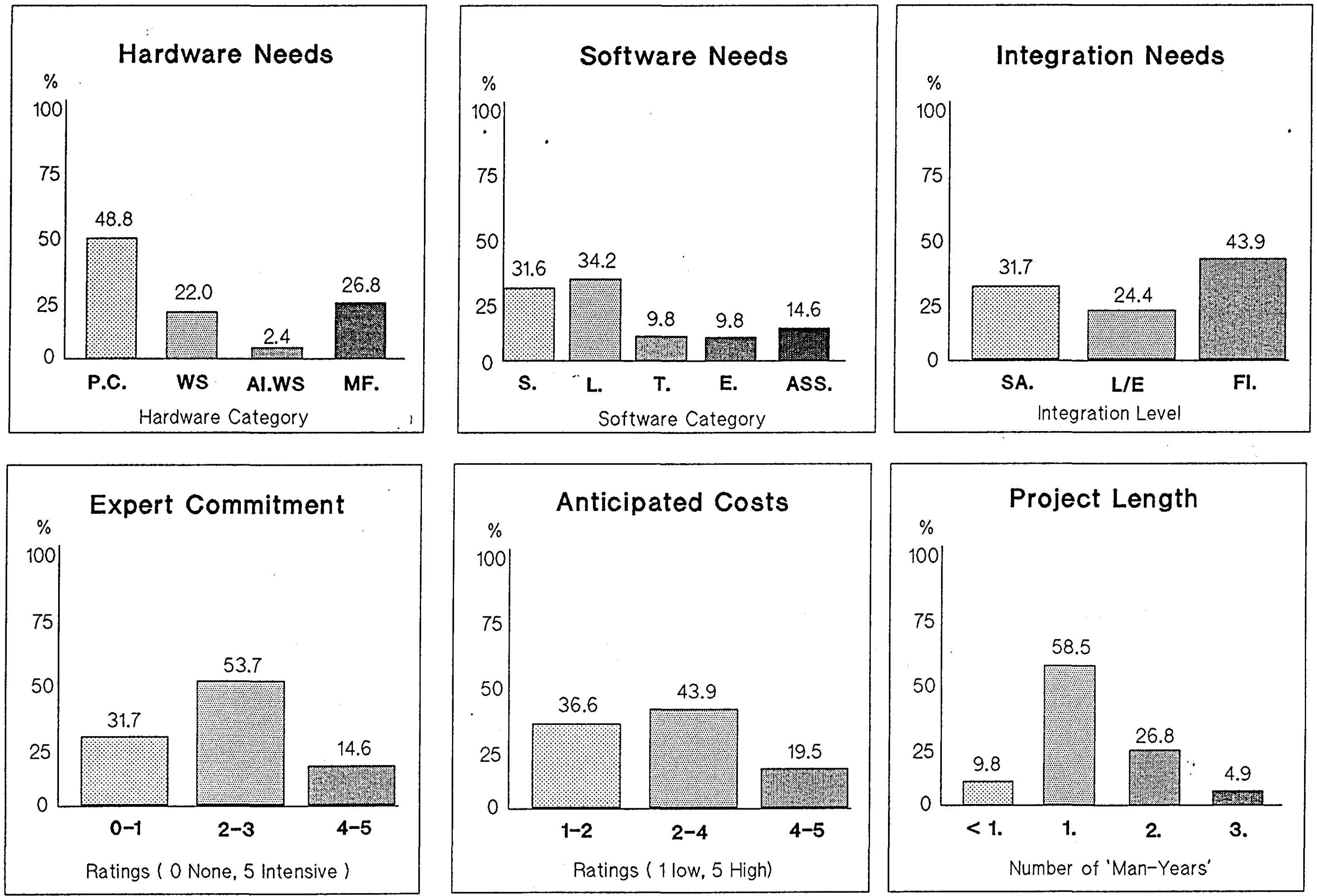


Figure F: A Breakdown of 'Tactical' Expert Systems By Company Function

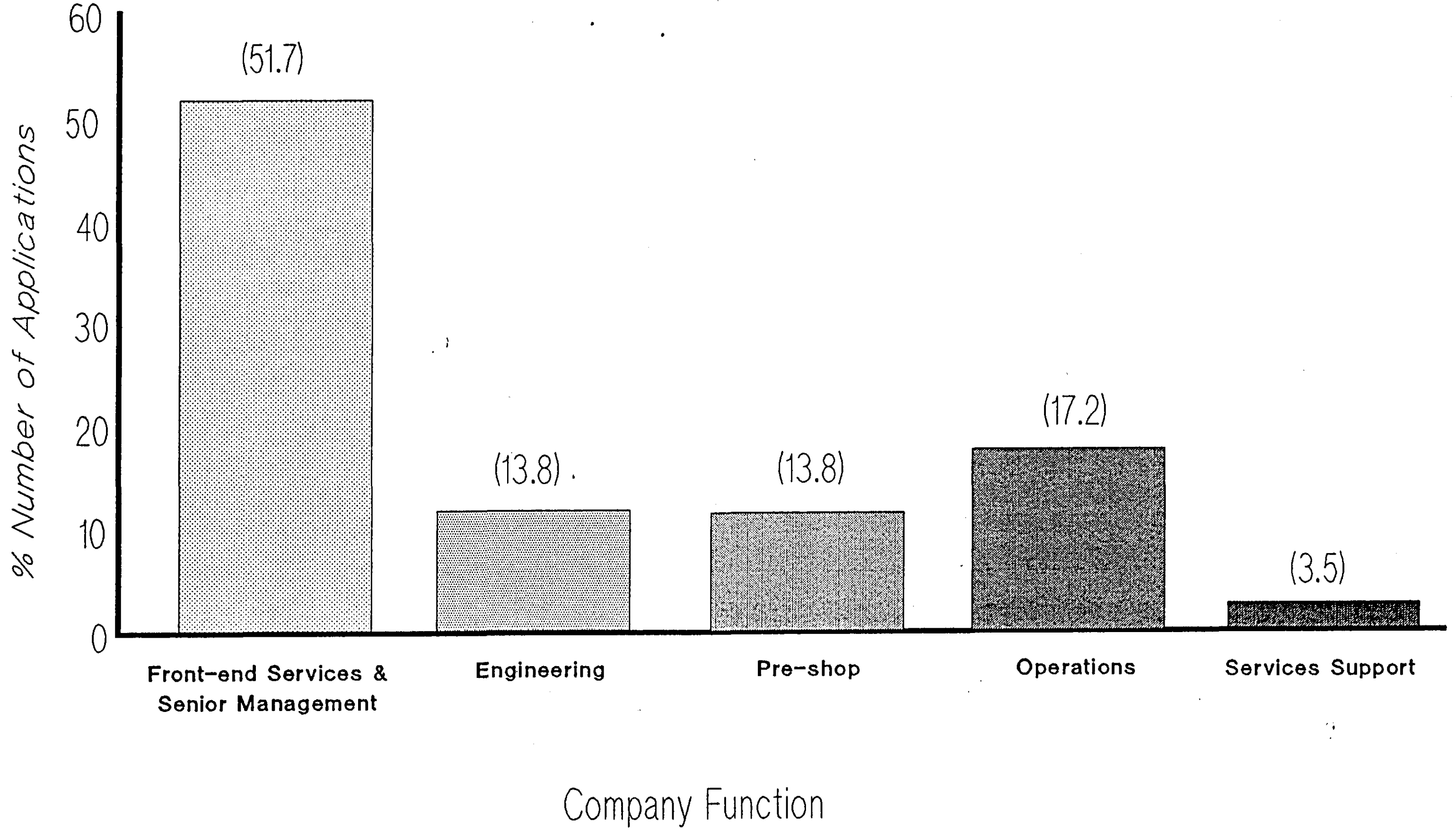


Figure G: An Analysis of 'Tactical' Type Development Requirements

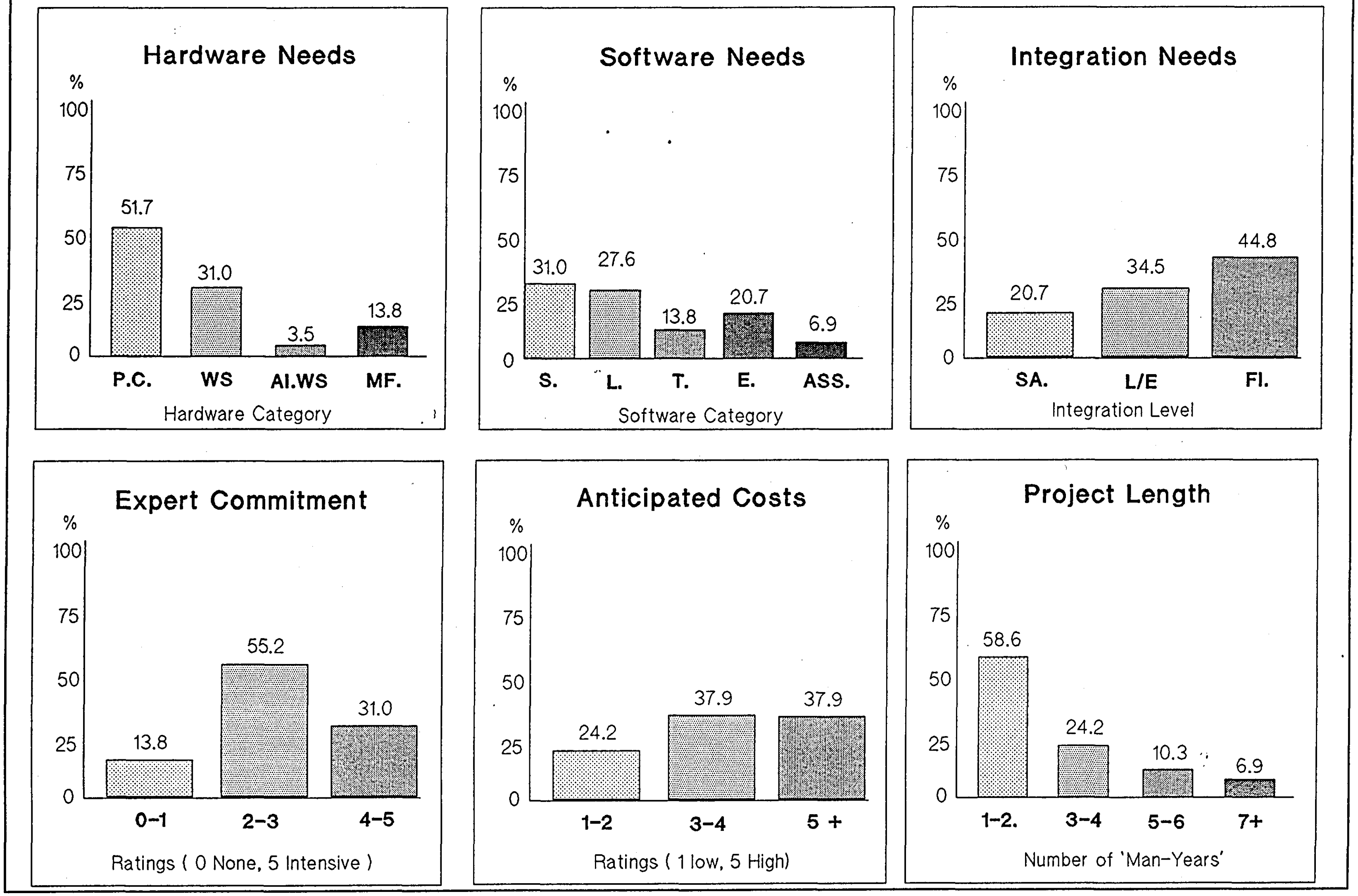


Figure H: A Breakdown of 'Strategic' Expert Systems By Company Function

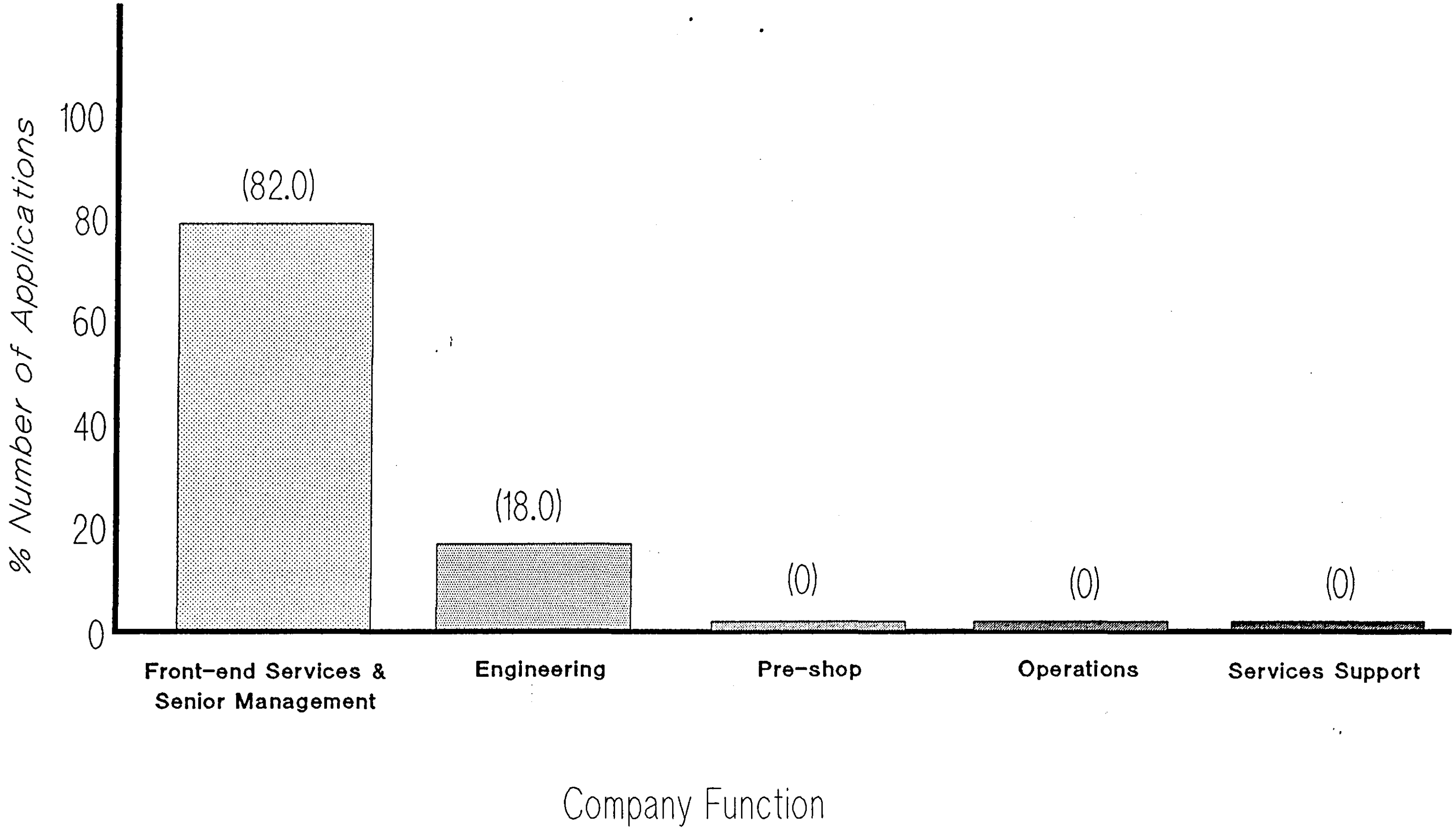
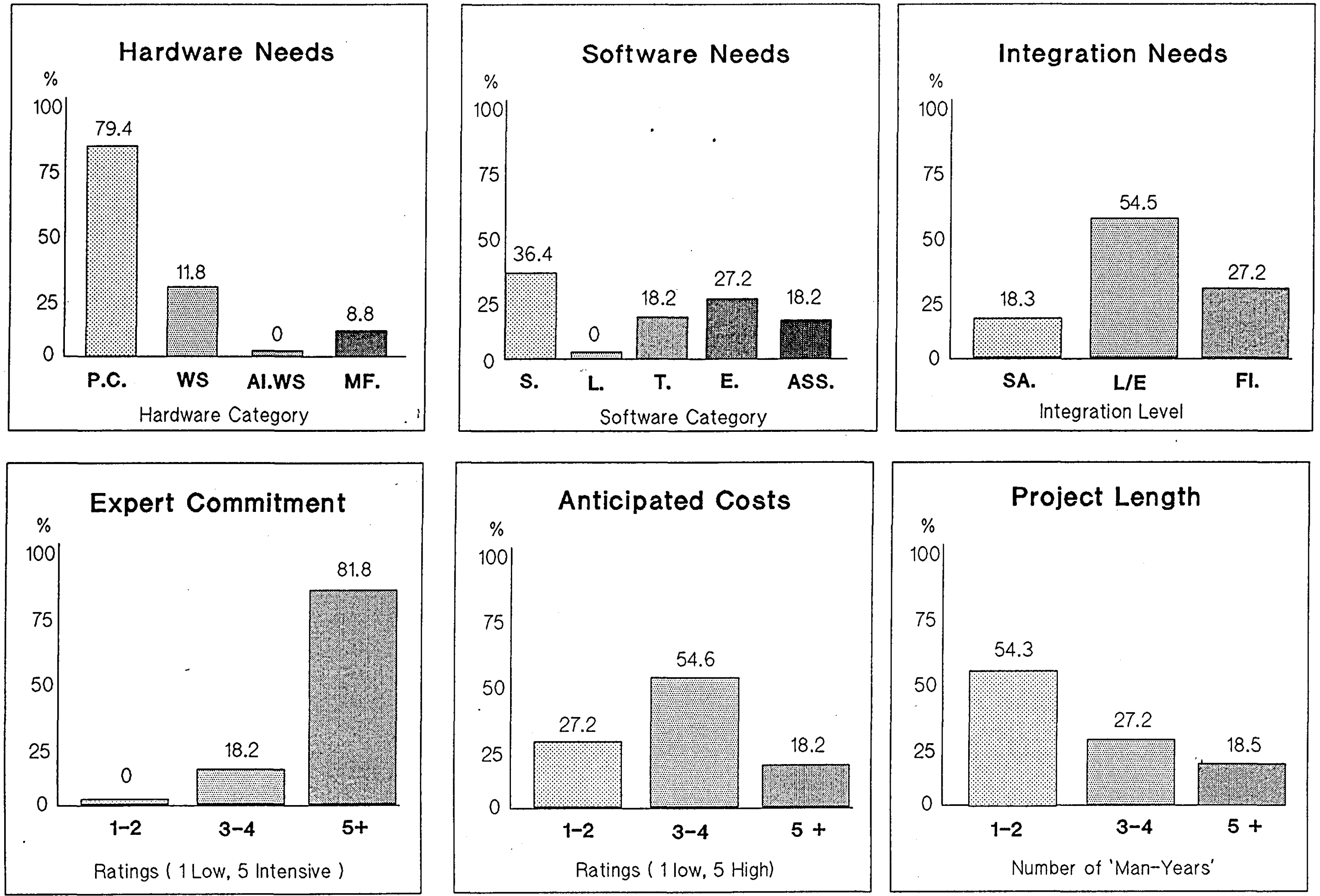


Figure I: An Analysis of 'Strategic' Type Development Requirements



Appendix VIII

Evaluating Potential Expert System Applications

Contents:-

Appendix VIII a.

A Description of Short-listed Applications

Proposal A_1: An Expert System to Configure Spares & Repairs

The company has a set of generic locomotive engine classes which span over 100 years of development, all of which continue to require spares and repair services. Based upon experiences accumulated over these years of the spares ordering quantities and repair requirements, the proposed system would select and configure the arrangement of spares which would be required by a particular client, based upon the engine specifications and working conditions.

The systems could be enhanced in two ways:-

> i) Machines which are returned to site2 for repair and overhaul require different test and maintenance procedures, skills and materials. This system could schedule a plan for maintenance which would include deadlines, testing procedures and spares inventories. The system may also be used to design a maintenance plan for the client upon issue of the product.
>
> ii) The system could also be used in maintenance stock control. Presently, the spares store within maintenance re-order stock on a random basis with no apparent analysis of needs. An ES could help to advise on the re-order types and quantities based upon more rigourous criteria than presently adopted and more in-line with company stock control practices.

Proposal A_2: Expense Claims Adviser

An expert system to help finance functions in the company to appraise and approve expense forms submitted by various cost-centres. It would make use of existing company procedures, but also offer explanations and may be used to automatically generate response forms which could recommend acceptance or refusal or request modifications.

Proposal A_3: A Maintenance Adviser for Flexible Manufacturing

A Flexible Manufacturing Cell has recently been implemented at the company's Preston site. The cell was built and installed by Italian contractors and comprises two machine tools, a vertical turning lathe, a vertical machining centre and an automatic machine handling centre. The on-going installation of the FMC highlighted the dependence of the company upon Italian service engineers in fault trouble-shooting and repair work. The proposal was for an expert system which would capture this expertise for use by company maintenance engineers and also provide a systematic method of recording faults and maintenance suggestions as company expertise in the FMC matured.

Proposal A_4: An Aid for Jig & Fixture Design

The design and fabrication of jigs and fixtures in the company was viewed as a bottle-neck in the manufacturing process because the expertise to design jigs was restricted to only three manufacturing planning engineers, one of whom was about to retire. The basic cost of a jig was £500 and it was considered that this sum could be reduced by minimising the amount of materials wastage and the time taken to configure designs to meet machine specifications. An expert system was suggested by the engineers themselves as an effective means of distributing this knowledge and providing a useful training aid to newly recruited graduate engineers. It would also improve their own productivity in configuring appropriate designs. It could be further developed by providing a direct interface with the routing and parts information database and, looking to future prospects, linking design to manufacture by a CAD-CAM link.

There are numerous parameters to consider in the construction, of jigs and fixtures, such as shape, complexity and process variables; whether standard, modular-based or new designs are required; and the tool materials necessary to meet working conditions. A final factor in this design problem is the operating constraints of the jig/fixture. These include: the number of faces of the piece to be machine; the quantity of components to be manufactured; the weight and size of the components to be manufactured; specific holding requirements; and machine availability (based upon routing from the company's production planning and scheduling database).

Proposal A_5: A Shipping Adviser for Manufacturing Services

This was a proposal for an expert system to advise on the distribution of finished goods according to company regulations, legislation and the specific requirements of the host client. Factors in this planning system include country of origin; routing; stops; diversions; due-dates; speeds; regulations; capacity; weight; volume; costs (milage, loading weightings; double-ups etc); packaging .design standards and techniques; and materials.

Proposal A_6: A Wire and Connector Configurer

The company's hardware specifications for electrical connector assembly are found in many pages of documentation and standards. The location, interpretation and application of these standards is a time consuming and laborious process. Specialists in the design of electrical circuits estimated that over 40% of their time was spent in such information gathering exercises. An expert system could assist the engineer in configuring wires and connectors and thereby reduce the search time: it may also prove viable with such a system to delegate these tasks to less specialist personnel.

The system may be enlarged to configure resistors, wiring, meshes and other electrical components. In addition to specifications, the system would provide guidelines on the most effective arrangement of wires and connectors for certain requirements in circuit boards and machines.

Proposal A_7: Computer Hardware Fault Trouble-shooter

There is a wide scope for fault diagnostics in computing within the client company, reflecting the company's dependence upon information technology to support business operations and a serious shortage of specialist staff to manage these systems. Furthermore, the turnover of staff in this area is notoriously high. An expert system could help in a number of ways:-

i) archive skills and knowledge of current computing systems

ii) improve the distribution of expertise throughout the company

iii) improve the productivity of computer specialists in solving problems

iv) improve the computing service offered to user departments

The scope and organisational commitment to ES in this field range from small-scale, to help experts diagnose specific computing systems for example, to company wide where an ES might be used to allow users of company databases to perform self diagnostics. Although at this stage a specific project had yet to be defined, a number of areas were identified which satisfied 'first project' criteria and had generated strong support from people from within and outside the computer department.

Proposal A_8: Vibrational Analysis

Of the ten short-listed applications, this suggestion was the most complex, technically, and was motivated by ideas originating from the vendor of an 'application specific' expert system rather than from within the organisation. The system called 'VIOLET (the Vibration Order List Expert) is designed specifically for vibration-based machine-health monitoring. It was proposed that this system should be applied to the company's combined test facility which is used to test locomotive motors before issue or to identify the faults of machines recalled for repair.

The system's knowledge base is made up of a history of vibrational test which may be used to predict which machine parts are likely to fail during operations, or act upon this information as a controlling function. Unlike other applications, this proposal is on-line and utilises frequencies, time signals, shock impulses, amplitudes and harmonics and other data representations from the motor's rotor and shaft sensors in order to make a prognosis. The system is intended to provide assistance to the specialist engineers in interpreting such vibrational patterns, measures and signals rather than acting as a substitute.

Proposal A_9: A Capital Investment Appraisal Adviser

This application has a dual role: first, to help technical engineers with no specialist expertise in finance and capital investment to be able to price and fully justify a capital investment proposal using a standard format of presentation which is effective in highlighting both the benefits and limitations of the proposal. At a later stage, it was hoped to enhance the system by providing a second function which would help evaluators of capital investment proposals to assess their worth using existing financial tools together with qualitative and longer-term business criteria.

Proposal A_10: An Expert System for Weld Selection

The company makes a large number of special order, one-off components to which it employs a wide variety of welding processes. The potential for an expert system for welding is threefold:-

i) Process, Consumables and Equipment Selection: The operator is prompted for information about the joint to be welded, such as material thickness, material composition, joint type and joint position; from which the system using rule-based inferencing techniques provide advice on the type of weld to be made, the consumables to use, the pre- and post- heat treatment settings and other welding parameters. Given the limitations in scope defined by 'first project' criteria, the system proposed could only perform this selection function. However, two future improvements are possible.

ii) Welder Scheduling: The system could be enhanced to schedule welders according to the specialist skill requirements of a particular welding job. Each welder must have a minimum number of qualifications before he is eligible to perform certain welding tasks and, furthermore, is expected to undergo re-assessment after which the qualification is re-issued for the forthcoming year. The scheduling problem is thus one of allocating welders with the correct and current qualifications to commensurate welding tasks.

iii) Costing the Welding Process: A further enhancement would be to provide a costing of proposed welds. This would be in pence per metre or per minute of weld and would take account of the following factors in addition to materials and labour costs:-

a) productivity of the welder
b) distortion costs (through rectification of the weld)
c) grinding-off costs
d) quality of weld
e) post-weld heat treatment costs
f) preparation heat treatment costs
g) depreciation on welding equipment
h) other overheads.

Appendix VIIIb: An Exhibit of the Feasibility Check-list *

1. Definition of the Problem

1a. Does the problem have a high proportion of common sense reasoning ?

1b. How frequently does the problem occur ?

1c. How easy is it to collect information about the problem ?

1d. How is information in the problem domain represented ?

1e. Is the problem domain narrow and identifiable ?

1f. How valuable is the problem domain to the organisation ?

1g. What is the perceived value of the problem domain to users/experts ?

1h. Would there be a demand for a solution to the problem ?

1i. How appropriate/applicable are conventional programming methods ?

1j. Does the problem require the use of heuristics or rules of thumb ?

1k. Is the amount of knowledge required to solve the problem large ?

1l. Is the problem domain stable ?

1m. How well is the problem understood ?

2. Validation of Expertise

2a. Does an expert exist ?

2b. Is there more than one expert ?

2c. Is the expert(s) chosen available to participate in the proposed project ?

2d. Is the expert(s) willing and co-operative ?

2e. Is the expert(s) able to articulate the nature and extent of expertise ?

2f. Do experts agree upon a solution ?

2g. Is the expert(s) accredited as being an expert in the field chosen (both within and outside the problem domain and by management) ?

---

* *An explanation of each question is given after the check -list*

Feasibility Check-list Continued...

2h. What makes up the expertise :-

- technical knowledge (explicit) ?
- practical experience acquired over a long-period(implicit/tacit) ?
- the use of physical as well as mental skills ?
- general business skills ?
- inter-personal skills ?

2i. How long does it take the expert(s) to solve the problem ?

2j. How well does the expert(s) chosen understand the problem domain ?

2k. What systems or resources does the expert require to solve the problem ?

2l. To what level of success does the expert solve the problem ?

3. User Requirements

3a. Can a user(s) be identified ?

3b. Would potential users welcome the system ?

3c. Would potential users be willing to participate in the design process ?

3d. Will the results of a using the system be politically sensitive ?

3e. Will the proposal require user training and/or relocation ?

3f. Are there different levels of competence of user ?

3g. Will the proposal change current user-expert/management relationships ?

3h What are the likely benefits to the user in implementing the proposal ?

3i. How would the user make use of the system ?

3j. What are the interfacing requirements ?

4. Development Prospects

4a Do resource requirements seem prohibitive ?

4b. Will the maintenance of the system prove difficult ?

4c. How quickly is the system likely to grow ?

Feasibility Check-list Continued...

4d. How might the system be tested and validated (e.g. are there sufficient test cases available) ?

4e. Is the proposed system project on another's critical path ?

4f. Are there strong supporters of the project up to senior management level

4g. Is an expert systems approach likely to be accepted ?

4h. Will the knowledge-based concepts prove too complex to implement?

4i. Does the proposal satisfy 'first project' criteria ?

4j. Can the proposal be justified ? Which of the following provides an acceptable measure of return on investment :-

- the average cost of decision-making to the company is reduced
- the value of a decision-making mechanism is improved
- the expert is relieved of routine tasks
- there will be an increase in expert productivity
- there will be an improvement in expert capability
- expertise will be preserved
- expertise will be efficiently disseminated
- the proposal is of educational value
- the proposal offers important technology transfer benefits
- other reasons (tangible/ intangible).

An Account and Justification of the Questions Used in the Feasibility Check-list

1a: Common sense reasoning is a highly complex and subjective form of knowledge which is difficult to narrow down and verify. Waterman et al(1986) stress that problem domains which have a high proportion of 'common sense' reasoning are inappropriate for expert systems.

1b: Unless the problem is of great importance to the company, it is difficult to justify occasional and infrequent and occasional use of a system.

1c: A problem domain will have primary and secondary sources of information. primary sources are elicited from the experts directly mainly through interviewing. Secondary sources include documentation, records, standards and codes of practice for instance and are of use in verifying primary sources. A problem domain which has substantial secondary sources of information is likely to be more easily and accurately represented than if it relied solely upon primary sources.

1d: The problem domain will be made up of varying proportions of deep and shallow reasoning: the former may require more complex representation techniques such as fuzzy logic, frames, semantic networks and attribute values; whilst for the latter more simple, rule-based techniques, are germane.

1e: To be feasible, it is normally required that the problem domain is narrow, well defined and well bounded. This prevents the situation where there is so much information that it becomes time-consuming and difficult to meaningfully represent, validate and test.

1f: The value of the problem domain to the organisation may be defined in business and strategic terms using value-chain analysis and critical success factors as used in Chapter 6.

1g: The value of the problem domain to users and experts is distinct and may be expressed in formal terms, essential to the functioning of users' tasks for example, but also in human and socio-technical terms- it is personally stimulating or improves personal effectiveness for instance.

1h: In evaluating the demand for a solution to the problem, it is necessary to balance the explicit needs of the organisation with the individuals and groups- users, managers experts, departments etc- which will be affected by the solution. Furthermore, demand for a solution may be localised to a single group or individual for political reasons.

1i: It is likely that a problem domain will be composed of both conventional and knowledge-based components. Some problems however, may adopt either technique to varying degrees of success: their use being chosen on the basis of political, financial and organisational factors rather than purely technical.

1j: The use of heuristics will require the means of assessing and representing uncertainty. Techniques such as certainty factors, fuzzy and baynesian logic are possible; although their use is theoretically disputed.

1k: The amount of knowledge required to solve a problem will differ according to the size of the problem domain but also the generic characteristics of the problem. For instance, diagnostic problems require proportionately less knowledge than design problems.

1l: The problem domain must be stable since expert systems cannot learn. Thus, an expert system in a rapidly changing domain will require frequent maintenance and updating if it is to remain valid.

1m: It is necessary that the problem is understood and appreciated by all those that would be affected by the proposed system, especially if it is to be used extensively in the user community.

2a: An expert is a person with specialist knowledge of a particular domain gained through years of experience.

2b: When there is more than one expert, it is important to define the interrelationship between experts by mapping out the boundaries of expertise and their interfaces.

2c: The expert must be available to work on the project during all design, development, implementation and post-implementation phases.

2d: The expert should be committed, interested and actively co-operative in the project for it to be a success.

2e: In order to be able to represent expertise through a model or other intermediate representation and eventually encode this knowledge using an appropriate programming representation, it is essential that the expert is able to describe his or her decision-making logic explicitly. The expert may find it difficult to articulate deep-level reasoning for example because it has become a sub-conscious activity.

2f: Where there is more than one expert, conflicts may arise in arriving at an agreed solution strategy. It is imperative therefore that there is consensus about the problem and how it may be resolved.

2g: If the expert is not respected by the user community, then the subsequent expert system will not be used.

2h: The characteristics that make up expertise may be unattainable by using an expert system. For instance, the expert may exhibit physical dexterity , have unique personality attributes, and have certain inter-personal skills which cannot be replicated.

2i: An indication of complexity is the time taken for the expert to solve a problem in a given domain. It might also suggest the role of the expert system: for instance, where an expert takes no more than five minutes to solve a problem, then the expert system might be used as a 'expert substitute'. However, if the expert requires more than thirty minutes, then the role of an expert system is likely to be as an expert 'aid'.

2j: The expert may only understand part of the problem domain at an 'expert' level and show a lower level of competence in other areas. It is difficult to establish the boundaries of expertise for this reason.

2k: The expert may require to use company databases and other systems as a component of the decision-making process. The design of an expert system may require similar access to these systems through linked, embedded or fully integrated capabilities.

2l: The precision of expert problem-solving determines the format of the user interface and the type of user eligible to make use of the expert system. In critical company areas, such as automated process control for example, it is imperative that the expert system operates at a 100% success rate and is able to resolve all of the problems that are likely to arise. By contrast, an expert aid is required only to provide suggestions and advice and only 50% of problems may be solved using the system.

3a: Where a specific role is defined for the expert system, it is necessary to identify appropriate users according to competence, operating environment and personal preferences.

3b/3c: Having targeted appropriate users, it is essential to gain their support and interest through consultation and participation.

3d: Problem domains should be avoided which are likely to cause political problems such as inter-departmental conflicts, break-down in communications and so on.

3e: The expert system proposal may require extensive user training and/or relocation of the user in the company. These may add significant costs to the project and may prove unacceptable to the users.

3f: Where users of difference competence are to use the same system, sophisticated user interfaces are required which might including different levels of consultation, help and explanation facilities; user profiling; and a logging and security system.

3g: The use of an expert system by users may change job descriptions and re-define the relationship between users management and current experts. It is important to anticipate these possible changes and plan for their effective management.

3h: As part of the justification process, it is useful to evaluate the potential benefits(or otherwise) that an expert system will provide for users.

3i/3j: The design of the expert system may provide opportunities for mis-use or conversely restrict the user from learning or operating effectively. It is necessary to match the functional role of the system with the personal needs and working context of the users.

4a: According to the characteristics of the problem, the nature of expertise and user requirements and given the constraints defined by 'first project'; criteria, an outline specification of resource requirements is possible from which unacceptable candidate applications may be filtered out.

4b/4c: According to the size, stability and complexity of the knowledge-base, some measure of maintenance requirements may be given. It is also useful to estimate the growth rate of the expert system from which future operating costs may be established.

4d: It is important to decide how the expert system will be tested and validated before it is constructed. This will depend upon the proportion of primary and secondary information sources amongst other factors.

4e: However desirable the prospects for an expert system, the likelihood of development may be lessened because of priorities in other areas.

4f: As with all projects, a 'champion' is required to support the project at an organisational level and preferably a project manager to facilitate development. In addition, user representatives and experts should support the project.

4g/4h: An expert systems approach may be technically feasible, but not viable for political, social and organisational reasons. Furthermore, the concepts may prove too difficult to be accepted.

4j: The justification for developing an expert system falls into two main categories: reducing costs; or adding value to an activity.

Appendix VIII.c.

An Evaluation of Application Proposals

This part of Appendix 8 provides an evaluation of each of the candidate applications, concluding in a recommendation to continue investigation of development suitability or abandon the proposal.

Proposal A_1: An Expert System to Configure Spares & Repairs

a) Positive Attributes

- Simple to understand the benefits of such a system in the organisation and therefore it appears a good 'first project'.
- It uses a generic configuration model of expert systems so that the potential of the application proposal is visible and of company wide relevance.
- The cost rating of the proposal is average for a 'system' type expert system.
- There is good potential to follow-on from this proposal and develop a more substantial application.
- It is reasonably straightforward to represent information
- The problem domain is fairly stable (new classes of trains arise approximately every 4 years).
- Experts are identifiable, available and clearly interested in the proposal.
- It would be very difficult to apply conventional programming techniques because of the substantive element of 'know-how' programming.
- The users of the expert system would be the experts themselves, thus simplifying design issues.

b) Limitations

- Despite the suitability of the domain, the likely exposure of the system is limited to perhaps two or three users and therefore the diffusion and organisational exposure is low.
- There is a high commitment to the project by the expert with a lack of supporting information from secondary sources (documentation, company standards etc).
- The problem occurs too infrequently to justify development
- The boundaries of the problem are uncertain
- Because of the high proportion of heuristic-based knowledge, it is likely that information gathering will be difficult and costly
- A large amount of knowledge will be required to provide basic decision-making assistance
- It presently takes the expert a long time to solve the problem: the expert system would have to address some small routine aspect of the expert's role to be feasible.
- Strong support from the project from senior management is not apparent. Furthermore, although the experts(2) have expressed interest in the project, it is unlikely that they will be able to participate in the project to the extent necessary for knowledge acquisition and validation.

c) Summary

The main justification for this proposal is that it will lead to an increase in expert productivity and also preserve existing skills. However, it is unsuitable as a first application given the level of awareness and understanding of expert systems in the company. Furthermore, there are a number of uncertainties which require clarification before further investigations can take place. These centre on the size, scope and complexity of the problem domain and the lack of organisational commitment by experts and management in this function of the company.

Proposal A_2: An Expense Claims Adviser

a) Positive Attributes

- The proposal has a low level of expert commitment: much of the system's knowledge is available from secondary sources.
- The system is a low cost option
- The benefits of the system are visible to the 'layman' and easily understood.
- The problem occurs very frequently.
- Information is available and accessible
- Knowledge is amenable to straightforward rule-based representations.
- The problem domain is stable, well defined and bounded.
- Experts (3) are interested, articulate and co-operative.
- Problems take less than 15 minutes to solve.
- An expert system is appropriate because of the formal relationship between experts and users (not a political or informal process, nor does it require a 'personal' touch).
- The proposed expert system would not affect existing company operations or procedures and therefore the transition the new system would be smooth.

b) Possible Restrictions

- Actual use of the system limited to a few users (therefore a low organisational impact and learning experience)
- No potential for further development
- Problem may be viewed as being trivial and therefore lacks legitimacy
- Unlikely that the expert will be available throughout the duration of the project.
- Unclear where support for the system lies.

c) Summary

The justification for developing this application is that it can help to improve the consistency of decision-making in this domain and also relieve financial staff of this routine task. Although the application is highly desirable on the basis of technical feasibility, it was rejected as a first project because it's impact and exposure in the company was considered negligible.

Proposal A_3: A Maintenance Adviser for Flexible Manufacturing

a) Likely Benefits:-

- Its basic troubleshooting model makes it highly applicable to most company functions.
- The problem is easily understood by the layman
- There is significant potential for a follow-on project
- Data collection is formalised through the use of an incident report form.
- The occurs fairly frequently
- Conventional programming is inappropriate for this domain.
- The problem is seen as being important
- There is considerable interest and support by functional and senior management and potential users

b) Possible Limitations

- The current knowledge base is small and is dependent upon new information generated from the incident report form.
- The size of the knowledge base is unknown
- The problem at present is unstable
- The availability of the expert is uncertain.
- The task requires both cognitive and physical skills

c) Summary

The justification for this proposal is that the system would help preserve maintenance expertise, improve current ES capabilities and help disseminate expertise in the company. There was considerable interest and exposure in the proposal and support from senior management. Although there was some uncertainty over the size and scope of the problem and ease by which knowledge is attainable, the potentially significant benefits from the system and its company wide appeal, led to the decision that further investigation was necessary to animate this potential more fully.

Proposal A_4: An Aid for Jig and Fixture Design

a) Positive Attributes:-

- Low cost project.
- Clear benefits and demand for the system.
- Potential for enhancement and improvement.
- Problem frequently arises.
- strong support from functional and senior management.

b) Likely Problems

- Poor expert commitment
- Problem broad and difficult to bound
- High level of heuristic analysis- no secondary sources
- Domain unstable
- May take the expert up to a day to design a jig
- Expertise constitutes both physical and cognitive skills
- It is uncertain who would use the system
- Maintenance is likely to be very difficult
- There were higher priority projects to be carried out.

c) Summary

The pretext for developing this proposal is that of improving the productivity of the expert and of preserving established expertise. However these potential benefits are much outweighed by the possible problems described above and so the proposal was not pursued.

Proposal A_5: A Shipping Planner for Manufacturing Services

a) Positive Attributes: -

- Low cost
- Stable domain
- Benefits appreciable

- Experts available and co-operative
- Identifiable users

b) Limitations

- Little potential in a 'follow-on' project
- Uncertainty over who the expert is.
- No senior management support
- Other priorities in conventional IT development

c) Summary

No organisational support for the project coupled with conflicting interests suggest that this project was not viable.

Proposal A_6: A Wire and Connector Configurer

a) Positive Attributes

- Problem stable
- Substantial supporting documentation
- Problem well bounded
- Strong support for the proposal from functional managers, and prospective users
- experts willing and co-operative.

b) Limitations

- Multiple experts with possible conflict
- Large amounts of information required to make a basic decision
- Problem may take up to 2 hours to resolve
- Possible difficulties in maintaining the system
- Unsure of the expert's role (substitute, Adviser or both)
- Unsure of the Users' role

c) Summary

Despite the potential of improved expert productivity, the technical feasibility of the system was judged uncertain and the proposal was dismissed.

Proposal A_7: Computer Fault Trouble-shooter

a) Likely Benefits:-

- As a basic diagnostics model, the application and benefits are well understood
- The application has a high potential impact and diffusion among user departments
- Extensive supporting documentation
- Knowledge may be encoded using rule-based programming
- Low cost
- Stable domain
- Senior management support
- Strong interest and support from user departments
- Identifiable and co-operative experts
- Diagnostic tasks well defined

b) Possible Limitations

- The system may lead to substantial organisational changes
- Large amounts of expertise may be required to make diagnoses

c) Summary

The justification for the proposal is fourfold: to increase expert productivity; preserve expertise; release experts of routine troubleshooting; and provide a better service to user departments. These benefits are related and greatly outweigh possible restrictions. For these reasons, this proposal was considered for further analysis.

Proposal A_8: An Expert System for Vibrational Analysis

a) Positive Attributes

- Demonstrates an example of a 'linked' application through the on-line interpretation of real-time data.
- Strong basis for future enhancement and follow-on developments
- Basis of the expert system is already available as an application specific product.
- Data collection is straightforward
- The problem is considered important and there have been requests for such a system for some time in the company.
- The problem domain is stable for each class of machine
- Experts are identifiable, co-operative and available during the duration of the project.
- Interpretation of the data inputs takes the expert between 20 - 60 minutes
- The users of the system would be the experts themselves

b) Possible Restrictions

- The application is complex, esoteric and the diffusion rate will be low throughout the company.
- The application m ay take more than one year to develop (despite the use of application specific software).
- The application has an 'Operations' business significance to the company and therefore this increases project risks
- The integration of sensor information to the software may be complex.
- The costs are likely to be high.
- Senior management support is uncertain.
- It was difficult to estimate development times because of the use of application specific software.

c) Summary

This proposal offers many benefits but is unsuitable as a first application. Furthermore, because of the relatively higher costs, risks and complexity, senior management support was critical. Yet, at the time of assessment such support was not forthcoming. For this reason, the proposal was rejected although strongly recommended as a possible second project when the benefits of the technology had been proven in the company.

Proposal A_9: *A Capital Investment Appraisal Adviser*

a) Positive Attributes

- Senior and functional management support
- Experts identified (two functions-Finance and Work Services)
- Strong support from users (engineers)
- Moderate diffusion rate
- Possibilities for further development
- Data is straightforward and may be encoded using rule-based representations.
- The re-assessment of capital appraisal was a priority and the proposed system offered a stimulus for change
- The problem domain, once bounded, was highly stable.

b) Possible Limitations

- Knowledge acquisition may prove difficult
- Some scepticism by Finance.
- No consensus on the process of capital justifications
- Problem solving (i.e. making a justification) can be lengthy
- System may revoke existing organisational procedures

c) Summary

This proposal differed from the others because the motives for its developed were founded on the premise that by undertaking the development process itself problems about the knowledge base and problem domain described above would be resolved. Since, in principle, the proposal was technically feasible and there was vigorous support from senior management and engineering functions particularly, it was decided to continue investigation of this proposal on the basis that if the application proved suitable for development, then the application would yield many tangible and intangible benefits and would be an excellent 'first project'.

Proposal A_10: *A Welding Process Selector*

a) Positive Attributes:-

- Low cost alternative
- Follows basic selection model- relevant to all company areas
- Potential for further development very high
- Data collection and representation straightforward
- Problem stable and boundable
- Significant demand for the project
- Experts interested and co-operative.
- Users identifiable and willing to participate
- Support by senior management
- Reinforce existing company procedures
- Secondary information sources available

b) Possible Restrictions

- Expert commitment high, despite secondary sources
- Possible difficulties in maintaining the system
- Large amounts of information are required to make a decision
- Diffusion of the system in the company is likely to be low.

c) *Summary*

The process of investigating the potential for an expert system in this area satisfied senior and functional management that a major knowledge-based undertaking would be desirable, of enlarged scope and complexity to the current proposal. Since this exceeded 'first project' criteria, the proposal was deferred until the technology was proven in another company function.

Appendix IX.

The Evaluation of Candidate Expert Systems Projects

Contents:-

Appendix IXa.

A Detailed Account of the Flexible Manufacturing Cell (FMC) Maintenance Adviser

Candidate Application: 1 of 3.

1. A Description of Problem Settings and Viewpoint

Based upon the multiple settings and analysis of viewpoints defined by Linstone and described in Chapter 3., and augmented by reference to the IDEFo models in Appendix IVb., this section analyses the FMC maintenance problem more fully prior to specifying design requirements.

1.1. The Technical/Physical Setting

The Flexible Manufacturing Cell (or 'FMC'), is designed to machine frame castings for locomotive motors. An outline of the cell shown in **Figure I.**, and consists of two Pensotti machine tools, a vertical turning lathe (VTL), and a vertical machining centre (NDM) which is served by a robot. Both machine are computer numerically controlled using programmable logic controllers (Allen Bradley AB-8650), and have automatic tool and head changing capabilities. Material enters the cell via two setting stations with large castings being loaded onto pallets secured with modular fixtures. Transfer to the machines is performed via a rail-guided vehicle (RGV) which loads and unloads pallets and also places unmachined or part-machined castings in buffer storage awaiting further processing. Specific machine control is exercised through programmable logic controllers (Siemens) and the entire system is controlled, monitored and interfaced with existing company information systems using Soflex derived software operating on a DEC Microvax computer. This layout is described in detail in the IDEFo model of Appendix IV.

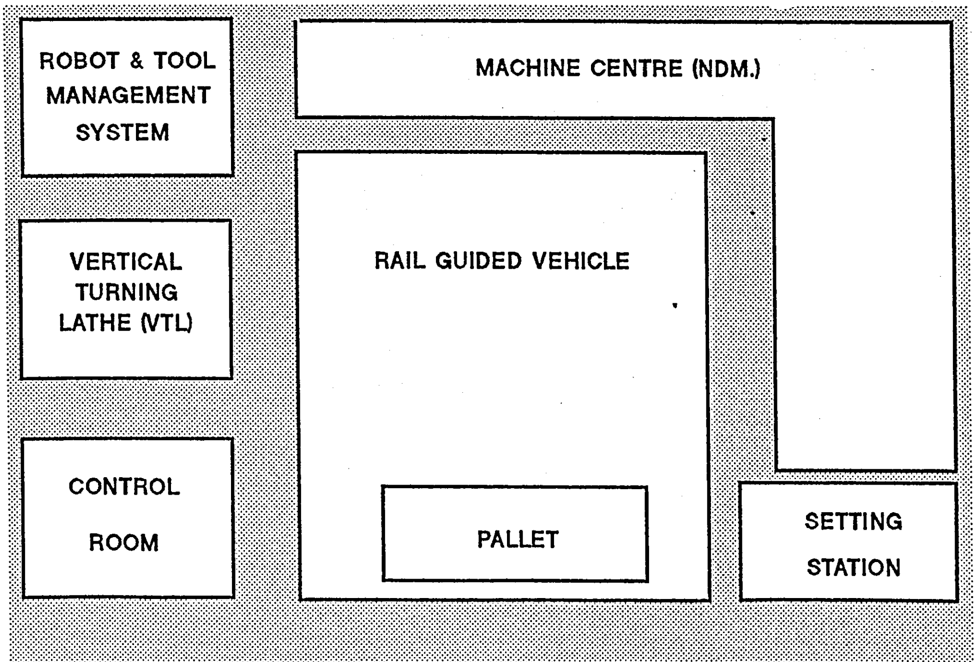


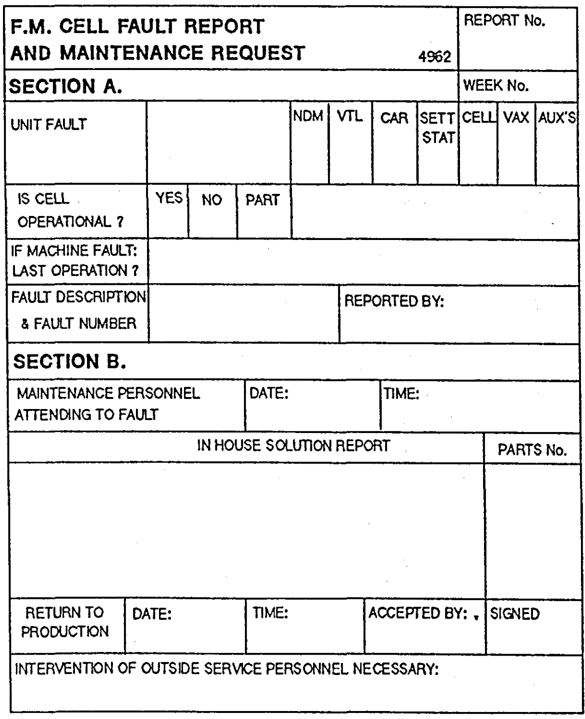
Figure I : An Outline of the Company's Flexible Manufacturing Cell

There are three basic characteristics which make maintenance of flexible manufacturing cells difficult: a large number of components which are interconnected in a highly complex fashion; a large number of ways in which the cell can fail; and that the time and place where a failure creates an observable symptom may be quite different from the time and place where the fault occurred. Compounding these problems are company specific factors which include: a knowledge bottle-neck caused by a lack of available diagnostic and debugging expertise residing in the company; and new maintenance engineers facing a steep learning curve.

1.2 The Socio-Technical Setting

Presently, maintenance engineers rely heavily upon outside sources of expertise . The purpose of developing an expert system was to develop an in-house maintenance capability and reduce this dependence. There were however, two organisational pre-requisites before such a system could be developed: to systemise the process of recording maintenance practice and procedures; and to centralise available resources and expertise on the FMC in each of the unit functions. The former was attained through a formal reporting procedure in which maintenance requests and subsequent maintenance actions were recorded on a report form, an exhibit of which is shown in **Figure II**. This provided a history of records which could be used to develop a knowledge base of cell faults and symptoms and subsequent maintenance actions to be undertaken. This proved to be of use irrespective of whether an expert system was developed or not.

Figure II: F.M.C. Fault Report



F.M. CELL FAULT REPORT
AND MAINTENANCE REQUEST 4962
REPORT No.

SECTION A.
WEEK No.

UNIT FAULT | NDM | VTL | CAR | SETT STAT | CELL | VAX | AUX'S
IS CELL OPERATIONAL ? | YES | NO | PART
IF MACHINE FAULT: LAST OPERATION ?
FAULT DESCRIPTION & FAULT NUMBER
REPORTED BY:

SECTION B.

MAINTENANCE PERSONNEL ATTENDING TO FAULT
DATE:
TIME:
IN HOUSE SOLUTION REPORT
PARTS No.
RETURN TO PRODUCTION
DATE:
TIME:
ACCEPTED BY: ,
SIGNED
INTERVENTION OF OUTSIDE SERVICE PERSONNEL NECESSARY:

The second organisational aim of centralising cell related expertise was not possible for the reason that each unit of the cell had very specific and complex forms of expertise which could not be combined readily as an organisational function and certainly not as an individual role, however sophisticated the expert system became. For this reason, the scope of the proposed expert system was reduced to include main hardware components of the cell only.

1.3 The Techno-Personal Setting.

The maintenance engineer is expected to resolve problems with the main hardware described above, software and auxiliary equipment such as transducers and power supply. Although the engineer is aided by cell monitoring systems which supply real-time progress and cell status information, the maintenance task is highly skilled.

There are a number of critical decisions the engineer is expected to make, the first of which is to decide whether the cell upon identification of the fault, can still function effectively or whether it is required to re-schedule work to other machines. This decision is based partly on whether it will be hazardous for an operation to fail from a sequence of dependent activities. Each cell unit may function fully, partially or not at all. For example, the vertical turning lathe may successfully complete a turning operation, but then the gripper robot may fail to remove the tool. If the fault is located in the machine cell, it is necessary to report the last operation so that the machine can reset from this datum so that its operations are synchronised with other units in the cell.

The cell controller, who overlooks the performance of the cell and programs job sequences, is expected to report faults displayed on the main control consul. The maintenance engineer thus receives a fault code and a location code (each unit of the cell is attributed with a number). Using this information, the engineer has to decide whether it is necessary to refer to outside specialist services (in which case the downtime and call out time are recorded from which the total maintenance costs are calculated) whether the fault can be resolved in-house.

1.4 The Individual Setting

There were two sources of expertise internal and external to the company. The primary source was from Italian service engineers who were charged with installing and testing the cell for the first few months of operations. During this period, interviews with the engineers were held to acquire as much knowledge of fault and maintenance procedures while they were available. The secondary source was from the suppliers documentation and from the learning experiences of the engineers and cell controllers, *inter alia* the maintenance report forms.

The users of the proposed expert system were to be maintenance engineers; and the system was to be used as an aid with the emphasis being upon the engineers becoming increasingly expert in locating faults and defining maintenance and remedy action. The role of the expert system would therefore change over time: at the beginning of cell operations the expert system would be used frequently; however as the engineers learnt more of the cell through experience, the system would be used only in exceptional circumstances to recall occasional faults or for training purposes for new staff. Maintenance of the expert system itself would be the responsibility of the engineers.

The current dilemma faced by engineers is two fold: one is that the engineer lacks the knowledge necessary for locating the causes of cell problems; and the other is that the engineer is not capable of fixing the problem once it has been located. The latter problem is highly amenable to an expert system solution because a knowledge base of repair procedures and other explicit forms of expertise can be gathered from the report form and other sources (although the problem still remains that the maintenance engineer will require training in order to understand and apply these procedures. On the other hand, diagnosing the causes of failure is a more intuitive and complex process because of the complexity of gathering knowledge from the expert and representing it in an expert system.

1.5. The Organisational Setting

The use of IDEFo is at a physical and information level as diagrams A4, A41, A5 and A6 show in Appendix IVB. The model is of use in that it provides an inventory of machines and information systems which are bounded or interact with the problem domain. The model shows that the systems monitoring capabilities of the FMC are advanced with tool status monitoring (A61), transportation system modelling (A62) which provide progress information , problem symptoms and operational status information (A5) to the cell controller. Although it would be feasible to define an on-line diagnostic system which would take real-time data inputs and respond to recommendations, this solution would be technically very complex because of the integration requirements between different computer systems as diagram A5 shows.

The recommendation for an off-line maintenance advisor was less ambitious and took account of the following constraints:-

a) many of the cell functions have automatic shut-down facilities; therefore the problem was often in re-setting and re-starting the system which was essentially a maintenance problem rather than a control problem.

b) knowledge of the system was evolving through practice and therefore a one-time installation of capital intensive, real-time process diagnostics was inappropriate,

c) Expertise was distributed because of the complexity of the system. The IDEFo model shows that the control software, machine control and cell monitoring staff were different and although each specialist was able to recognise symptoms of a problem in respective areas, the root problem may be located elsewhere. This made the maintenance role especially difficult since often conflicting information was received.

d) There was no formal recording of faults off-line which exacerbated the process of learning for maintenance staff.

1.6 The Political Setting

The initiative for the proposal came from maintenance engineers themselves and was championed and supported by the FMC project manager. The view by senior management was that although the project was necessary, there were more significant projects to undertake which were concerned with the full integration of the FMC with other company information systems.

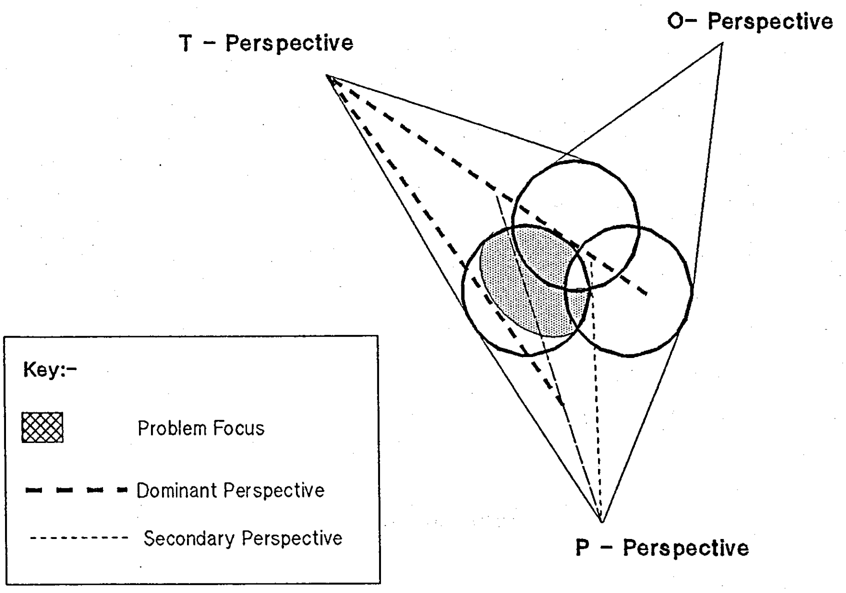
A further problem concerned the authorisation of maintenance work. In that the FMC was effectively divided into specialised hardware and software units, there were subsequently specialist engineers responsible for each of these units. The potential benefits in reducing downtime by equipping maintenance engineers with a general understanding of the FMC would thus be lost in the political process of seeking authorisation. This problem could only be resolved by the implicit approval and participation of such specialists in the project and this was not guarantied.

1.7. The Decision Focus

From the assessment above, it is evident that the domain is primarily technical and the problem addressed appears to be one of a techno-personal nature - how might maintenance engineers reduce the downtime of cell faults by improving their ability to locate faults. A dominant technical perspective is therefore appropriate,as **Figure**

III shows, in representing this problem because information processes and activities are , as the IDEFo model shows, formal and hard. Furthermore, a formal method of information gathering and retrieval is appropriate through the maintenance request form. The problems that arise are mainly technical also and are borne from the complexity of the system; although there are personal problems these are related to the technology. Thus the important settings from a T-perspective are physical, technical and techno-personal, as the shading in **Figure III** highlights.

Figure III : FMC Maintenance: Defining the Development Context.
(How is the Problem Being Viewed ?)



However, because the problem is at a technology level, it is not enough to assume that the maintenance engineer will behave rationally in determining faults and debugging the cell, and therefore a secondary personal perspective is necessary, as **Figure III** shows, to determine the personal circumstances by which the engineer is required to operate (how he is to relate personal needs to those of the technology ; how he is to interface with the expert system how he is to interact with the expert and other users and so on). A personal perspective also reveals that an improvement in both personal cognitive and physical skills are required in order for the proposed system to function effectively. Such educational and training needs should be considered as being separate to the design of the expert system. A P-perspective also highlights a·possible future conflict when the experience and understanding of the maintenance engineer at variance the collective knowledge in the expert system. At this stage, the engineer may continue to use the advice of the system through a lack of confidence, or conversely, make decisions independently of the system which are unwarranted. The discretion afforded to the engineer may be based more upon personality than technical competence.

There were further problems of a technical and personal nature and these could not be resolved in any other way than to experiment through the development of a small evaluation prototype.

2. The Construction of an Evaluation Prototype

Although the problem domain was complex, the process of encoding elicited information was less so and production rules were adequate in representing the acquired knowledge. However, since production rules are a rather awkward and lengthy way of representing knowledge when compared to frames and object hierarchies, this placed greater importance upon structuring the knowledge base for ease of maintenance.

A sample knowledge base of twenty main rules (although each rule was made up of a number of sub-rules and factual statements) was developed and is listed functionally in the Addendum below. The rules are backward chained and therefore begin with a hypothesis, there is a fault in the vertical turning lathe for instance, and then work backwards to identify symptoms, constraints and the eventual cause which satisfies the hypothesis. The prototype was developed on a low cost expert system shell which was currently under evaluation in the company, but nevertheless was sufficient as a vehicle for evaluating the proposal.

A lesson gained from constructing the prototype, and clearly visible from the addendum, was that it would be very difficult to define the limits to the system in terms of how large the domain was allowed to expand, but also in terms of verifying the information and removing bugs and conflicts. Even with a system of 20 rules the difficulties in cross-checking and validating both the knowledge base and then the system (knowledge base plus inference engine plus user interface etc) required that the shell being used was able to check consistency and had sophisticated edit facilities.

The purpose of the prototype however in this case, was less to verify the technical feasibility of programming the knowledge, than to provide some indication of the design requirements for the user interface which was the least well understood aspect of the proposal. Since the proposal would be off-line and not integrated or embedded with other software, large amounts of information would be required from the user in order for the system to offer basic advice. A high proportion of the information moreover, would be required to be presented in graphical format and certainly interactive video facilities would be the most effective means of communicating highly complicated maintenance procedures and advice. Furthermore, the system would have to allow the engineer to quickly volunteer information with all screens being menu driven. A further complexity was that different levels of consultation would be required since experience of the cell varied between engineers.

3. Summary

By adopting a dominant T-perspective, the maintenance advisor was considered as a bottom-up technical solution to a technical problem. However, by focusing upon technology related requirements, wider design issues (attained from a top-down O-perspective) which related to the functioning of the FMC in the organisation were overlooked. These included issues such as human resource management, systems integration and implementation planning; and the management of change. Closer attention to these issues from the onset, may not only have obviated the need for the expert system proposal, but may have lead to the to the re-design of the cell function itself.

At the level in the company at which the proposal was to function, the greatest uncertainty about the proposal was in precisely how the engineer would use the system and other techno-personal issues. The construction of the prototype revealed even at this small scale, major difficulties in the design of an appropriate user interface. These complexities made development within a year, an original stipulation as a first project, unattainable and the project was subsequently abandoned.

4. Addendum: Rule Listing for the Maintenance Advisor [*]

KEY: * - Rule Decomposition
$ - Refer to Rule Sub-Module for Test

In addition context sensitive help facilities were available using the F1 function key.

Main Rules:-

a) Faults in Vertical Turning Lathe

```
RULE 1:  IF FAULT IN VTL $
              AND CELL FAILS TO OPERATE*
              AND OPERATION WAS FACE MILLING*
              AND ERROR MESSAGE READ ' NOT ENTERING I.S.O. 50 O.K.
                                                  X.AXIS SPECIFIC HOLD. '
              THEN INCORRECT SPINDLE ORIENTATION*
                    AND CHECK STANDARDS *

RULE 2:  IF FAULT IN VTL $
              AND CELL REMAINS OPERATIONAL
              AND PART PROGRAM RUN TO 'FINISH'
              AND FAIL TO ZERO 'C' AXIS *
                    OR FAIL TO INPUT M40 *
                    OR ERROR MESSAGE 'TRANSDUCER ANOMALY.C'
              THEN TYPE 'CLAMPING'
                    AND PRESS KEY P2.

RULE 3:  IF FAULT IN VTL $
              AND CELL IS PART OPERATIONAL *
              THEN POSSIBLE DAMAGE OF FAN ON COOLANT PUMP MOTOR *
              OR SERVICE & REPAIR TESTS *
                    AND NOTIFY RECORDS *

RULE 4:  IF FAULT IN VTL $
              AND CELL FAILS TO OPERATE $
              AND FILES IN A/B CONTROL DELETED AFTER START-UP *
              THEN SUGGEST PRINT-OUT OF FAULTY FILE *
                    AND REPLACE DISTORTED DATA BACK INTO FILES *
                    AND COPY FILES INTO HARD DISK.*

RULE 5:  IF FAULT IN VTL $
              AND CELL FAILS TO OPERATE $
              AND ERROR MESSAGE 'M19 INTERMITTENT'
                                    OR ' INTERMITTENT FAULT IN M19'
              THEN ADJUST MICRO SWITCH ON TOP OF RAM *
```

---

[*] *Note: The rule format is written functionally rather than as it appears in the shell program for ease of understanding.*

RULE 6: IF FAULT IN VTL $
AND CELL FAILS TO OPERATE $
AND ERROR MESSAGE ' TABLE LUBRICATION: RT11:MF3,PR3'
THEN C AXIS FAILS TO MOVE *
AND SUGGEST RESET TRIPPED RT11.*

RULE 7: IF FAULT IN VTL $
AND CELL IS PART OPERATIONAL $
AND CROSS RAIL STOPPED BY LIMIT SWITCH *
OR CROSS RAIL FAILS TO ACHIEVE POSITION *
THEN SWITCH MACHINE OFF *
AND TAKE BRAKE OFF MOTOR CROSS RAIL *
AND HAND WIND CROSSRAIL ONTO SET POSITION *
AND SWITCH MACHINE ON *
AND INSERT PINS VIA HYDRAULIC CONTROL *
AND RESET SIEMENS PLC *
AND RE-SYNCHRONISE BOTH TOOL CHANGE MAGAZINES *
AND SWITCH ON MAINS *
AND SWITCH ON CNO *
AND RE-AERO THE MACHINE TOOLS *

RULE 8: IF FAULT IN VTL $
AND CELL PART IS OPERATIONAL $
AND E BUTTON WAS DEPRESSED *
AND NO POWER TO A/B CONTROL *
THEN CHECK 4 AMP FUSE V1 BLOWN *

RULE 9: IF FAULT IN VTL $
AND CELL IS PART OPERATIONAL $
AND GRIPPER DOES NOT REMOVE THE TOOL *
THEN CHECK PRESSURE SWITCH AR7 IS CLOSED *
AND CHECK PR7 PRESSURE IS CORRECT *

RULE 10: IF FAULT IS VTL $
AND CELL IS PART OPERATIONAL $
AND COOLANT OVERFLOW PROBLEM *
THEN CHECK SWARF BLOCKAGE *

b) Faults with NDM

RULE 1: IF FAULT IN NDM *
AND CELL OPERATIONAL *
AND ERROR MESSAGE 'COOLANT THERMAL CUTOUT 1:RT13' *
THEN CHECK COOLANT IS SWITCHED ON *
AND CHECK THERMAL CUT-OUT RESET TO 3 AMPS *

RULE 2: IF FAULT IN NDM $
AND CELL IS PART OPERATIONAL $
AND HYDROSTATIC LIGHT ON RED *
AND THERE IS NO HYDROSTATIC DELIVERY *
THEN RESET CT1 & CT2 *

RULE 3: IF FAULT IN NDM $
AND CELL REMAINS PART OPERATIONAL $
AND LAST OPERATION WAS PALLET LOADING *
AND THE C-AXIS FAILS TO DISENGAGE PALLET *
THEN MACHINE TABLE REMAINS UNCLAMPED $

```
RULE 4:  IF FAULT IN NPM $
             AND CELL IS PART OPERATIONAL $
             AND AUXILIARY FAILS TO WORK *
             THEN POSSIBLE RELAY CN1 FAILURE *
                     AND SWITCH AB TO MANUAL *
                     AND SWITCH AB TO RESET LOGIC *

RULE 5:  IF FAULT IN NDM $
             AND CELL OPERATIONS UNKNOWN $
             AND HYDRAULIC OIL TANK EMPTY *
                     OR HYDROSTRATIC TANK FULL *
             THEN OIL LEAK FROM CLAMP ON Z AXIS *
             AND REPLACE WITH NEW O-RING $

RULE 6:  IF FAULT IN NPM $
             AND CELL IS PART OPERATIONAL $
             AND LAST OPERATION IS LOADING ISO50 TOOL TO NS500HD*
                     OR HYDRAULIC LEAK FROM HIRTH COUPLING RING *
             THEN COUPLING NOT MESHED *
                     AND REPLACE MISSING O-RING *

RULE 7:  IF FAULT IN NDM $
             AND CELL OPERATIONS UNKNOWN $
             AND ROBOT FAILED TO UNLOAD TOOL *
             THEN TYPE KEYBOARD RELEASE
                     AND PRESS FEEDHOLD (LIGHT ON)
                     AND PRESS FEEDHOLD (LIGHT OFF)
                     AND PRESS START
                     AND TYPE KEYBOARD BLOCK
```

c) Others

```
RULE 1:  IF FAULT IN CAR $
             AND CELL IS PART OPERATIONAL $
             AND ERROR CODE IS 81 *
                     OR ERROR IS 83
             AND ERROR MESSAGE IS 'WRONG TELEGRAM FROM ERROR
                         CONTROL' *
             THEN LSF KEY OUT OF CONTROL PANEL *
                     AND RESET LIGHT GUARDS *

RULE 2:  IF FAULT IN AIR LIFTING BLOCK $
             AND PRESETTING HEADS NOT POSSIBLE *
             THEN ADJUST SETTING SCREW ON CONTROL ARM *

RULE 3:  IF FAULT IN SETTING STATION  $
             AND CELL IS PART OPERATIONAL $
             AND DIGITAL READ-OUT OF X-AXIS NOT DISPLAYED *
             THEN CHECK PCB CORRECTIONS *
```

Appendix IXb.

A Detailed Account of the Computer Hardware Fault Troubleshooting Problem

Candidate Application 2 of 3.

1. Overview

This proposal differed from the other candidate applications in that for a given domain, there were a diverse number of design alternatives based upon different interpretations of the problem. The IDEFo model and MPC analysis are used to understand the problem settings and context more fully from which each of the alternatives may be evaluated. It would be inappropriate at to attempt to define the scope and detail of a development project in this domain until such analyses (supplemented by the construction of an evaluation prototype and the use of a development assessment check-list) have been carried out. A full account of the development project is provided in the Chapter Seven.

2. A Description of Problem Settings and Viewpoint

The basic computer hardware troubleshooting function is carried out by specialist members of the computing department as a service to user departments. Currently, these specialists spend a high proportion of their time in solving user problems rather than undertaking developmental and research work. The size, scope and organisational roles differ for each of the options according to the specific computer systems in question: however, each shares with the other a number of common features which are characteristic of the problem domain :-

a) There are different levels of problem solving which may take the expert less than a minute to solve or many hours.

b) There is a severe knowledge bottle-neck in the distribution of specialist skills in the company.

c) At a high level of troubleshooting, expertise is intuitive and the expert will adopt rules of thumb to identify fault symptoms and narrow down the possible faults to a small definable area. For more detailed problems, the expert will adopt more systematic troubleshooting procedures.

d) The expert must be effective in locating the fault in a short period of time and of recommending subsequent action.

e) There is no single specialist who is expert in all computer systems but most are able to use these systems competently.

f) The majority of user queries attended to by computer specialists take less than five minutes to solve on the telephone and are considered ‘trivial’ by the specialists.

g) There is a variety of types of users which make use of the computer systems in the company ranging the naive user to the semi-expert. In dealing with each type of user, the specialist intuitively adjusts the mode of instruction accordingly.

h) The turnover of specialist staff is low for Mainframe computer systems, but higher for Office Systems, Computer Aided Design, and Network/ Communication Systems.

The justification of an expert system approach arise from its ability to capture the intuitive logic of the specialist in addition to formally documented methods of troubleshooting obtainable from company codes of practice and vendors' technical manuals. According to its organisational role, the value of an expert system based trouble-shooter may be expressed in terms of its ability to:-

a) reduce company dependence for troubleshooting and debugging upon a small number of specialist staff,

b) relieve specialist staff of routine decision-making tasks

c) allow non-specialist computer staff to perform diagnosis of faults in a number of computer hardware areas

d) capture the skills of specialists for all time

e) provide a useful training aid

f) incorporate standard procedures and company codes of practice into the troubleshooting process.

2.1. The Individual Setting

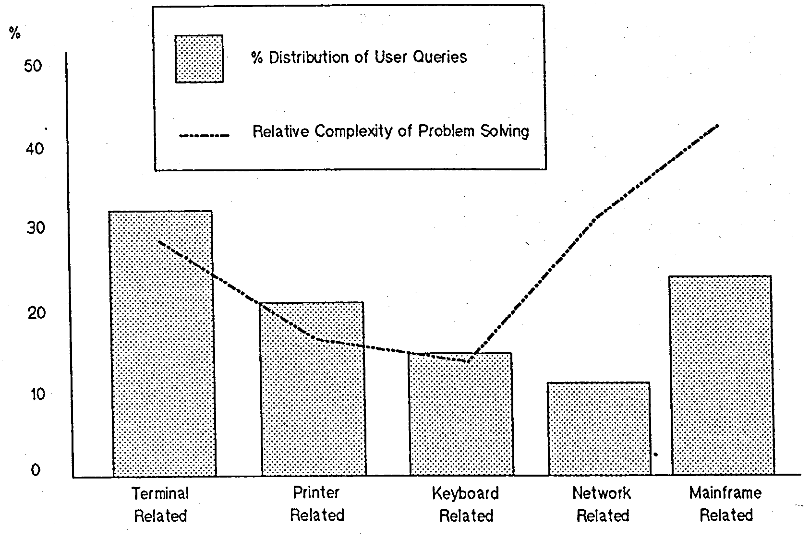
This setting revealed the nature of the service which was being requested by users and, consequently, the structure and frequency of the queries that the experts or specialists were expected to resolve. In order to gain some insight into these needs, telephone queries for users of the IBM Mainframe were logged over a two week period. The results of this analysis, shown in Figure I, highlight a number of problem features:-

i) 68 % of all reported queries (printer problems 20%, terminal problems 32% and keyboard problems 16%) could be solved by the expert quickly over the telephone, and moreover, were considered as an unnecessary inconvenience. In design terms, the problem-solving procedures for these problems was fairly straightforward and could be encoded as production rules in shell-based software. It was also significant that there was a large amount of secondary source information (from vendor and user manuals for instance) in these areas. The specialists felt that such problems could be delegated to an intermediary function, thus supporting the notion of a computer help-desk, and in areas, to selected users.

ii) The remaining set of problems, presented in Figure I, are network related(8%) or are concerned with electro-mechanical problems with the Mainframe itself. The complexity of problem solving rises dramatically for such problems for a number of reasons: firstly, expertise in these areas is characterised by deep causal reasoning; the specialist may take hours to identify the source of the fault and take subsequent action; and some of these faults may be intermittent. In terms of development, it is technical difficult and organisationally unwarranted to provide an ES based solution to such problems and efforts would be better directed at improving the training and availability of the specialists to attend to these problems more effectively.

Although these results refer to the IBM only, a similar analysis of VAX and Office Systems queries revealed a similar trend. In may be concluded from this that the greatest benefits to the company are in displacing a broad-band of top-level troubleshooting from the expert and making it more accessible to users. In terms of design, this requires that the expert system alternative is capable of undertaking breadth first searches of problems rather than the depth-first approach of deep-level reasoning.

Figure I : Analysis of User Queries on the IBM Mainframe Computer



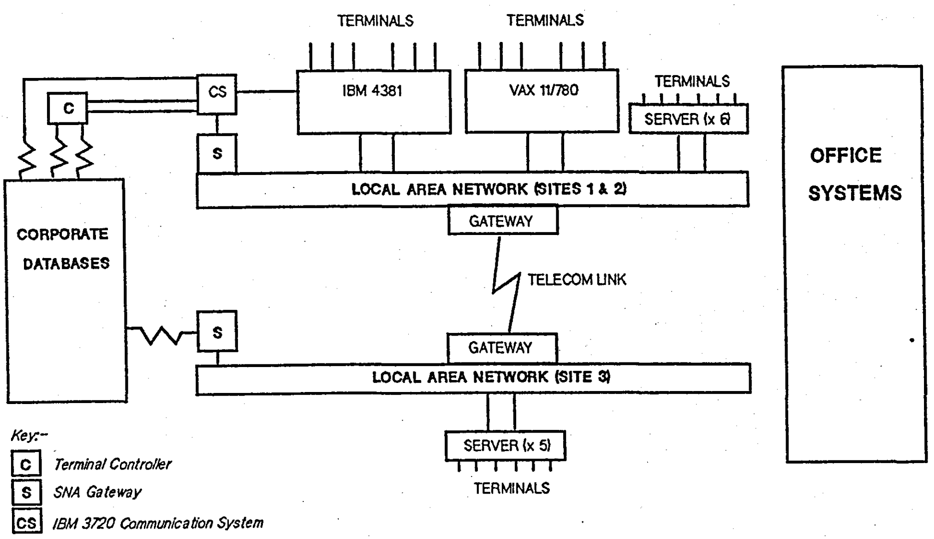
2.2. The Technical / Physical Setting

A detailed account of the company's computer hardware layout is provided by the IDEFo model, the relevant sections of which are exhibited in Appendix IVb. The IDEFo model was useful because it defined not only what systems were being used, but also how they were being used in terms of the types of users; the basic information and resource requirements necessary for the system to operate; the information outputs and destination from systems; and the constraints operating on systems which define how effectively they are being used. Moreover, by being hierarchical, the IDEFO model shows a decomposition of activities from a business level down to a computer systems level and beyond to an information level. This is useful in defining the scope and likely impacts of proposed additions or modifications to computer systems.

A more simplified representation of the company's systems architecture is presented in Figure II. This figure only shows those items of computer hardware which were considered appropriate and feasible for inclusion in the proposed trouble-shooter. This did not include workstations, interfaces and database systems for machine and automated manufacturing, such as the MICROVAX for the FMC; nor does it include computer aided design, drafting and modelling hardware and software. The main systems which were included in the proposal and their scope of use are:-

<u>a) IBM 4381:</u> This is the company's primary database management system. It is accessed directly by users physically connected to the mainframe using dedicated

Telex terminals. Alternatively, it is accessible to users with network terminals via the local area network. The potential for an ES in this domain is twofold: there is a set of shallow level problem-solving where an expert system based help-desk would be suitable and relates to the use of the system. However, for more complex problems which refer to specific problems with the mainframe hardware itself, an expert system could only be used as an expert or specialist aid; or in the absence of the expert, as a guide to 'semi-expert members' of the computing department.



FigureII: Diagrammatical Representation of Company Systems Architecture

(Only showing those systems which are to be included in the Troubleshooter)

b) IBM 3720: This device allows the IBM 4381 to communicate with the corporate databases and thereby allows selected IBM and network terminal users to have direct access and also modify corporate database records and files. The role of an expert system would be as an aid to the specialist only in recalling past faults. As with the mainframe, because of the complexity of the domain, it would be inappropriate to delegate troubleshooting to users and non-specialist members of the computing department.

c) Terminal Controllers (IBM 3274): This system provides dedicated terminals linked to corporate manufacturing, estimating, cost control and accounting databases. A highly specialist task which would be difficult to replicate through an expert system at other than a superficial level.

d) SNA Gateway: This communications device links the IBM 3270 to the Local Area Network and subsequent network terminals at all company sites. Advice and technical expertise in this area is obtained from the 'corporate information help-desk.

e) VAX 11/780: This engineering system is used for design analysis and modelling work and is accessible to user directly at Site 2., and via the network to Site 3 users. As with the IBM 4381, there is a broad-band set of problems which relate to how the system is being used which could viably be represented in an expert system and, organisationally, could be delegated to an intermediary function between the expert and the user, as with the help-desk concept.

f) Gateways: These devices link local area networks at sites two and three via a high speed telecom link. Expertise in this domain is highly esoteric, but despite this merits consideration for inclusion in an expert system because currently, there is only one expert.

g) Terminal Servers: These devices provide multiple terminal access onto the local area networks. Specialist advice about these devices is provided by the corporate information help-desk.

h) Terminals: Figure I has already shown that terminal related problems which includes use of the keyboard, monitor and connected printer problems, constitute the bulk of all user queries. This broad-band of relatively straightforward problems appears ideally suited to an expert system based, non-expert operated help-desk. Where problems were more complex in this domain, the ES help-desk could equally be of use in referring problems on to the expert with a report of symptoms and other technical information. In all cases, the sequencing of troubleshooting would be bottom-up from the terminal to the mainframe, i.e. from that hardware which is closest to the user to that which is furthest away in size and complexity.

i) Office Systems: This set of hardware includes Personal Computers(PCs) and related printers and peripheral equipment; dedicated word processors; and typewriters. In both technical and organisational terms, office systems are significantly different from previously mentioned hardware systems. Despite this, it shares with other systems the feature that most problems are relatively straightforward and resolved within three minutes on a telephone. On this basis, it would be viable to incorporate office systems expertise and procedures with other systems within the same ES help-desk.

2.3. The Socio-Technical Setting

A number of technical roles have been identified for an expert system. However, the most important setting in this problem situation is organisational - how can the computer department improve its computing service to users ? The priority to users is that a system is prevented from going 'down' through mechanical failure (hardware, power etc), systems failure(software, communications) or incorrect systems use (human error). There are three organisational responses to this problem at different levels in the company. The first is to ensure that modern equipment is used and maintained regularly: this is a corporate business decision and is constrained by company information technology strategy. The second response focuses upon improving the individual effectiveness of users and specialists through careful training, improvements in the level of computing literacy, improvements in access to help and supporting information and the monitoring of departmental needs. Where these two responses are pro-active in that they are planned changes in anticipation of the effects of the use of computer hardware, the third response, to improve the troubleshooting capabilities of the computer department when faults arise or the systems go down, is a reactive approach. In practice, elements of all three are necessary: plan for investment, manage the change process, and respond to problems effectively when they arise, with the potential contribution for expert systems being in the latter response. However, the extent of this contribution varies according to the degree of organisational commitment towards improving the company's troubleshooting capability. There are thus a number of socio-technical alternatives and these include:-

a) The diversion of all user hardware queries to a single source, a help-desk, which was manned by specialists in rotation.

b) The diversion of selected user hardware queries to a help-desk which was manned by a non-specialist, but computer literate member of the computer department, and assisted by a computer diagnostic system.

c) The organisational devolution of hardware troubleshooting to user departments

d) The technical devolution of hardware troubleshooting through the development of self-help computer diagnostic aids,

e) Improving the productivity by which the specialist/expert resolved user queries.

With the exception of c), which was discounted for structural and political reasons, all other alternatives were evaluated according to the feasibility of an expert system solution.

2.4. The Techno-Personal Setting

There are two, related, problems which are faced by computer hardware specialists in the company. Firstly, that they are often overloaded with work in which case the priority is in improving the productivity and response rate to user requests. Ideally, the specialist should monitor systems performance and key users in order to anticipate problems before they arise (clearly this is not possible with office systems). The closer to the point of failure that the expert is aware or notified of the problem, the easier it becomes to resolve. Conversely, the high causality between problems means that faults and errors generate cumulatively from a single problem such that many users may report different problems when infact they are all symptoms of the same fault.The second problem is that the expert may be unavailable to perform any troubleshooting in which case this function is delegated to another member of the computer department less suited to perform the role. The priority in this case is in ensuring that the seconded member is equipped with the correct advice to perform the troubleshooting and debugging function, even if the advice is as basic as to shut-down a system.

In both cases, it is conceivable that a ‘help-desk’ approach may be of use although the precise role for the system and organisational use cannot be accurately predicted without testing these concepts through the development of an evaluation prototype (*as Section 3 of this Appendix describes*).

2.5. The Organisational Setting

The service orientation of the computer department is shown from the IDEFo diagram A12321 in Appendix IV Part A. User department requests for major system enhancements are received by computing department management who, in accordance company policy and development expenditure approval prioritise development requests. More routine queries and requests are handled directly by operations staff, as A12321 shows, although this open access to users is often constrained by the planned work allocations made by senior computing management. The problem for specialist operations staff therefore is one of balancing development work and meeting senior management directives , with that of being available to respond rapidly to user queries and of monitoring systems performance.

2.6. The Political Setting

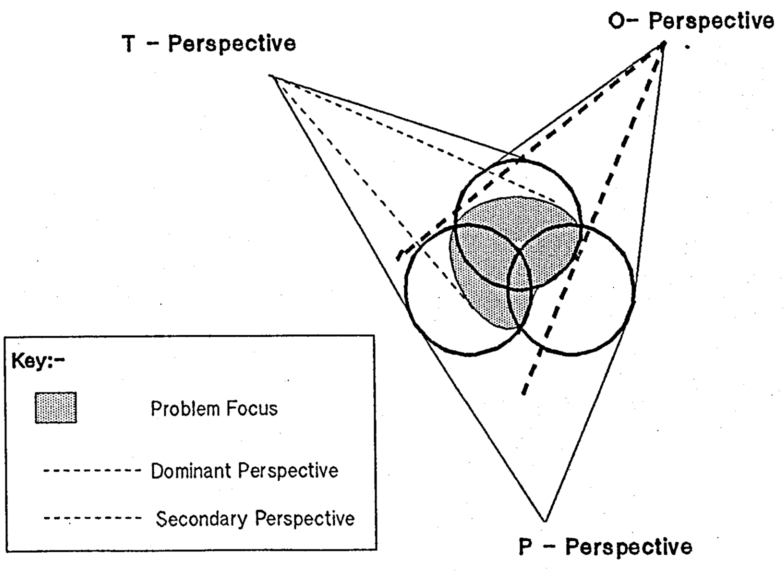
The project would be championed by the company’s technical director and managed by the company computing manager. There would be six specialists involved in the project (IBM-2; VAX-1; Office Systems-2; and Communications/Network-1) and a number of user representatives covering each of the main areas of application. The computer department was eager to improve its service in the company and was aware that a number of company functions were dissatisfied with the back-log of projects and quality of service. The proposal was thus seen as a useful means of improving its servicing image to user departments.

The development prospects in political terms, appeared strong since there was a consensus from experts, managers and users that a help-desk of some form would be necessary and expert systems would help to make this possible.

2.7. The Decision Focus

Although there is an technical proposal to develop an expert system, it is necessary to adopt a dominant organisational perspective, as Figure III highlights, so that the focus during development is upon improving the organisational effectiveness of the computer service in the company rather than specifically the performance of the technical system. Moreover, such a focus ensures that the system is orientated around the information required by user departments rather than that which is available form the expert. This emphasis also ensures that the correct viewpoint is maintained during an analysis of critical settings, which in this case are organisational, socio-technical and political, again as *Figure III* shows.

Figure III: Computer Hardware Fault Diagnosis: Defining the Development Context.



The problems which are likely to arise from this proposal are mainly socio-technical and centre on two main issues: how should the expert system be used ; and who should use it ? At this stage, the concept of an expert system based help-desk appeared organisationally and politically acceptable to management, experts and potential users. However, at the phase of specifying design requirements and the nature of the interface between the system, operator and users, a number of conflicts emerged which could only be resolved through experimentation and discussion with management using an evaluation (throw-away) prototype to communicate and test design ideas. This is discussed next.

3. Construction of an Evaluation Prototype

The function of the prototype was to provide a user-orientated troubleshooting service for computer terminal related problems, notably monitor, keyboard and mainframe printer problems. In technical terms, the system proposal was straightforward: the decision-making logic could be represented in a production rule format; knowledge acquisition was facilitated by the availability of secondary documented sources; and there was a high level of interest and commitment in the project. However, the greatest uncertainties in the design of the system were organisational and political rather than technical. The purpose of the prototype was thus to evaluate whether such a troubleshooting service was organisationally appropriate .

The scope of the prototype is shown in *Figure IV.* The types of problems that the system addressed were high level and shallow, as the addendum to this appendix shows, and intended to be solved quickly by the user directly.

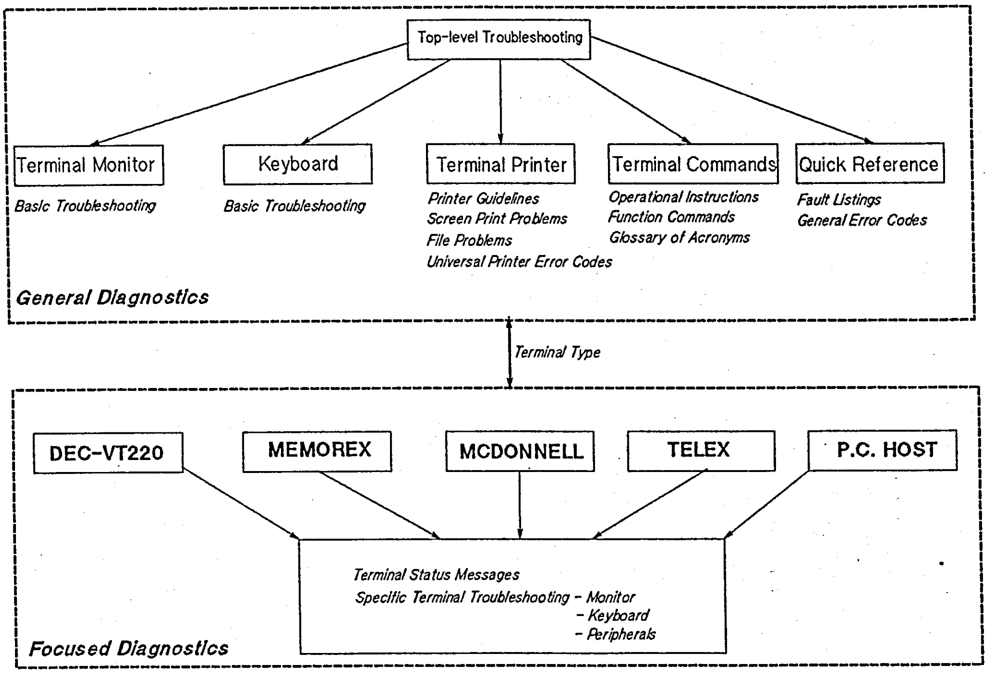
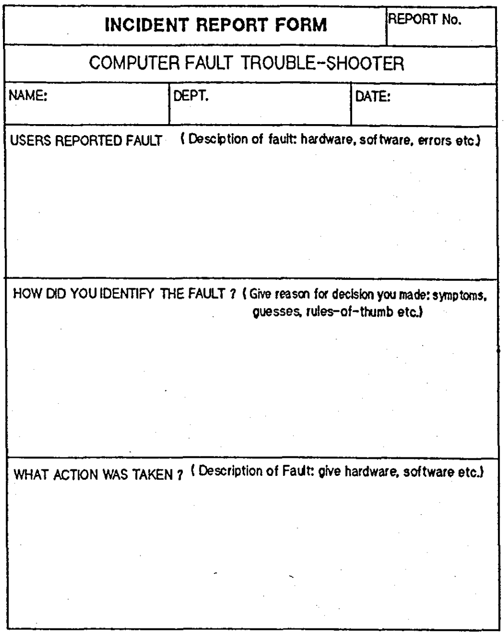


Figure IV: Organisation of Evaluation Prototype Knowledge Base

Knowledge acquisition had taken place concurrently with other activities and was based upon weekly half hour interviews with a single expert, and supplemented by a fault log filled in by selected users. The aim in using the fault log was to ensure that commonly recurring user problems were recorded from the viewpoint of the user as well as the expert. An exhibit of the fault log is shown in *Figure V.* It is divided into three sections. The first section is completed by the user and provides a description of the perceived fault including the type of terminal, keyboard, printer and other details. The second section describes the decision-making logic of the expert in identifying the root problem, symptoms and other factors which influence the course of problem solving, including the type of user (whether a naive computer user or semi-expert). Having identified the cause, the third section of the fault log describes the subsequent action taken by the expert or user in order to resolve the problem. A solution may be communicated to the user by the expert using a telephone or may require specialist repair by the expert or outside contractors. The necessity of a troubleshooting aid is that the user is quickly told whether the problem is easily solved or not; this prevents 'experimentation' by the user which may exacerbate the problem and make it difficult for the expert to identify the original fault.

Figure V: Exhibit of the Incident Report Form



INCIDENT REPORT FORM		REPORT No.
COMPUTER FAULT TROUBLE-SHOOTER		
NAME:	DEPT.	DATE:

USERS REPORTED FAULT (Desciption of fault: hardware, software, errors etc.)

HOW DID YOU IDENTIFY THE FAULT ? (Give reason for decision you made: symptoms, guesses, rules-of-thumb etc.)

WHAT ACTION WAS TAKEN ? (Description of Fault: give hardware, software etc.)

Knowledge and information acquired from interviews and the fault log were transcribed into a graphical format similar to that shown in the addendum. This made the decision-logic more visible and allowed meta-rules, or high level rules, and sub-rules to be identified and defined. From this, a rule-based shell currently under evaluation in the company (the full details of this product and the selection process are described in Chapter 7) was used to encode these rules from which a knowledge base of 100 rules evolved. The testing of this knowledge-base and the design of the user interface was achieved through a two stage process of verification with the expert and consultation with selected users.

An effective means of testing the knowledge base was to use the system alongside the expert. This provided an opportunity to verify current knowledge and also shape the mode of consultation between the system and the operator of the system according to the nature of the communication between the user and the expert on the telephone. There were a number of problems in using a telephone as the communication device: foremost was the difficultly of knowing whether the user understood what was being asked of him or her; and conversely, whether the expert was able to interpret the information given by the user (the user invariably expressed problems in terms of symptoms rather than root causes) and thereby define a set of troubleshooting procedures for either the user to personally undertake or for the expert himself.

At this stage, the knowledge base was reasonably complete. However as a system, there were deficiencies in the design of interface and uncertainties over the mode of consultation that the user should adopt when referring to the system for advice. In response to these difficulties, five selected users were approached, ranging in competence from naive to semi-expert and split between the company's two manufacturing sites and asked to operate the system in their functional areas for a trial period. The feedback from this exercise was essential in restructuring the design of the interface, from re-phrasing questions and the order of execution of questions, to providing context sensitive help facilities and improving the transparency of the knowledge base and access to the logic behind the decision-making process. However, it also revealed a number of socio-technical problems inherent in the process of devolving troubleshooting to users. These included :-

a) That there is no single definition of ‘user’. Each has different experience and competence in using the company’s terminals for different reasons. The computer department felt that certain users of the system could not be represented.

b) It was considered that however simple the interface between the expert system and the user, a minimal level of competence in using computer terminals was required to make effective use of an expert system based trouble-shooter.

c) In order to reflect different levels of competence, different levels of consultation were necessary reflecting the present knowledge and understanding of users if they were to have direct access to the trouble-shooter.

d) There was a practical constraint that not all terminal users had access to a personal computer upon which the trouble-shooter would reside.

The prototype was not targeted well enough at any particular set of users with the effect that it failed to serve none adequately. For instance, competent users found the consultations simplistic and slow in arriving at the information they required; furthermore these users expected a greater self-help role from the system such that it would allow them to assume a detailed troubleshooting role. By contrast, naive users could only be expected to follow the most simple instructions, press the key PF2 on the keyboard and the fault should clear, after which further problems should be referred to the expert

4. Summary

Despite the aforementioned problems users, management and experts were encouraged by the prototype for the following reasons:-

i) In most cases, the user was given a response which the expert considered expedient and the user found coherent, even if this was an instruction to turn off the terminal and consult the expert. In the few live and test cases where the prototype failed to provide an answer at all, the user was instructed to consult the expert.

ii) Although the use of the prototype had been restricted to user whom were considered as ‘receptive’ to such a system by the computing department, and therefore the political and organisational factors associated with company wide use could not be defined completely, it was felt by management and specialists in all areas of computing that further development of the prototype would yield favourable results with minimal adverse effects.

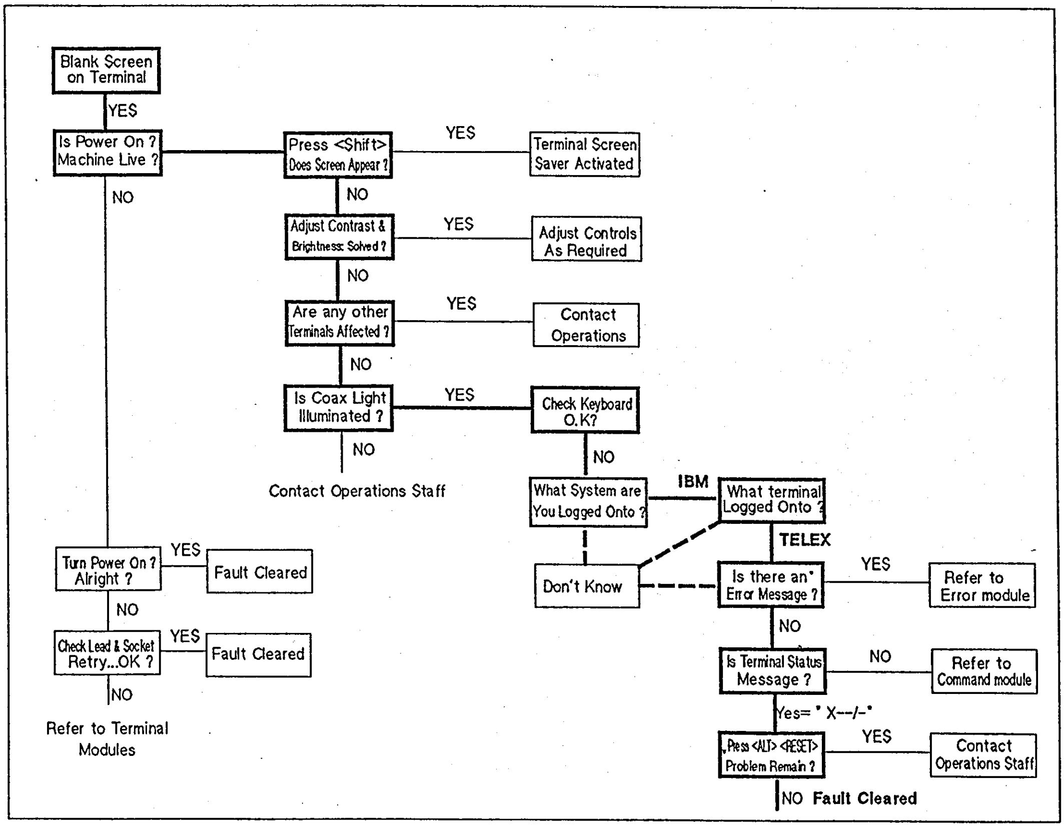
iii) The prototype was proving to be a highly effective communications and technology transfer vehicle in promoting an understanding of capabilities and potential in the company as part of the awareness campaign described in *Section 6.4.2* of Chapter 6.

iv) The prototype took one month to develop, although this did not include the duration over which faults were collected using the fault log shown in *Figure V*. However, this did confirm the original view that the knowledge would be straightforward to encode and suggests that a more substantial operational system covering all aspects of computer hardware troubleshooting could be developed in less than a year.

These positive results from the prototype however are not justification in themselves for the construction of a full-scale operational system but rather confirm the validity of the design principle and suggest a number of design prerequisites which are necessary for the system to function effectively in an organisational setting. This is supplemented by a more detailed specification of requirements achieved through the use of a development suitability check-list discussed in *Section 6.8.5.* and *Appendix 10.*

Addendum: Sample Consultations

Emboldened text and lines show the course of a particular live consultation. This example shows how the user is taken from a high level of troubleshooting where small corrections are suffice to a more detailed and complex level where expert intervention is necessary. Normal thickness lines and text indicate the decision options the user encountered during the above consultation. The average duration of a session with the prototype varies between two and ten minutes depending upon the complexity of the problem, the skill of the user in using the system, the knowledge and competence of the user, and the frequency with which the user makes use of help and explanation facilities during a consultation.



Appendix IXc.

A Detailed Analysis of the Capital Investment Appraisal Adviser Proposal

Candidate Application: 3 of 3.

1. Overview

This suggestion has had many political, organisational and cultural ramifications in the company, and has motivated the re-assessment of financial tools and practices currently in use. The process of enquiry into the technical feasibility of a possible ES in this area has identified shortfalls where a solution requires not simply an innovative arrangement of software, but a change in company wide approach and culture.

The proposal is an attempt to bridge different personal interests in the company representing two traditionally different cultures, engineering and finance. IDEFo provides a useful formal definition showing the structural arrangement between these groups (in terms of information flows, procedures and documentation), whilst MPC shows the interplay between personal and organisational perspectives

The study has two phases: the first is to look at present capital justification techniques used by the company and, based upon a history of records, provide guidelines to engineers on how to present information and use financial tools (namely payback and discounted cash flows) most effectively. The premise of this ES proposal is that this will standardise the presentation format of capital investment proposals, improve the efficiency of this function, and ease the evaluation process. The system could also aid in the processing of documentation involved in the purchase, transfer or disposal of plant and equipment. For instance, it would determine which forms are required, how they are to be filled in, whose approval is required and where the completed form should be sent. The system therefore provides an advisory and administrative service to engineers.

The second, and more ambitious phase to the proposal is to consider the effectiveness of capital investment justifications and appraisal. There are clear conflicts between corporate and company financial criteria for appraisal and those of justification by engineering and operations functions. These differences centre upon what engineering see as the 'unsuitability' of current company financial tools to measure the value that new capital plant will add to the company in the long term. This added value is both a quantitative measure of direct costs and benefits, *inter alia* using the traditional payback approach, but also more qualitative or 'intangible' (Primrose & Leonard: 1988) investigation of the costs of non-investment expressed in terms of downtime, maintenance costs, accuracy, non-quality, subcontracting and so on. Coupled with this , would be a global business approach to justification where a capital proposal would be evaluated based not only upon its individual merits, but also to the extent that it fits into the company's manufacturing, information technology and business strategies.

The role of the expert system in this second capacity would be as facilitator between three distinct organisational functions, engineering, finance and work services. The latter presently acts as am intermediary between engineering and finance and ensures that engineering proposals are valid and correct before they are submitted to finance.

2. A Description of Problem Settings and Viewpoint.

Based upon the multiple dimensions or settings, defined by Linstone and described in *Chapter 3*, and augmented by references to the IDEFo model, a useful definition of the problem is possible. This is made more complete by an appreciation of how the problem is being viewed by an analysis of dominant and secondary perspectives and their influence in shaping settings and decisions.

2.1. The Technical Setting

There are two facets of capital justification: cost savings and revenue generation. In both cases, there is some controversy over the suitability of financial tools used, especially in the justification of new technologies like Computer Integrated Manufacturing (CIM) which have pervading long-term implications for the company.

Software packages were reviewed which approached financial justification in a wider context (see for example IVAN by Organisation Development Limited). However, they were insensitive to the particular social and political settings and the computer interaction was considered insufficient for the users the system would address. An expert system approach was considered as a viable alternative because experiental and context sensitive information as well as formal procedures and techniques could be incorporated into the design of the system.

2.2. The Socio-Technical Setting

Current company practice is to make use of a three page 'request for expenditure' form. This provides a limited technical and financial format for justification, and provides no guidance for the engineer on how to complete the form to the greatest effect. Furthermore, it offers no advice on how to calculate and present financial measures of depreciation, discounting, rates of return and payback; nor does it accommodate non-quantitative factors. Engineers use these forms in an often ineffective and inconsistent manner which places greater demands upon the work service function and makes the process of evaluation by finance often lengthy and discretionary. The purpose of the Adviser system would be to systematically guide the engineer through a standard process of costing supplemented by more qualitative measures of benefits and for finance to take cognisance of these during a wider process of evaluation.

2.3. The Organisational Setting

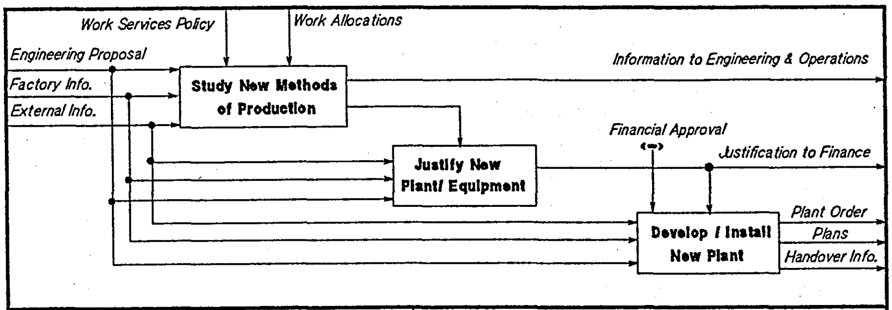
The IDEFo model shows the components of the problem and the two phases of the proposal. *Figure I* * shows the relationship between operation functions and work services. Requests are received and validated independently through reference to factory and external information. Contingent upon policy and work service plans and priorities and the viability of the proposal, a full justification is made by work services on the operation function's behalf and presented to finance. Again contingent upon finance approval, orders are placed upon plant or equipment, implementation plans are defined and the information is handed over to the operation function for installation.

At a higher organisational level, *Figure II* shows the inputs from work services to the finance department and more significantly, the policy and corporate constraints which condition the authorisation of expenditure. Also significant is that the same

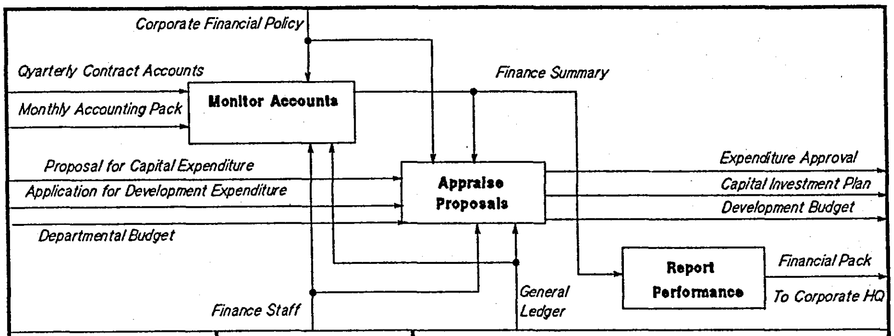
---

* *Note: Figures I and II are taken from diagrams A1211 and A112 respectively of the main volume of the company model which, because of its size, is not included in the thesis.*

organisational mechanisms which process departmental budgets and monitor and evaluate their performance are used for capital investment proposals, despite time horizons and purpose being very different.



Appendix 9c. | **Figure I** | **A Work Services Perspective on Capital Justification**



Appendix 9c. | **Figure II** | **A Work Services Perspective on Capital Justification**

2.4. The Techno-Personal Setting

Senior engineers and operations management lament over the failure to receive backing for capital proposals which they see as being essential to the company, because of the simplicity of payback and other financial tools and the inability to persuade finance of the company's long-term needs. This is exacerbated by the average engineer's basic understanding of financial measures.

Work service managers complain that the proposals they receive require extensive modification and verification and there is a high variation in the presentation and quality of justifications. They also express some discontent over the way in which proposals. once formally submitted to finance, are evaluated.

Finance managers are very much constrained by corporate directives which, as the above IDEFo model shows, defines a set rate of return on investment, and limits the allocation of funds, with some exceptions, on a quarterly and yearly basis thus limiting the scope for long-term investments. The effect of the Es proposal would be to question the legitimacy of inter and intra-organisational procedures and relationships.

2.5. The Individual Setting

The impetus behind the system proposal was from senior operations and engineering management who expressed the value of the system in the following terms:-

a) speeding up the process of submitting requests and approving capital expenditure,

b) standardising the above procedures,

c) allowing the engineer to express so called 'intangibles' in direct financial cost/saving terms,

d) enabling engineers with no experience of accounting procedures to submit quality proposals,

e) providing advice , reasoning and support in a consultative mode of interaction,

f) replacing discretion with rigourous and systematic evaluation

g) providing finance with an appreciation of strategic and qualitative factors as part of a wider evaluation.

The project would be championed by the senior manufacturing engineer who had influence in finance and accounting circles as well as in operations. This person would also be project manager.

There were two sets of distinct users. The first set comprised of engineers and operations management who would used the system as an aid to financial justification; whilst the second set of users were finance managers charged with appraising capital investment proposals. As before, the system would be used as an aid for the financial manager rather than as a replacement.

The body of expertise required to fulfil this dual role comes from three sources. Firstly, details of established financial tools and procedures comes from finance; second, knowledge of new methods and tools is provided by work services who also specify the organisational changes required by the systems implementation. Finally, senior engineers specify interface characteristics and the level of consultation for users.

2.6. The Political Action Setting

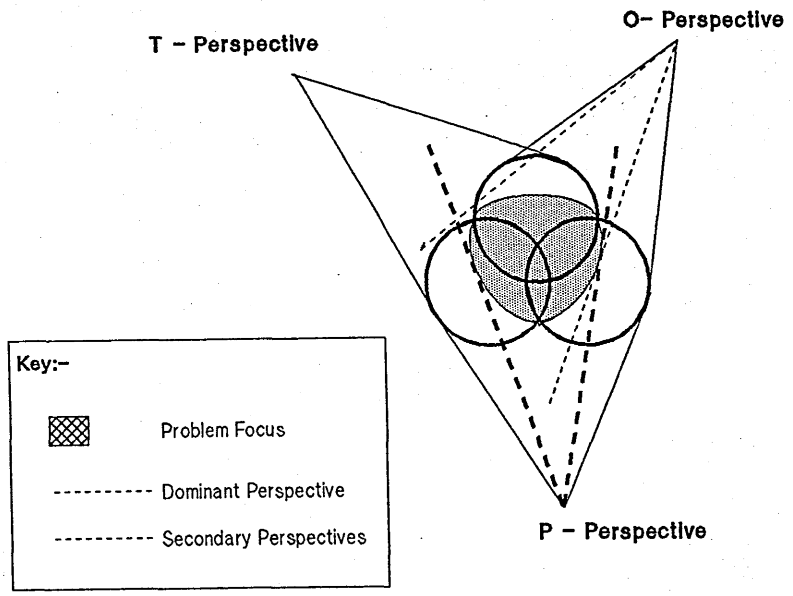
The effect of the ES proposal would be to make previously well defined and hierarchical forms of communication more fluid . The organisational role of finance as a controlling function would have to change to become more interactive and advisory. Although this may be beneficial to the company, such a transition is cataclysmic to finance with the effect that authority and the rights of discretion are usurped. It is likely that such changes are only possible through a evolutionary change of company culture and values which arise through a climate of increased awareness, understanding and co-operation. The ES proposal by contrast and in the current company climate may be viewed by finance as a political process and therefore likely to be of long-term damage to inter-departmental relations. For this reason, the proposal was not pursued, although this decision in itself has dismayed engineering functions and may change the structure of the problem to become even more political.

2.7. The Decision Focus

Where the previous two proposals were concerned with the use of ES technology to resolve technology related problems, this proposal intended to use ES technology to resolve an established organisational problem. The consequent effect was that in order to be successful, the system would be required to change company culture and the political relationship between key organisational individuals as well as company practice and procedures. The sensitivity of the domain coupled with the excessive expectations of the technology to effect this change made it politically unsuitable, especially in view that this was the company's first exposition of the technology.

Figure III shows that the most important and illuminating perspective in this proposal has not been technical but personal. A personal perspective revealed the political nature of organisational change and the motives for work service and senior operations management requesting this change. This perspectives highlights the personal and organisational expectations from using the technology: for engineering and operation functions it provided the opportunity to change procedures and make relationships with finance more flexible. For finance, there was a perceived threat by the system and the understanding that internal changes would be discordant with corporate directives.

Figure III : Capital Investment: Defining the Development Context.
(How is the Problem Being Viewed ?)



A secondary organisational perspective of the problem reveals the present social and cultural circumstances in which current tools are used and moreover, the cultural changes necessary in order to implement the proposal and yet retain organisational stability. It might also be used to suggest changes in financial practices which do not require ES technology but may achieve the same objectives in a more harmonious and evolutionary. manner.

3. Construction of an Evaluation Prototype

The above analysis revealed that it would be incautious to develop a prototype at either phase of development until a complete consensus on the role of the system was agreed upon between engineering and operations, work services and finance. The likelihood of this arising in the next year was remote and therefore the project proposal was annulled.

Appendix X

Applying Development Suitability Criteria During Project Specification

Contents:-

Appendix 10a

An Exhibit of the Development Suitability Check-list [*]

---

[*] There have been a number of revisions to this check-list: the exhibit shows the final version. Changes to the design of the check-list have been made on the basis of feedback from computer department personnel and the author's own experiences in applying the check-list to the computer hardware troubleshooting prototype.

Section One: Application Analysis

1.0: How critical is the location of the proposed application ?

1.1: What is the business impact of the proposed application ?

1.2: What are the objectives in developing the proposal?

1.3: What are the interrelationships between the proposal and its environment ?

1.4: Using IDEFo, which activities are affected by the proposal (causality, systems boundaries)?

1.5: What are the inputs, outputs, constraints and mechanisms to the proposal's boundaries ?

1.6: Is the proposal in-line with corporate objectives ?

1.7: Is the proposal in-line with the company's Information Technology Strategy ?

1.8: Can performance standards be defined at this stage ?

1.9: Who should be involved in the project ?

1.10: Who will be affected by the project ?

1.11: Have other technical alternatives to the problem been considered ?

1.12: Have other non-technical solutions been studied ? Consider the following:-

- greater commitment to education and training
- rationalisation
- restructuring /re-organisation
- recruitment (make more experts available for example ?)
- the use of new management techniques / change in company practices

1.13: How has this problem been tackled in the past or in other organisations ?

Section Two Problem Attributes & System Requirements

Users:

2.0: Have users been defined ?

2.1: What minimum input is the user expected to contribute for the application to work?

2.3: Are users required to be computer literate ?

2.4: What are the likely user training requirements ?

2.5: How are users to respond to the system (e.g.to be instructed, advised, tutored, etc.) ?

2.6: How should data/information be presented to the users (e.g. graphs, text on-line, off-line) ?

2.7: What interfacing facilities will the user require ?

2.8: Will users accept an iterative process such as ES development ?

Experts & Expertise:

2.9: Are the expert's management supportive of the project ?

2.10: Are relationships good between the expert and developer/user/manager ?

2.11: Can expertise be defined explicitly

2.12: Can expertise be specified so that it is accessible to problem solving ?

2.13: What form does the problem-solving expertise take (structuring- broad, fuzzy etc.) ?

2.14: Is it possible to define relationships rules and procedures (representation) ?

2.15: Is there a means of specifying the consequences of a given decision ?

2.16: How accessible is the expertise ?

2.17: What is the function of expertise (e.g to control sub-tasks, control search, reduce the uncertainty of information, process preferences/priorities, resolve multiple objectives etc)?

2.18: What is the degree of dependence on expertise ?

2.19: What constraints operate on the use of expertise ?

Data/Information:

2.20: What is the source and structure of the input data ?

2.21: What are the functional information characteristics:-

- source : internal-to-external
- scope : well-defined-to-very wide
- aggregation : detailed-to-aggregated
- time horizon : historical-to-future
- currency : old-to-new
- accuracy : high-to-low
- frequency : high-to-low
- structure : unstructured-to-structured
- availability : given-to-inferred.
- reliability : high-to-low.

2.22: How can data be acquired ?

2.23: What classes of questions need to be asked in order to obtain sufficient data ?

2.24: What information is required to justify a decision ?

2.25: What information is required to take action on the basis of a decision ?

2.26: How should data/information be presented at each of the above stages ?

Section Three : Systems Architecture & Design

Knowledge-base & Inferencing Needs

3.0: How will knowledge be validated ?

3.1: How is knowledge to be formalised in a knowledge base ?

3.2: What is the expected size and structure of the knowledge-base ?

3.3: What is the expected rate of change of the knowledge base ?

3.4: How long will knowledge acquisition, representation and validation take ?

3.5: What methods of inference are likely to be used ?

3.6: What is the ‘goodness-of-fit’ between the problem and the knowledge-representation ?

Hardware Features

3.7: What is the current technology layout

3.8: What are the expected infrastructural requirements ? (stand-alone, integrated etc.)

3.9: What are the hardware requirements (e.g. Mainframe, work-station, P.C.)

3.10: What is the cost restraint on hardware ?

3.11: What are the hardware delivery constraints (e.g. portability, dust-proof etc)

3.12: What are the hardware memory size and clock speed requirements ?

3.13: Does the hardware system require a mouse, a colour/graphics monitor and peripherals?

Software Features

3.14: What are the current in-house software capabilities in the company ?

3.15: Are software linkages anticipated (e.g. embedded, linked, hybrid, real-time etc)?

3.16: What are the software tool options (e.g. conventional software, A.I. languages, A.I. Toolkits, Shells, Application Specific Software)?

3.17: What is the cost restraint on software ?

3.18: Which representation techniques should the software have ?

3.19: Which control strategies is required ?

3.20: Which uncertainty technique does the software require ?

3.21: Does the software provide the appropriate source code ?

3.22: Does the software offer a built-in programming language ?

3.23: Is the software expected to provide security ?

3.24: How easily does the software allow updating and maintenance ?

3.25: Is a run-time version of the software required ?

3.26: Is the software required to be modular ?

3.27: What help and explanation facilities are necessary?

3.28: Is the software required to handle real-time and/or on-line information ?

3.29: What service and support is offered by the software vendor ?

3.30: Who is required to use the software and therefore how easy is it to learn and operate?

3.31: Is the software confined to a particular specification of hardware (e.g. a laptop)?

3.32: Does the software require file-handling capabilities ?

Section Four: Soft Issues in the Development Process

4.0: Does the expert feel threatened by the proposal?

4.1: What is the potential skills impact of the proposal (enhance, down-grade, shift)?

4.2: Will there be any adverse political effects in using or developing the proposal ?

4.3: Is the proposal likely to re-define organisational/group/individual relationships ?

4.4: Is there consensus on the role of the expert system ?

4.5: Who is legally/organisationally responsible for the system's decision-making ?

Section Five: Project Planning & Development Management Issues

5.0: How can the user participate in design and implementation ?

5.1: How are users' interests to be managed ?

5.2: Who is charge of knowledge engineering ?

5.3: Who is charged with systems programming and development ?

5.4: Who is the project manager ? Who is the project champion ?

5.5: How are overhead and direct costs to be calculated (e.g. will the expert's time-off costs be booked to contract ?)?

5.6: What is the time-scale of the project. Have deadlines been set ?

5.7: Does the time estimate allow for testing, verification, debugging, implementation & validation?

5.8: What is the estimated size of the system ?

5.9: Who is to write the user and technical manuals and any other possible documentation ?

5.10: How is the system to be validated ?

5.11: Does the development team include: sponsoring and project managers; user -representative; user management representative; experts; and developers ?

5.12: Can project dead-lines be set ?

5.13: Have training and maintenance programmes been devised ?

Technology Transfer

5.14: How well is the technology understood ?

5.15: What are limitations of the proposal ?

5.16: What would happen if the project became widespread or was greatly scaled up in the Co. ?

5.17: Are management expectations of the project too high ?

5.18: Is it important that the project serves to diffuse knowledge of this technology in the Co. ?

Section Six: Anticipated Problems & Future Prospects

6.0: Are there any financial constraints ?

6.1: Is the system difficult to justify ?

6.2: Will there be a lack of support or development expertise over the lifecycle ?

6.3: Are there likely to be any supplier problems (e.g. failure to meet dead-lines; maintenance; support; over-selling of capabilities; documentation problems) ?

6.4: Can you envisage any technology problems? For example:-

- failure to meet performance requirements ?
- wrong system for the problem ?
- high risk project ?
- unreliable hardware or difficult to use ?
- unreliable software or difficult to use ?
- site/locational problems ?
- elicitation/representation problems ?

6.5: Can you envisage any human and organisational problems? For example:-

- user acceptance ?
- industrial relations difficulties ?
- possible de-skilling or downgrading of job functions ?
- over-dependence and/or loss of autonomy ?
- loss of responsibility ?
- lack of training (as a possible substitute for training) ?
- organisational restructuring/re-organisation ?

6.6: Are there likely to be any possible operational problems. For example: -

- will it be possible to enhance the system ?
- could the system be extended to more users ?
- is the structure of the problem likely to change ?
- how maintainable is the system ?
- will the type of users change ?
- how will the system accommodate user learning ?

Section Seven: An Evaluation of Costs

Tangible Costing Information:

a) Software:

7.0: What is the cost of the basic software ?

7.1: Who will use the software and how much training is required ?

7.2: Are additional software vendor services and documentation required ?

7.3: Are source or built-in programming languages extra cost ?

b) Hardware:

7.4: Can existing hardware facilities be used ?

7.5: What new hardware is required (e.g. additional terminals, printers, disk-drives)

7.6: What are the integration requirements and costs ?

7.7: What are the maintenance costs on new and existing hardware ?

7.8: What are the instalment costs ?

c) Systems Development

7.9: How long is the project estimated to take (Total man-months) ?

7.10: What are the costs of project planning and project organisation (management support; selection of personnel etc.) ? Which of these costs are allocated directly to the project and which costs are carried as company overheads ?

7.11: What are the costs of knowledge elicitation:-

- what is the cost of expertise to the company ?
- how many experts are involved in the project ?
- what is the expert commitment to the project (£/man-hour)
- What are the costs and duration of information gathering ?

7.12: What are the costs of knowledge representation:-

- is a knowledge acquisition tool required ?
- how long will it take the developer to represent and verify knowledge ?

7.13: What are the costs of knowledge-base development:-

- what is the estimated size of the knowledge-base (rules etc)
- How much programming effort is required ?

7.14: What are the costs of interface development:-

- what are the costs of user participation ?
- how is the interface tested and validated ?
- how much prototyping can be allowed (cost/duration limitations) ?

7.15: What are the system test and verification costs:-

- what is the beta-test duration ?
- what is the minimum accepted standard of decision-making?
- what is the test and validation effort ?

7.16: What are the documentation costs:-

- what documentation is required (user manual technical manual etc)
- who should produce the documentation ?

7.18: What are the Implementation costs:-

- relocation, displacement and disruption costs ?
- installation supplies and services ?
- manpower costs?

d) Systems Use

7.19: What are the costs of system support ?

7.20: How frequently will the system be used ?

7.21: How many users are there to be ?

7.22: How much training is required ?

7.23: What are the costs of maintenance ?

7.24: What is the realistic life of the project ?

7.25: What will happen to the system thereafter ?

7.26: What are the costs of planned enhancements ?

7.27: Are there any other possible future costs ?

Intangible Costs

7.28: What are the opportunity costs of developing an ES application ?

7.29: Are there costs to the user ?

7.30: Are there costs to the expert ?

7.31: Are there adverse political/organisational costs ?

7.32: What are the costs to the organisation should the project not succeed or not be used?

7.33: What is the cost of invalid or erroneous decision-making ?

7.34: Will the project impinge upon the effectiveness of other people or operations ?

7.35: What are the project risks and how proven is the technology ?

7.36: How is the project being financed ? Is there a budget limit ?

Section Eight: An Analysis of Potential Benefits

8.0: What are the direct financial benefits of the proposal:-

- cost saving through expert substitution (down-rating of labour costs)?
- cost reduction through fewer staff (rationalisation of labour)
- reduce the cost of expertise to the company by for example:-
    - improving decision-making productivity
    - reducing the average time of consultations
    - improving the availability of expertise
- reduce the cost of wastage of users' time in requiring expert assistance
- cost saving in preventing errors, faults etc.

8.1: What are the 'added-value' benefits of the proposal :-

- gain competitive advantage ?
- improve company image ?
- distribute expertise ?
- improve the speed/efficiency of decision-making ?
- improve the effectiveness of an individual/group/function ?
- to create new business/organisational opportunities ?
- to improve the quality of a service ?
- to improve communications
- to improve user/expert job satisfaction ?
- to preserve valuable expertise

8.2: What is the technical importance of the proposal:-

- necessary as part of company's computer systems architecture development ?
- to enhance current company understanding of the technology's potential ?
- part of the company's strategic development plan
- to facilitate the technology transfer of future knowledge-based projects ?

Appendix X: Part B.

An Account of the Development Suitability Check-list

I. Background

A purpose of the development suitability check-list was to consolidate all previous and separate analyses- business, technical, social and human within a single assessment in order to provide a comprehensive pre-development specification of requirements. The check-list was to be used by members of the computer department in liaison with targeted users and experts; however in the first instance, the check-list was used exclusively by the author.

Although individual questions and criteria have been borrowed from other studies, the check-list, and more importantly the context of its use is unique for a number of reasons:-

i) A feature of the check-list was that it would be multi-dimensional and therefore draw attention to non-technical and process-orientated issues.

ii) The check-list distinguishes between technical feasibility and development suitability as two separate phases of pre-development assessment.

iii) The check-list promotes the analysis of needs from more than one perspective. The check-list was based upon analysis of each of the three perspectives (personal, organisational and technical) and settings defined in earlier chapters.

iv) The check-list may be used as a basis for planning and requirements specification. It is not considered as a complete means of specification however and its use is supplemented by prototype development and other analyses described in Chapter6.

v) The check-list is not to be used directly by user departments as a ‘self-help’ guide, but is designed to be used by the computer department as part of a wider process of inductions and evaluation.

II Design

The questionnaire is divided into seven sections as Appendix 10a has shown. The structure of the check-list was motivated by the settings defined by Linstone(81) and the premise that project requirements would be better understood by analyzing each of these settings from more than one perspective.

The questions for the check-list were obtained from numerous studies, each in the main adopting a singular perspective as *Figure I* shows. Clearly the nature of investigation which is stimulated by each question differs according to the perspective or viewpoint adopted. Within a T-perspective for example, questions focus upon hardware and software selection, systems integration, systems specification, technical design and the use of formal methods such as cost/benefit analysis for justification. Current check-list approaches in expert systems development adopt this viewpoint. An organisational perspective looks externally from the organisation of the project to consider business and strategic issues; and internally at the project management and the required processes of development. The individual perspective by contrast focuses at a personal level upon user and experts needs and in particular from a 'techno-personal level, at human interface and interaction requirements.

Although each of the question sources shown in *Figure I* address particular issues which are relevant at different stages of suitability assessment, the emphasis in using the check-list should be upon combining settings and viewpoints, where possible, so that a greater understanding of development needs and development context is gained. Where there is conflict between different settings and values held in each of the viewpoints (for instance which is the more important human needs or business needs?), there is no formula or quantitative technique which provides a basis for selection. Rather that each setting contributes tangible and intangible factors which must be weighed up and considered through discussion and interaction with all personnel affected by the system. In this sense, it is both a formal and informal assessment: indeed, Linstone observes that by transforming O- and P- perspectives into formal theories, they are 'unconsciously transformed into T perspectives', thus losing the essential basis of their value.

Figure I: *The source and Dominant Viewpoint of Criteria used in the Development Check-list*

P-Perspective	**T-Perspective**	**O-Perspective**
User-centred design Criteria (Candy & Lunn:88)	ES Tool selection Criteria (van Koppen:88)	Socio-political Criteria (Markus:84)
Human Factors Design Criteria (Clegg:88)	ES Design Criteria (Hayward: 86)	Socio-technical Criteria (Mumford:89)
Human-Centred Design Criteria (Brodner:88)	Task Characteristics (Prerau:89)	Business Factors (Rockart:84)
Technology Assessment Criteria (Linstone:84)	Development Suitability Criteria (Waterman:86)	Technology Assessment Criteria (Linstone:84)
	Technology Assessment Criteria (Linstone:84)	Development 'Process' Factors(Hirschheim:85)
	Requirements Planning (Liebowitz:89)	Planning/Implement'n Factors (Bramer:1988)

III Check-List Use

The check-list is intended as an aid for computer department personnel. It does not claim to cover all issues important in the assessment suitability however; nor odes it attempt to prioritise criteria as the weighting of each will clearly vary according to the viewpoints of the project's participants and the characteristics of the application.

Experience has shown that computer department personnel may experience some difficulties in using the check-list for the following reasons:-

i) Computer personnel may fail to see the significance of ‘softer’ factors (human, organisational, social) in assessing development suitability.

ii) It is more difficult to elicit soft information

iii) Not all the questions may be answered or are relevant whilst some are more important at particular phases of development.

These problems are symptomatic of computer personnel adopting an implicit technical viewpoint in addressing what are multi-dimensional factors: but they also reflects the difficulties of combining hard and soft criteria in the same assessment process. To resolve this problem requires training and guidance, with equal emphasis being made on defining the context of how the check-list is be used, as much as on defining what the questions mean. Although it is ambitious to expect the check-list to induce shifts in perspective of computer personnel, it is hoped that by presenting the check-list as a *process* of evaluation made up of a diverse range of multi-dimensional factors, a more balanced viewpoint in assessment will materialise. This aim is assisted greatly when representatives from different groups are actively involved in the assessment process. However one should be aware that using the check-list in this way is a substantial information gathering exercise and is intensive of other peoples’ time. For instance, during the evaluation of the computer hardware troubleshooting proposal, it was necessary to liase with senior management, users, experts, project management, software and hardware vendors and members of Computing and Finance departments. Other organisations, may find such a depth of consultation unacceptable or unattainable for political or resource and time reasons.

Appendix XI.

Cost Calculations for Chapter Seven

Box 1: Profiling The Costs of the Expert

From Equation 7.4:

Personal Call Level = (Net Working Day[1])/(Call Loading[2])

= 383 - 4.5 = **85 calls per day**[3]

Thus the personal Call level of the expert exceeds the Actual Call Rate[4]. *From Equation 7.5:*

Thus the effective staff required = (Actual Call Rate) / Personal Call Rate
= 67 - 85 = **0.79 Staff**

However, since this loading is distributed among four experts not one, the personal staff loading is only one quarter of this, and therefore, (0.79 - 4) = **0.195**

Thus, approximately one fifth of the expert's time is taken up in troubleshooting. Representing this in terms of hours of commitment per day, we get

Total Expert Commitment to Troubleshooting= (No. of calls per day)x Call Duration
= 67 x 4.5= 301.5 minutes / day
= **5.025 hours per day**

* **Box 2: Analysis of Payback**

At Payback, Total Costs of the Helpdesk =Total Costs of As-Is Situation, therefore at Year Á,

Helpdesk Development Costs + (Helpdesk Operating Costs).Á = (As-Is Operating Costs).Á

---

* **Notes to Box 1:**

[1] Net Working Day = 75% of working day = (0.75 x 8.5) = **383 minutes**

[2] Average Duration of Calls for the Expert = **4.5 minutes**

This assumes that there is a faster response to routine problems than the helpdesk but that because the expert also addresses more complex problems, the average duration of calls balances out to around 4.5 minutes.

[3] The call loading of the expert was determined to be **85** calls per day. However this figure represents an average and calls vary according to teh expert & day of the week & time of the day.

[4] The Actual Call Rate is the average daily call rate and was determined to be **67.**

Therefore payback rate, $\acute{A} = \frac{\text{Helpdesk Development Costs}}{\text{(As-Is - Helpdesk Operating Costs)}}$

$\acute{A} = \frac{12144.44}{(29{,}275.06 - 22580.56)}$ $\quad$ **$\acute{A}$ = 1.81 Years**

At Year 1: Payback is (Total As-Is Costs) - (Total Helpdesk Costs)

Thus (29, 275.06) - (£12144.44 + 22,580.16) = **- £5449.54**

At Year 2: Payback is (Total As-Is Costs) - (Total Helpdesk Costs)
Since there are no development costs in the second year we get,

£29, 275.06 -(£22,580.16 + **£5449.54**[a]) = **+ £1245.36**

Similarly, at Year 3: Payback = 29,275.06 - (22,745.23) + (1245.36) = **£ 7940.26**

Box 3: The Effects of Increasing Maintenance Costs

On the basis that the costs of helpdesk maintenance were 10% of Development Costs and the development costs were calculated at £12144.44 then,

If the ratio of maintenance to development costs increases to 30% and 50% respectively then, Maintenance costs at 30% = £3643.33 and at 50% = £6072.22. This changes the operating costs of the helpdesk as follows (*figures in bold represent a change*)

Cost factor	**£ / annum**
Hardware Depreciation	460.00
Hardware Maintenance	184.00
Maintenance Costs	**3643.33** (at 30% increase)
or	**6072.22** (at 50% increase)
Operator costs	8000.00
Expert Referral	7094.12
Call Costs	5628.00
Total	**£25009.45** (30%) or **£27438.34** (60%)

If Helpdesk development costs remain the same, this changes the payback in the following way:-

At 30% payback = $\acute{A} = \frac{12144.44}{(29{,}275.06 - 25009.45)}$ = **2.85 years**

At 50% payback = $\acute{A} = \frac{12144.44}{(29{,}275.06 - 27438.34)}$ = **6.61 years**

Box 4 The Effects of Changing the Costs of Expertise by ± 25%:

If the cost of expertise to the ompany is £19.61 per hour then changing this rate by ± 25% gives,
At + 25%, cost of troubleshooting = £24.51 per hour
At - 25%, cost of troubleshooting = £14.71 per hour

These change both the development and on-going costs of the helpdesk and the on-going costs of the as-is situation in the following way:-

a) Changes to the Helpdesk Development Costs: If expert commitment = 204 hours during development, then costs are:

---

[a] The losses made in the first year are carried into the next year

204 x 14.71 = £3000.84 (at -25%) and
204 x 24.51 = £5000.04 (at +25%). Development costs change as follows:-

Cost factor	**£**
Hardware Depreciation	460.00
Hardware Maintenance	184.00
Software Tool	1500.00
Development Staff	6000.00
Expert Involvement	**3000.84** (at -25%)
OR	**5000.04** (at +25%)
Total	**£11144.84** (-25%) or **£13144.04** (+25%)

b) Changes to Helpdesk Operation Costs: In this the costs of expert referral and maintenance costs change. Thus if the normal costs of troubleshooting are £23, 647 per annum, this changes to:-25% of £23647 = **£17735.25** and +25% of £23647 = **£29588.75**

If the costs of expert referral are 30% of total costs of troubleshooting then at -25%, costs = 30% of £17735 = £5320.58 and at +25% costs = 30% of £29588 = £8867.63. Operational costs change as follows:-

Cost factor	**£ / annum**
Hardware Depreciation	460.00
Hardware Maintenance	184.00
Maintenance Costs	**1314.4 and 1114.48**
Operator costs	8000.00
Expert Referral	**5320.58 and 8867.63**
Call Costs	5628.00
Total	**£20,707.14** (-25%) **and £24454.03** (+25%)

c) During AS-IS operations: Here, the cost of expertise changes as before, such that :Cost of troubleshooting = £17, 735.25 (minus 25%) and £29, 558.75 (£23647 plus 25%)

Allowing for the cost of call charges (£5628 per annum) total ongoing costs come to **£23,363.25** and **£35.186.75**

d) Payback: On the basis of the above figures we get:-

At -25% of costs of expertise:

$$\text{Payback} = \frac{11,144.84}{(23,363.25 - 20,707.14)} = \mathbf{4.2\ years}$$

At +25% of costs of expertise,

$$\text{Payback} = \frac{13,144.04}{(35186.75 - 24454.03)} = \mathbf{1.2\ years}$$

Box 5: The Effects of Increasing Call Rates:

At Year 1, Call Rate= 73

Staffing level = Call Rate/ Personal Call Level
Therefore: for helpdesk, staffing level = 73/69 = 1
for experts , staffing level = 73/85 = 0.86 or 0.22 loading per expert.

If a call charge is £0.35 per call then call loading = £0.35 x 73 =£ 25.55 per day or **£6132** per annum.

If the expert's personal call duration is 4.5 minutes per call then expert commitment to troubleshooting

= 73 x 4.5 = **5.475 hours per day**

If the cost of expertise is £19.61 per hour then cost of troubleshooting is 5.475 x 19.6 = £107.37 per day or **£25,767.54 per annum**

<u>At Year 2, Call Rate = 79</u>

Helpdesk, staffing level = 79/69 = 1.15
For experts , staffing level = 79/85 = 0.93 or 0.23 loading per expert

As for year 1, call loading = 0.35 x 79 = £27.65 per day or **£6636 per annum**

Expert commitment to troubleshooting = 79 x 4.5 = **5.925 hrs/day**

Cost of troubleshooting = 19.6 x 5.925 = £116.19 per day or **£27,885.42 per annum**

<u>At Year 3, Call Rate = 88</u>

Helpdesk, staffing level = 88/69 = 1.28
For experts , staffing level = 88/85 = 1.04 or 0.26 loading per expert

Call loading = 0.35 x 88 = £30.8 per day or **£7392 per annum**

Expert commitment to troubleshooting = 88 x 4.5 = **6.6** hours per day

Cost of troubleshooting = 19.6 x 6.6 = £129.426 per day or **£31,062.24 per annum**

Calculating the effects upon costs of these changes:-

a) Helpdesk Development Costs: No change at £12,144.44

b) AS- situation on-going costs: Both call charges and expertise costs increase as follows:-

Year One: Call loading= £6132 per annum
Expert costs= £25,767.54 per annum
Total Cost = £31,899.54

Year Two Call loading= £6636 per annum
Expert costs= £27,885.42 per annum
Total Cost = £34521.42

Year Three Call loading= £7392 per annum
Expert costs= £31,062.24 per annum
Total Cost = £38,454.24

c) Helpdesk On-going Costs: The cost of expert referral and call charges change as follows:

Year 1: Cost of expert referral is 30% of total costs
thus 30% of £25767.54 = **£7730.26**
and call charges rise to= **£6132.00**

Adding these increases to the other unchanged operating costs, the total cost of operating the helpdesk amounts to **£23,720.70**

Year 2: Cost of expert referral= 30% of £27,885.42= **£8365.6**
Call charges = **£6636**

Total Operating costs rise to **£24860.04**

Year 3: Cost of expert referral= 30% of £31062.24= **£9318.7**
Call Charges = **£7392**

Total operating costs rise to **£26569.14**

d) Operating benefits

i) Payback Using the Figures of Year 1:

Payback = Helpdesk development costs / (cost difference of operating costs)

= 12144.44 / (31899-23720) = **1.72 years**

ii) In **Year 1** return on helpdesk = 31899 - (12144.44+ 23720) = **- £3966.14**

In Year 2: return on helpdesk= 34521.42- (24860+ 3966) = **+ £5695**.28

In Year 3: return on helpdesk= 38454.24- 26596 + 5695 = + **£17580**

Box 6: Effects of Increasing Development Times: Calculations

Increasing the times of development will affect the Manpower costs as follows:-

a) Systems Developer: cost of services = £600 per month If the time over run is 4 months then the costs increase by 4 x 600 =£2400 giving a total cost of £8400 .

b) Expert Involvement: If the new time accounting for overruns is 48wks at 3 hours a week for each of the four experts then the total number of hours is (3 x 4 x 34) = 408 hours

If the cost of expertise is £19.61 per hour then the cost is £8000.88

Thus the development cost structure changes as follows:-

Cost	**£**
Hardware	644
Software	1500
Development Manpower	**8400**
Expert Involvement	**800.88**
Total	**18544.88**

c) Helpdesk Maintenance: Since the costs of development have increased, then on the basis that maintenance costs are 10% of development costs then these should increase also,

Maintenance costs = 10% of £18544.88 = £1854.49

Thus, operating costs increase to become £23220.61

Thus payback is (18544.88) / (29.275.06 - 23,220.61) = **3.06 years**

Appendix XII. Expert System Tools:

An Analysis of Vendors in the United Kingdom

Contents:-

Appendix XII: Section A

An Exhibit of the Vendor Questionnaire

Notes on Completion

This questionnaire requires a detailed understanding of the product and should be completed by the Technical Manager or a similar member of staff.

The questionnaire refers to the principal expert system offered by your company. However, if there is more than one, please make use of the additional copy(ies) provided.

Section A- General Details

A1. Name of Business:

A2a. Approximately how long has the company been in operation? ________ (Years)

b. Are you a producer or Supplier of Expert Systems?

A3. Name of Expert System:

A4. Brief Description of system:

A5. Price Range (£):

A6 Please indicate the type of user the expert system is intended. Please tick one or more of the following boxes:-

☐ Naive User

☐ Domain Expert

☐ Lisp/ Prolog Programmers

☐ Knowledge Engineers

☐ Professional Systems Developers

Others, Please Specify:

A7. Target Application:

a) Approximately, how many users are there in the UK?

b) Are there any particular fields in which you specialise? If Yes, please describe:

Section B- Operating Environment

B1. What Hardware does the system run on? *Tick one or more of the following : -*

IBM P.C. Compatible ☐ workstation ☐ A.I. workstation ☐ Mainframe ☐

B2. What are the System's size restraints?

Minimum Memory Requirements:_______________Kb.
Minimum Disk Requirements:___________________Kb.
Maximum Number of Rules (Approx.):____________
Is the system Memory Dependent? YES \ NO *(Please Circle)*

Are there any other hardware/software restraints? *Please describe:*

B3. What is the development tool? *Tick one or more of the following : -*

Shell ☐ Toolkit ☐ A.I Language ☐

'Environment' ☐ 'Off-the-Shelf' Package ☐

Other ☐ *Please specify :*

B4.

If a Shell, please answer the following. Otherwise go directly to Question B5

i) What version of Shell is currently available?

ii) Are any future upgrades planned? YES \ NO *(Please Circle)*

iii) What is the source language of the Shell (e.g. Pascal, 'C', Lisp, Prolog etc.)__________

iv) Is the shell **Rule-based** or **Inductive**?

a) **If rule-based:-**
Is there a Rule Directory? YES \ NO *(Please Circle)*

Are there explanations of each rule? YES \ NO

How are rules updated or modified? *(Please describe)*

Can rules be tested automatically? YES \ NO

b) **If Inductive:-**

Can the system write its own rules? YES \ NO

How are rules updated or modified? *(Please describe)*

5

If the Expert System Uses an 'A.I. Language' answer the following, Otherwise proceed to Section C.

What is the language code? *Tick one of the following:*

Lisp (*including dialects)* ☐ Prolog (*including dialects*) ☐

OPS5 (*and derivatives*) ☐ Poplog ☐ Smalltalk ☐

☐ **Others**. *Please Specify:*

B6. Does the language tool have a Compiler or Interpretter or Both?

B7. If a shell, is the A.I. language used as the source code or to 'customise' the shell only?

Section C - Development Environment

C1. How is Knowledge Represented in the Expert System? *Tick one or more of the following:*

Rules ☐ Frames ☐ Semantic Nets ☐

Attribute Value Pairs/Triplets ☐ Procedural Attachment ☐ Viewpoints ☐

Object Hierarchies ☐ Inheritance ☐ Others *(Specify)* ☐

C2. Which of the following control strategies does it have? *Tick one or more of the following:*

Forward rule Chaining ☐ Backward rule Chaining ☐ Both ☐

Pattern Matching ☐ Procedural Control ☐ Conflict Resolution ☐

Utility Optimisation ☐ Network Traversal ☐

Others. *Please Specify* ☐

C3. How does the system deal with uncertainty? *Of the following, tick those which it uses:-*

Certainty Factors ☐ Probability Logic (Specify type) ☐

Bayesian ☐ Fuzzy Theory (Specify type) ☐

Others ☐ *(Please Specify)*

C4 User Interface/ Explanation Facilities

Which of the following facilities does the **basic** system have. *Please tick.*	YES	NO
Embedded Capabilities (drives other packages)		
Software 'Hooks'(can link to other software- e.g spreadsheets etc)		
Capable of dealing with mathematical functions		
Automatic Update		
Graphics capabilities		
Allows interrogation of the knowledge-base		
Displays reasoning steps		
Word search		
Cross-Referencing		
Demons		
Menu-Driven		
Colour		
Windows		
Is there on-screen help available		
Are there context-sensitive help facilities		
Can the system handle real-time input data		
Can the system handle numerical input data		

Please add any other special features of the system :-

Section D: Applications

GEC is particularly interested in manufacturing and engineering applications of expert systems. Have your systems been used, by yourselves or by your clients, in any of the following areas. ***Please tick and give examples where possible.***

Domain Area	**Please tick**	**Please give examples where possible**
Diagnosis (Esp. Faults)		
Process control		
Planning/ Scheduling		
Selection		
configuration		
CAD		
CAM		
Management Systems		
Monitoring		
Interpretation		
Estimating/ Cost-control		
Simulation		
Prediction		
Tuition / Training/ Advice		
Stock Control		
Quality Control /Assurance		
Personnel		
Others		

Section E- Support

E1. User Support

a) Are manuals available? *Please tick one or more of the following*

Users? ☐ Systems? ☐
Programmer? ☐ Operations? ☐

b) Is there Source Code Available Yes / No? *(Please circle)*

If Yes, at extra cost? Yes / No?

c) Support Environment
Please indicate which of the following services are available:

Training, ☐
Bug-Fixes, ☐
Software upgrades ☐
Maintenance ☐
Others *(specify)* ☐
Hot-line, ☐
Customerisation ☐
Newsletter. ☐
programming service ☐
Applications development, ☐
feasibility/identification ☐
requirements analysis, ☐
knowledge engineering ☐

E2. Security.

Is it possible to assign and update passwords? YES \ NO *(Please Circle)*

Is it possible to restrict access to the knowledge base? YES \ NO

E3. Is there a 'Run-Time' version available? YES \ NO

E4. Are there Royalties (i.e. Licence Costs) ? YES \ NO

E5. Is the system Modular (i.e. can enhancements, new versions etc. be added-on) YES \ NO

E6. Is there a built- in programming language? YES \ NO

E7. Is there a demonstration or introductory disk available? YES \ NO
(If so, a copy would be most useful in our evaluation)

Please return the questionnaire in the envelope provided

Thank-You for Participating

Appendix XII: Section B

Vendor Questionnaire: Definitions & Assumptions

Section A: General

(A2) If it assumed that expert systems have been marketed seriously for only the last ten years, then the age of the organisation will reveal whether it was set up specifically for expert systems development or whether it is an established computer business which has diversified into this field. It will also give some indication of how long the company has been marketing the product. It is also useful to know whether the company operates as a producer of expert systems, or supplier (or both) . Generally, suppliers have less resources and support capabilities than producers.

(A5) The price range indicates the varied costs of systems according to the hardware used. For instance, many companies develop a basic tool for the PC XT machine as a means of demonstrating the product's potential. This tool is then upgraded by the addition of modules which add to the tool's capabilities. The same tool may also be modified for use on higher specification PCs or workstations. The difference in costs may be substantial between machines. The costs used in the questionnaire are those for the basic tool or shell operating on the minimum specification machine possible, where the realistic minimum is taken as being a PC AT (286) machine.

(A6) It is important to understand who, among a range of users, from the computer naive to the professional systems developer, the system is intended. Expert systems have been heralded as the layman's programming tool. However this refers to their ease of use rather than their ease of development. Infact, expert systems development requires particular skills, with some products requiring more than others. Understanding the role required of the developer, will give some indication of the level of complexity and the skills necessary for effective use of the product.

(A7a) Unless new to the market, the number of expert systems sold is usually a good indicator of product and/or service quality. A large customer base also facilitates improved design on the basis of feedback from users. Clearly shells are aimed at a more popular and widespread audience than for example large environments, and therefore comparisons should be made within the categories stated in question B3.

(A7b) Expert systems should not be considered as "general problem solvers". They are useful to a specific class of problem where human expertise is the key factor. Within this class however, there are many types of application ranging from diagnosis to simulation. Some products specialise in particular application areas, process control for example, whilst others are marketed on the strength of their functionality. An awareness of the intended role for the product will assist in matching tools to the problem.

Section B: Operating Environment

(B1) Hardware was divided into four categories: Personal Computers (AT machines and above); Workstations; A.I. Workstations (these are dedicated machines which operate within a artificial intelligence based environment and may be used to develop a range of run-time expert applications); and mainframes. The respective systems disk and memory requirements are detailed in question **B2**.

(B2) The choice of development tool is constrained by the hardware available-environments require more hardware power than shells for instance-and by the nature of the problem. Using the classification used by Ovum and CRI, the range of tools is divided into five broad categories:-

i) Shells: essentially, these are complete expert systems but without the knowledge component. Their strengths lie in the relative ease of programming and the sophisticated explanation and development aids. However, they are restricted in the types of problems they can solve and are difficult to integrate with other systems. Shells are a relatively low-cost entry into expert systems and yet, their use may yield high returns.

(B3) There are two types of shells, Rule-based and Inductive, distinguished by the way in which knowledge is encoded. In the former, knowledge is represented by explicit procedural statements or rules- IF-THEN for example. By contrast, Inductive systems generate rules and define relationships automatically by analysis of sets of data or from example. Inductive systems are applicable to a very limited class of applications and are therefore less popular than rule-based shells.

ii) A.I. Languages: these include Prolog and Lisp and their respective derivatives. Using languages allows for an unlimited number of problems to be solved. However, considerable expertise is required to use these languages and all user and help facilities have to be defined by programming rather than being automatically supplied as with shells.

iii) Toolkits: This category combines a number of the attributes of shells with the functionality of languages to provide a more productive and usable system.

iv) Environments: These are the "Rolls Royces" of the expert systems world. They provide an A.I. programming and development.environment complete with analysis tools, sophisticated design models and automatic code generation all operating within a common, user-driven interface.They represent, in financial and resource terms, a major commitment to expert systems.

Section C: Development Environment

i) Knowledge Representation

(C1) There are many different ways of representing knowledge, each method being more appropriate in some cases than others. There is a trade-off between the number of knowledge representation techniques supported by the expert system and the additional costs and complexity that each subsequent technique adds to the total. Most shells will have at most two techniques, whilst environments will have up to sixty. However, in designing the questionnaire, the following were judged to be the most useful and important techniques:-

a) *Rules*: The single most frequently used technique and probably the easiest to use. Rules are used to represent knowledge which is "shallow"- i.e., a broad level collection of isolated facts.

b) *Frames*: This techniques is used to represent "deep knowledge" where there are identifiable patterns, hierarchies or relationships and therefore allows for deep-level reasoning such as abstraction and analogy. Frames are used to describe objects in details and contain subdivisions known as attributes which describe a single aspect or relationship with the frame. For example, one frame might represent a particular class of locomotive and within this class there are attributes or "slots" containing information on engines, bogeys, control gear, carriages and so on.

c) Semantic Nets: A semantic net is a network of nodes linked together by arcs. Nodes stand for general concepts, specific objects, general events or specific events. Arcs describe relationships between nodes. For example, nodes might be train and engine, and the arc could be "has-a", thereby adding a relationship to the nodes.

d) *Attribute Value Pairs:* This is the most elementary form of knowledge representation. These are variables (attributes) that can obtain values. This kind of knowledge representation is used for unstructured data types in many tools, the value of an attribute cannot be changed once it is determined.

e) *Procedural Attachment*: this facilitates object orientated programming. A couple of lines of code (rules or a general purpose programming language) can be attached to an attribute to determine how to obtain a value.

f) *Viewpoints*: These are basically subsets from a database. A viewpoint mechanism, with a system that remembers the way facts are inferred, enables the simultaneous investigation of several lines of reasoning. This is a very effective reasoning technique for simulation.

g) *Object Hierarchies*: In this form of representation, there are parent-children links between objects. These links usually represent relations like 'is an instance of', or 'is a subproblem of'. There are usually one or two relations available but the user cannot define relations of his own.

h) Inheritance: this is an additional feature of a semantic network or object hierarchy. Values of attributes can be inherited over a relation. This inheritance can be specified for each relation. The objects that are related to each other obtain each others' information automatically.

j) *Others*: Twenty seven further knowledge representation techniques were mentioned by vendors. However, these were mainly derivatives of the above in some modified format.

ii) Control Strategy

All expert systems are forced in a certain direction of reasoning determined by the control strategy. This is the way knowledge is used to establish new facts and come up with a result. As with knowledge representation, there is a plethora of control strategies available according to the characteristics of the problem. However, most are a subset of the following:-

a) Forward-rule chaining: conclusions are drawn from established facts. Forward-chaining is used mainly for advisory and planning systems.

b) Backward-chaining: is trying to prove a hypothesis or goal by evaluating the premises of that goal. Backward chaining is often used for fault diagnosis. If a tool provides both ways of inference, it is more generally applicable.

c) Pattern-Matching: some of the more advanced tools are able to recognise not just relationships between objects through inheritance, but causality between groups of objects making up patterns. This higher level of recognition allows for the use of rapid search and action techniques in monitoring and on-line systems.

d) Procedural Control: this process is necessary to avoid senseless direction of reasoning. It is also applicable in areas where human knowledge is purely procedural-such as the interpretation of regulations/legislation for instance.

e) Conflict Resolution: If more than one rule is applicable to a certain situation it has to be decided which rule should be executed. Prolog takes the first applicable rule: several tools however, try to apply the best possible rule according to a certain strategy which may be defined by the user.

f) Utility Optimization: Where conflict resolution follows a strategy, utility optimization surveys all possible rules and adopts that which produces the highest possible cumulative probability.

g) Network Traversal: This is the ability to switch knowledge-representation techniques or network of causal relationships and adopt a completely different line of reasoning using a different control strategy.

iii) Handling Uncertainty

When an expert system asks questions, the user may not be able to supply the desired answers either because the answers may not be known at all, or the user is uncertain over its validity. Most expert systems deal with uncertainty using either structural or numerical methods, the following of which are the most widely used:-

a) Certainty Factors: these are numerical measures of the confidence held in the validity of a fact or rule. It allows the inferencing program to work with inexact information, for example in scoring techniques used for configuration and selection type expert systems.

b) Probability Logic/ Bayesian Probability: these techniques use mathematical functions to calculate the probability of some event occurring based on the fact that other events have occurred. For example, if the petrol gauge drops to zero, then there is a high probability that the car will soon stop running. Depending upon the type of knowledge involved, probability may be a more suitable way to deal with uncertainty than certainty factors. In general though, these are more difficult to implement.

c) Fuzzy Logic/ Fuzzy Theory: Fuzzy reasoning is helpful in dealing with imprecise information. It accepts that often, there are no clear divisions between categories of problems and that experts' responses may differ. Therefore fuzzy logic creates broad categories which leave choice open to the discretion of the expert. Fuzzy reasoning is also useful because it makes it possible to assign a numerical value to what may appear to the user or expert as a qualitative decision.

iv) Interface & Explanation Facilities

The following facilities should be considered in defining the requirements of ES software:-

a) Embedded Capabilities: To perform its problem-solving function, the expert system may need to perform some calculations or do other jobs best assigned to algorithmic routines. The algorithms are embedded in the expert system software, which refers to them when necessary.

b) Software Hooks: These allow the expert systems to communicate with one another or with conventional programs in order to get the inputs they need to solve problems. Most shells use software hooks to link up with external software packages such as spreadsheets or database systems.

c) Mathematical Functions: Some shells have computational capbility that allows them to perform mathematical operations. The problem may require calculations on input data prior to a search of the rule base for example. Most shells have only a very limited selection of mathematical functions.

d) Updates: Some of the inductive rule-based systems and the more complex toolkits allow for the automatic update of rules based upon recorded events.

e) Graphics: Some real-time applications require graphic models or visual screen guidance. Furthermore, the presentation of results are more lucid in graphical form. If the system does not have graphics incorporated into the software, then it is useful to know whether there are software hooks to other graphics programs.

f) Knowledge-base interrogation: It is often necessary to change, restructure and update the knowledge base and the efficiency by which this is done is determined by how easy it is to interrogate the knowledge-base. Different packages provide different facilities for modification depending upon the knowledge-representation technique.

g) Displays reasoning steps: A useful facility to have during development and operations is one which describes how rules have been executed and the direction of reasoning during consultation. This makes verification easier and is also an important source of information for the user.

h) Menu-driven: For ease of use and efficiency, menus are the most effective means of communicating with the user at a relatively high level of complexity. Most expert system tools use menu facilities of some fashion.

i) Demons: These are programming labels which, when encountered, inform the inference module to perform a specific task or follow another predetermined line of reasoning.

j) Word Search: This facility is especially useful when programming and maintaining the system in order to ensure consistency.

k) Windows: Windows is a useful facility to have since it allows a number of consultations, in different areas of the knowledge-base, to take place.

l) Help: Help facilities are essential at the development level- define syntax errors, invalid rule statements etc-and at the operational level where the user may have difficulty using the system or understanding the questions. Most systems offer one or two levels of help. The first displays the rule which was last executed (or failed to execute) from which the user is expected to elicit the required information. A more satisfactory approach is to provide context sensitive help where each help screen relates directly ,and in varying levels of detail (using Hypertext for example), to the problem experienced by the user.

m) Real-time inputs: For on-line systems and process control for example, it is essential that the expert system can handle real-time and numerical inputs directly. Many shells are unable to accept information directly and require the data-handling facilities of linked conventional software which significantly slows down the speed of transaction.

Section D: Applications

Using a similar classification for manufacturing industries used ·by Ovum Consultants and the Department of Trade & Industry, applications were divided into seventeen categories. Most are generic applications, such as selection, interpretation and configuration; however, a few refer to specific tasks such as stock control and computer aided design.

Most respondees to the questionnaire highlighted where their customer base was focused, in terms of application profile, and **Section 4.5** lists specific applications which were quoted. Further information on each of these applications is available upon request.

Section E: Support

(E1) Good documentation will ease the learning process and use of the expert system. However, the emphasis of the documentation will vary according to whether it is intended for the User, the Knowledge Engineer, Expert or Programmer. Furthermore, the manual may be biased towards the development phase, or conversely the operational phase. Some manuals have a general manual for all these people and levels, whilst other companies offer detailed manuals for each.

An important part of tool assessment is understanding precisely what support is offered upon purchase. These range from a simple customer hot-line for simple problems, to a full customer development programme which would include requirements analysis, training, design and implementation for the first application. Companies supplying Shells tend to concentrate on 'arms-length' support, such as newsletters and hot-lines, whilst Environments show a greater involvement through on-site training, knowledge engineering and even maintenance of operational systems.

(E2) For intended applications where the knowledge held in the expert system may be confidential, and in other instances where it is important that users are not allowed to have access to the knowledge base, then security is an essential consideration. Many systems offer 'run-time' versions where the knowledge-base is completely unaccessible to all users, whereas other systems have levels of access into the knowledge-base according to the identification of the user.

(E4) Some vendors require a royalty or 'use fee' on distribution copies of expert systems using their product. This can be a major expense if many copies of the expert system are to be distributed. Other vendors incorporate a license cost into the sale price and allow unrestricted use of the software thereafter.

(E5) If an expert system is described as being 'modular', then it is possible to add-on other functions and upgrade the software to a higher specification tool. It also indicates that the software can communicate with new or old versions of the software.

(E6) As experience in developing expert systems increases, then developers may wish to link or integrate the expert system with company data-bases or company-specific programs in which case, the system requires a built-in programming language. This allows interfaces to be written to other conventional systems, the format of which will vary according to which built-in language is used. Many shells use the language ‘C’ or ‘C+’ because of its functionality with other source code.

Appendix XII: Section C

An Analysis of Expert System Vendors

1. Analysis of the Market

This section looks at the characteristics of expert systems suppliers and their operations, policy and future product and servicing strategy.

i) The State of the Expert Systems Market in 1990

The expert systems (ES) market has seen change in the last three years. There have been a number of mergers and take overs which suggests some contraction, but more a consolidation of the market into a number of increasingly profitable organisations. Suppliers of Expert Systems (referred to henceforth as 'Vendors') have ceased to hype their products as being different and now emphasize them as becoming part of the mainstream of the Information Technology(IT) industry. Their products are positioned in the market as practical software development tools, offering direct advantages in terms of productivity and speed of development.

The integration with mainstream IT is proceeding on two fronts. First, expert systems are increasingly being developed by computing departments in manufacturing rather than specialised expert system groups. Second, knowledge-based systems and expert system tools are being seen, and sold, as part of a set of tools for developing applications and achieving strategies towards computer integration. On this basis, there are three likely development trends which will have significant impact on businesses:-

a) Tools that combine knowledge management and data management in a single structure or formalism (for instance G-base or Generis).

b) Generalised application tools for design (ICAD, Concept Modeller), diagnosis (Testbench) and simulation (Stem and Simkit).

c) Application specific tools for tasks such as circuit-board diagnosis (Synergist) and machine vibration interpretation (Violet).

ii) The average age of Vendors

The age of vendors is a good indication of the strength of the product and service and may also indicate something about the structure and operations of the company. The average age of vendors supplying shell based tools was significantly less than those supplying tools and environments (Figure 14a, 14b). Infact nearly 55% of vendors supplying shells had been in operation for less than five years. By contrast 70% of tool and environment suppliers had operated for over eight years, with 36% over eighteen years old. These differences reflect the recent emergence of shells as an alternative to the more complex and longstanding use of Artificial intelligence languages.

A further reason lies in the relative success in marketing shells as a low cost, low risk and (alleged) easy-to-use alternative to conventional computer programming. The mid-1980's therefore saw an onslaught of new businesses set up specifically to market a range of poor quality products which have only recently left the market. The history of tools and environments has been more stable. The research and development costs and support requirements usually meant that all but the largest organisations were capable of developing and supplying them commercially. In the case of shells and tools, Figure 14a & 14b, show that the great majority of vendors are both producers and suppliers of ES software. Of the vendors which supplied ES software, most were marketing well established North American and French products in the UK.

iii) Pricing Policy

The way in which an ES is produced through the use of development tool may be priced considerably higher than the delivery system. Serious ES development will require training and at least one dedicated developer. The return on investment realised depends much on the problem domain and to a lesser degree, on the pricing of the development system. Once all costs are considered in the project, the price of the development tool is often less than 10% of the total project investment. The price of shells alone varied from less than £500 to over £11,000 (Figure 14c). However most fell into the price bracket of £2,500-£5000. Tools and Environments are clearly directed at a higher priced market (although third generation shells such as Egeria are beginning to encroach on application territory which previously belonged exclusively to toolkits), with the average price of tools being £15,000 (Figure 14d). The cost of environments can rise to above £100,000, representing even for large manufacturing firms, a major strategic investment.

iv) Customer Base

The relatively low costs and 'successful' marketing of ES shells is reflected in the number of users in the UK. Almost 50% of vendors reported a customer base of over 200 users, with 5% of over 10,000 (Figure 16). By contrast, the cost and complexity of tools and environments have proved strong barriers to their widespread adoption, with 25% of vendors reporting a customer base of less than 10 users.

It is dangerous however to make comparisons between tool categories because each is aimed at significantly different markets. Furthermore, the size of the customer base is not the sole criterion to use in the choice of tool: what may appear successful in other organisations may be wholly inappropriate for the proposed application.

v) Support and Services

Vendor support is a necessity in any new technological field. Training for beginning and advanced developers is desirable. On-site training for organisations working with highly sensitive problem domains may also be necessary. Documentation is important; complete documentation is characterized by separate reference and user guides, complete indices and third party books offering code examples. The range of back-up services vary from simple consultations to complete turnkey development. A useful measure of how commited a vendor is to a particular form of support is to find out how many are employed in this section.

From the questionnaire (Figure 2), results show that there was generally a good representation of support functions in each of the areas identified as being important. Vendors of shells tended to adopt an 'arms-length' approach to support through newsletters, hot-lines and emphasis upon self-help and proper use of the manuals. This approach was taken because the large number of users and wide range of applications restricted the level of 'personal' help. Training and application development were usually available, but at extra cost, significantly, greater than the price of the software.

It is broadly true that as the complexity of the tool increases, there is a corresponding improvement in the range of support functions offered by the vendor. Most vendors of toolkits for example, offered maintenance and knowledge engineering services and free training; and vendors of environments managed the first application for the client as part of the training process. For the latter, the expert system is marketed more as a package of product and services, whilst shells focus their marketing at the product level.

2 Analysis of Requirements

This section focuses upon resource, product and service requirements specified by vendors in order to develop their systems

i) Hardware

The development hardware required for most expert systems is the Personal Computer (Figure 13a.). Within this category, most systems require an AT-specification machine (65%), with 15% needing a 386 model. In all cases, the higher the specification of machine, the faster and easier the ES becomes to use. Graphics facilities and now multi-tasking all improve the software performance and development productivity. Only 20% of ES software required a workstation for development and these tended to be for use by the more complex tools and environments. Figure 13a shows the decline in use of dedicated A.I. workstations for development, highlighting a general trend towards the use of expert systems on conventional data-processing and business hardware. Moreover, although the development of ES may require a relatively high specification machine, it is increasingly possible to operate the run-time or delivery system on a much lower specification of machine, typically AT personal computers.

Linking ES software to mainframe computer systems is an important element of mainstream IT integration. However development on mainframe systems is often difficult without previous knowledge of the ES working on a PC or workstation. Furthermore, mainframe expert systems represent a large financial commitment which may be unwarranted without previous demonstration of capabilities on lower specification machines. For this reason, vendors have begun to market upgrades of shells and tools which may be transported directly onto mainframe systems. Results from the questionnaire show that 62% of shells could be upgraded onto mainframe systems, whereas for tools and environments, this figure was reduced to 47%, despite being potentially more worthwhile.

ii) Software

In identifying an appropriate tool for a problem, the developer should start at the highest programming level possible and work downwards. A company commitment towards expert systems should be based on a set of varied tools. Standardizing all expert system projects to one tool will fail, unless the organisation only tackles projects which are suited to the tool in use. While normal I.T. policy will try to standardise on software tools, this is not feasible for expert systems work which must be based on a library of tools.

The ES market is still orientated towards the supply of shells which account for more than half of the market share (Figure 13b). This indicates the popularity of rule-based knowledge representation, but it also reflects the relatively low level of commitment afforded to ES technology by manufacturing organisations. ES tools and environments, making up 36% and 9% of the market respectively, require by necessity that they are more embedded in the organisation and integrated with existing conventional systems. Most shells, by contrast, are stand-alone, relatively low risk and often represent an 'experimental' and specultaive approach towards ES development. This view is substantiated by recent reports which show that many shell-based ES developments fail to reach the operational stage because of a lack of understanding of the concepts and requirements behind ES technology and a corresponding failure to commit resources (training, expertise, appropriate hardware and software etc) to the project.

Off-the-Shelf packages, that is ES shells sold with pre-coded knowledge of a particular domain, such as a pensions advisor and electronic circuits design aid, accounted for only 4% of the market share at the time of the study. However, there have been a number of systems entering the market since 1988 and it is predicted that this type of ES will become more important in the next five years.

The construction of shells strongly favour the use of rule-based techniques, IF_THEN statements for example, to other techniques, with 70% of al shells being of this type (Figure 13c). The limitations of Inductive shells, such as XpertRule, to applied manufacturing problems account for their diminishing share in the total market, standing just under 8.5% in 1988. However, there are benefits in combining rule-based and inductive techniques, and the development of these combined shells, called 'hybrids', account for 20% of the shell based market.

As expected, the majority of toolkits and environments make use of Prolog or Lisp based languages (Figure 13d). More recent tools though have combined the respective declarative and procedural attributes of both languages within a high-level language environment. These customised and self-developed systems, such as 'Flex', now account for almost a third of the tool market.

iii) Personnel

An important issue in expert systems is who should be charged with their development? The supposed value of shells is that they allow users and experts to develop applications themselves. Figure 1.1 shows however, that although shells are not intended for the programmer, the most appropriate developer would be either the knowledge engineer allocated specifically to the project, or the professional systems engineer. As with conventional programming, expert systems require proper project management if they are to be successful. Expert systems development should therefore be managed by the existing computing function and not be heralded as a means of deinstitutionalizing computing in the organisation.

Figure 1.2, shows that programming skills are essential for the use of A.I. tools and languages and the naive user, as with shells, was one of the least preferred 'target' users. For environments (Figure 1.3), the preferred user is strictly professional programmers, knowledge engineers or system analysts with no direct development role for the naive user or domain expert.

3. Analysis of Development Requirements

It is difficult to understand and appreciate what requirements are defined by an application and subsequently, how to choose the most appropriate implementation tool. Problems arise because developers use ES software which they are familiar with rather than evaluating a possibly more suitable one; or they may be given no choice because their company has adopted a standard 'house shell' with which all their ES problems can be solved. It is not possible to exhaustively list all the desirable features of ES software as criteria, since these will be specific to each application. However for the purpose of evaluation in the questionnaire, the following factors were applied:-

i) Knowledge-representation techniques

Stored expertise exists in many formats other than reasoning knowledge. Most organisations use descriptive knowledge, procedural knowledge and presentation knowledge. Descriptive knowledge is found in databases and text documents; procedural knowledge is composed of spreadsheet models and programs written in C, Cobol, Pascal, or other languages; and presentation knowledge is widely used as forms, report templates and graphs. These sources need to be directly accessible by the reasoning or 'symbolic' knowledge of the expert. The variety of knowledge types reflects the limitations of using a single knowledge-representation technique, such as production rules. Rules have the disadvantage of being difficult to localise and degbugg, a relatively weak rule syntax, the absence of object-orientated programming capability and speed (most rule based techniques are much slower than conventional software). The greatest justification for using rule-based techniques is with derivation type problems where relationships are expressed in terms of cause and effect, such as data driven systems or systems where the data is not well defined.

Figure 7, shows that virtually all shells use rules-based representation techniques, whilst less than half used other methods, the next most frequently used being Procedural Attachment, Frames, Attribute Value and Object Hierarchies. It is significant that the third generation shells (those developed after 1988) now have at least four knowledge representation techniques acknowledging that a manufacturinng problem may require more than one technique to be solved effectively.

The principal knowledge representation technique for tools remains production rules (Figure 8.), but there is a more important role for object hierarchies and frames. These provide an ease of representation of a variety of closely related objects. This is accomplished through inheritance (properties are inherited from a higher hierarchical class to a lower one) and message passing (automated transfer of information between objects and classes of objects). These techniques are used where problems are difficult to express in terms of rules and are more easily thought of in terms of goals and subgoals which are represented as objects.

Figure 9, shows that there is a fairly equal use of knowledge representation techniques for environments. The two main methods remain object-orientation and rule-based programming, but there is also a prominence of vendor's own methods specific to their operating environment. These non-standard methods make learning slow and difficult and constrain communication flexibility with other ES and conventional software.

ii) Knowledge Control Techniques

Knowledge control refers to the way knowledge in the knowledge base is executed. The part of the ES which performs this task is known as the inference engine. The inference engine must be able to process the stored expertise in a variety of ways if developers are to be assured that a system will meet the needs of the users. One item to look for is the ability to reason in a forward direction. Problems where there are few known variables, no stated goal, or the need to consider most or all rules require forward chaining. Backward chaining will be used where there is a specific goal and a number of unknown variables. Many problems demand a mix of these two processes; hence the main control strategy for shells (Figure 10), Toolkits (Figure 11) and Environments (Figure 12) is a dual forward and backward system.

A major benefit in using A.I. tools and languages is that most knowledge control techniques may be programmed, so that development can be truly customised to the needs of the application. However, this functionality must be measured against the difficulty of programming in Prolog or Lisp for the conventional programmer or User compared to the ease of use of shells. Environments offer a middle ground between the two, providing a broad selection of control techniques but incorporated within a supperior shell-like development and delivery environment.

ES software should ideally provide a way to selectively process subsets of rulesets, seek values for variables by making them temporary goals, and choose the order in which rules will be selected. Criteria used for rule selection order will vary according to the structure and requirements of the problem: but they might include priority, processing time for execution (cost), certainty, natural order, random order, fewest unknown variables in premise, or any combination of these.

iii) Development & Delivery Facilities

In general, facilities for development are more sophisticated and widespread in ES software than for delivery (Figure 3.). For instance, many shells were capable of displaying reasoning, had demons to improve programming efficiency, and allowed knowledge based interrogation during development; but very few had adequate user functions such as context sensitive help facilities, menu-driven screens and graphics. Tools and languages had poor development functions and delivery was dependent upon the quality of programming, and consequently many essential user facilities appear discretionary and are therefore neglected in design. Environments are more likely to enforce a user-orientated discipline by incorporating these functions as steps in the design process. This makes development highly effective, but often they have a considerable overhead in terms of their size and consequent effect upon performance. The more efficient the delivery system, the more cost effective the hardware that will support it- this is a main benefit of shells compared to more complex software and development tools.

Many of the shells were capable of linking to selected software such as DBase and Lotus 123, and could also manipulate mathematical functions and numerical inputs. However, few could support real-time applications (unless, like a number of off-the-shelf packages, they were specifically designed to do so). All the environments and the higher specification toolkits have some rudimentary facilities to help the real-time application developer.

iv) Handling Uncertainty

Although there is some controversy over mathematical correctness, it is useful to have facilities that assign numeric factors to rules and facts and thereby rank the 'worth' or likelihood of deductions made by the system. A number of techniques are utilized and these vary according to the ES software used. Figure 15., shows the uncertainty techniques used for each of the three categories of ES software. The main technique for shells and environments is 'Certainty Factors', followed in both cases by specific techniques characteristic to the software. The significant observation arising from Figure 15 is that most A.I. tools have no explicit uncertainty handling facility built into the software. However, they do have the capability to incorporate uncertainty but like control techniques, this requires skilled programming and a knowledge of uncertainty principles.

4 Analysis of Applications

Ovum have identified that operationally, the take-up of expert systems has been more intensive in the 'high-tech' industries such as electronics and the computing industry itself, whilst the level of usage in manufacturing remains very low. Many companies have experimented with ES technology at the shell level and progressed no further, this despite the fact that the greatest benefits will accrue when integrated with 'mainstream' I.T. applications.

It has been shown that ES software varies enormously in both structure and capabilities and for this reason, certain types will relevant to specific application categories. For example, for derivation type problems, such as diagnosis, interpretation and monitoring, rule-based or hybrid rule-based systems offer advantages over other representation techniques. For formation problems though, such as design, planning and prediction, object-based systems (using frames, object hierarchies and procedural attachment for example) will yield greater returns on programming effort. ES software can be a cost effective and efficient means of building computer systems and may also provide the only feasible means of 'computerising' some complex tasks. In both instances however, the success of the system will depend upon selecting the right application and specifying the correct tool.

Although User organisations may be under the impression that 'shells' are general problem solvers and have a wide portfolio of applications, infact their restrictive representation and knowledge control capabilities have constrained their effective use to a small band of applications specific to a problem domain. Furthermore, vendors may intend that their software is used specifically in certain applications by providing the appropriate customised functions. Figure 17 divides all ES software according to the intended role specified by the vendor. Four broad categories were identified: real-time applications(such as planning, scheduling and process control); Data-processing and IT interfacing (Database management systems, intelligent interfaces, Knowledge-based information systems etc); 'Others' (defined as small miscellaneous application categories); and 'General' (defined as being relevant to most categories of application). Only 38% of shells are intended for 'general' use, whereas the improved flexibility of tools and environments accounts for the reason why over 62% may be considered for general applications.

Figures 4, 5. and 6, give profiles of applications for customers of shells, A.I. toolkits and environments respectively. For shells, the most frequently recurring applications are diagnosis, monitoring, process control and tuition and training. There is also an increasing role for shells in personnel and quality control and other areas which appear to have a substantial qualitative element to reasoning or significant user interaction and involvement. The common expertise for these applications include finding consistent, correct and incorrect interpretations of the data and understanding the interactions between sub-systems. Information may be partial, contradictory, unreliable or poorly distributed in the organisation: and the problems tend to be broadly similar to ones which have previously been solved and implemented. This structure is ideal for rule-based shells, the benefits being the 'expressibility' of the rule format, modularity, modifiability, explainability, relative ease of learning and portability. Figure 4 shows that there are applications where shells have not been successful or are used infrequently. Most notable are the areas of Computer Aided Manufacture and Computer Aided Design.

Diagnosis is again significant (in terms of number of applications) for A.I. tools and languages. However, this function is often combined with monitoring, process control and simulation type activities The most important development for A.I. tools though, is becoming knowledge-based planning and scheduling systems and other real-time and integrated applications. Moreover, despite the costs and complexity, tools are beginning to be used more for constraint driven design systems in CAD. From Figure 5., tools appear inappropriate for personnel, quality control and estimating type problems and are used infrequently in CAM.

Environments have a very broad application profile, but their use is significant in the areas of simulation, planning, scheduling and prediction and other areas where a flexible frame based knowledge structure is required or where hypothetical reasoning is important.

Appendix XII: Section D:

Market-to-Users: Application Profiles

D1. Introduction

During the investigation, a number of applications were highlighted by the vendors. These had either been implemented and were operational in client organisations, or were at some stage of development. The following listings give some indication of where commercial shells, tools and environments have been used in the manufacturing sector.

D2. Expert System Shell Applications

a) Diagnosis:

System software diagnosis
Printed circuit board fault analysis
Computer hardware diagnostics
Network fault finding
Diesel engine fault debugging and detection
Maintenance testing equipment diagnosis
On-line networking plant diagnosis
Deriving optimum testing procedures from case histories of faults
Network monitoring on personal computers.
Circuit building diagnosis
Mechanical health diagnosis and monitoring
Programming Help Desk Adviser

b) Process Control

Fuel system manager
Exceptions handling
Alarm control
High-level control over engine test rigs
Energy management system

c) Planning & Scheduling

Maintenance Planning
Matching equipment to regulations (for tender purposes)
Database Multi-job scheduling
Manufacturing scheduling & planning
Job scheduling system (PC linked to Mainframe)
Timetabling system for managers
Resource timetabling
Discrete resource allocation
Computer-aided knowledge planning system
Manufacturing Cell control

d) Selection

Software package selection
Personnel regulations Adviser
Personnel selection Adviser
Intelligent front-end systems to assist naive users select the most appropriate action.
Training course / educational needs selector.
Various component based selection advisers
Heat treatment selection

e) Configuration

Software module configuration
Machine/hardware sales configuration system
Product selection (constraint driven)
Component assembly
Design configuration for manufacture
Configurer for circuit cabinets
System design for testability

f) Computer-Aided Design (CAD)

Design Quality reporting
Sensor design & placement
Graphical design of industrial plant
AUTOCAD interface
Various Intelligent CAD systems
Constraint driven design / Design for Manufacture
Helical spring Design

g) Computer-Aided Manufacture (CAM)

Monitoring and fault diagnosis from production controller
Knowledge-based controllers for CIM
Intelligent robotics
Supervisory control systems
PCB manufacture
Automated design to product

h) Management systems

Maximise investment returns
Investment planning
On-line information analysis on pricing, marketing and leasing
Unix system administrator
Salary calculations
Budget and resource calculations
Database access and information retrieval systems
Project management systems
Strategic planning systems
Personnel resource allocation
Requirements analysis

Make or Buy decision analysis

i) Monitoring

Computer System disk usage
Condition monitoring of underground trains and computer network management
Monitoring and control equipment in generators
Qualitative simulations
Market trend analysis
Manufacture monitoring and simulation
Signals monitoring and analysis

j) Interpretation

Ambiguity resolution system
signal emitter interpretation
Noise interpretation
Data analysis
Interpretation of Contract orders
interpretation of regulations

k) Estimating & Cost Control

Revenue enhancement
Bid-estimation system
Project cost estimation system
Research and development costing

l) Simulation

Remote diagnosis
Manufacturing load based simulation
Lead-time driven manufacturing requirements simulation
Modelling electrical power supply
Qualitative simulation
Simulate the effects of changes in sales or the market
FMC/FMC simulation

m) Prediction

Prediction of movement and corrosion
Prediction of item replacement
Predictive maintenance for mechanical equipment
Predicting operator behaviour in semi-automatic systems
Testability analysis

n) Training & Tuition

Database administrator's assistant
Computer-based training
Advice on legislation
Computer staff networking and operational support
Computer-aided software engineering

Advice to employees using administrative systems
Computer operations adviser

o) Stock Control

Slow moving spares / stock Advise
Stock requirements analysis
Stock location analysis
Distribution scheduling
Warehouse control
Stock control systems

p) Quality Control

Tolerance Advise
Taguchi statistical analysis for quality control monitoring
Testing procedures Advise for Finished Goods
Monitoring goods inward product quality and failures/supplier analysis
Quality control engineering for wire manufacture and other components
On-line quality assurance for manufacturing processes

q) Personnel

Recruitment adviser
Claims & Benefits adviser
Holiday scheduling system
Employment legislation adviser
Pensions adviser
Personnel appraisal adviser
Retirements adviser
Resource allocation system.

Appendix XII: Section E

Listing of Figures

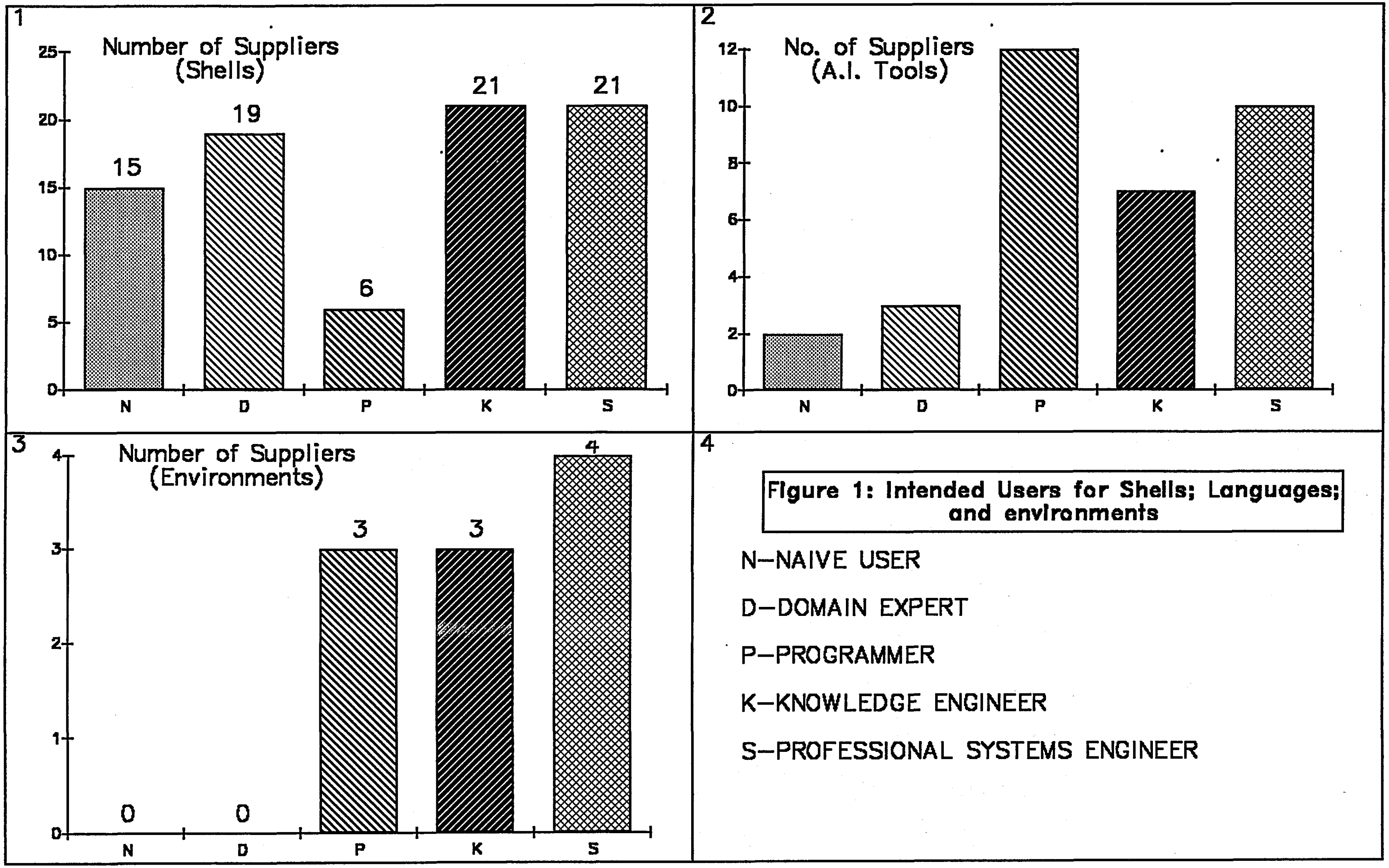


Figure 1: Intended Users for Shells; Languages; and environments

Figure 2: Frequency chart of service and support functions provided by Vendors

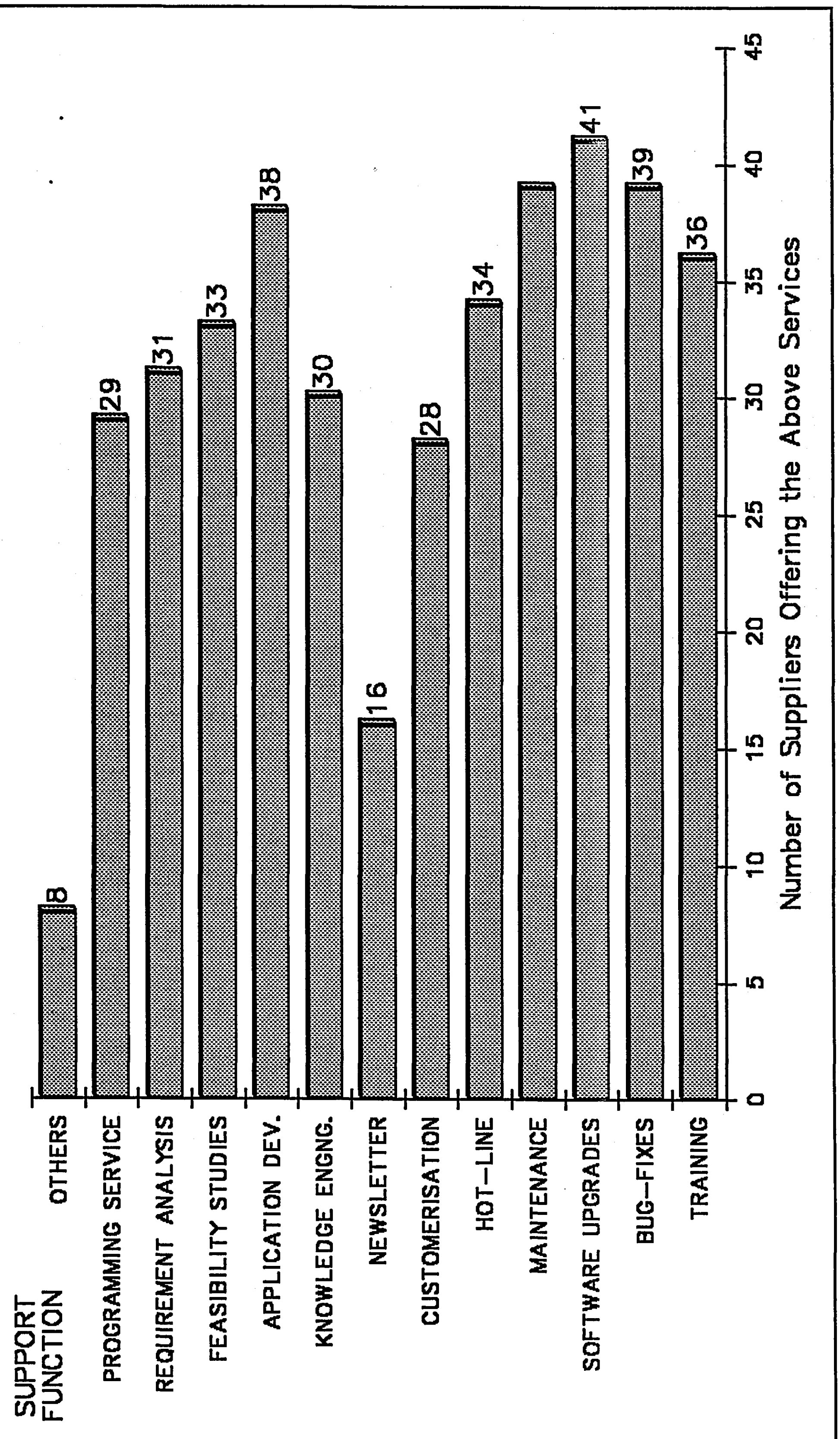


Figure 3: Frequency Chart of Interface & Explanation Facilities

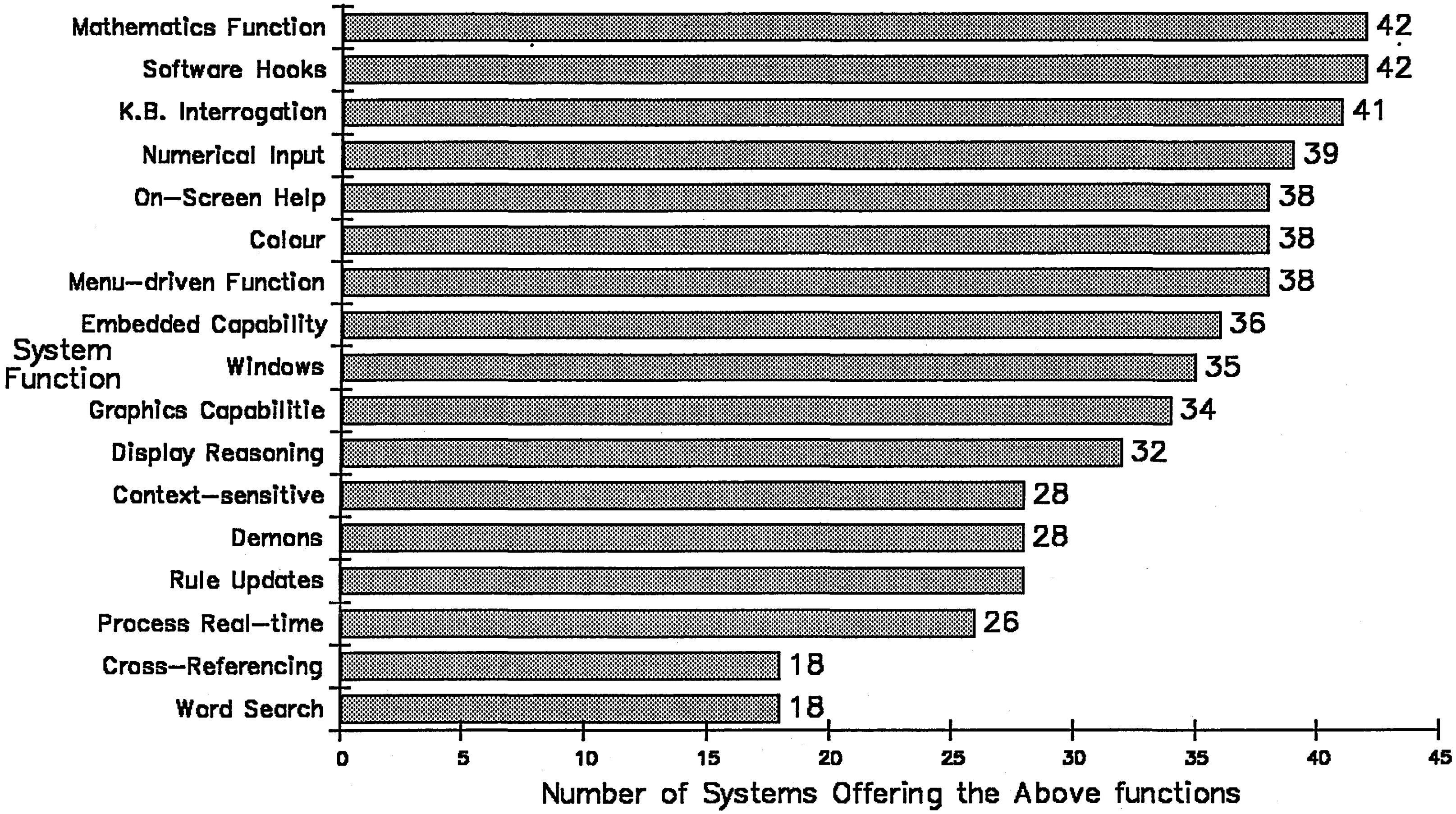


Figure 4: Profile of expert system shell applications according to generic group classifications

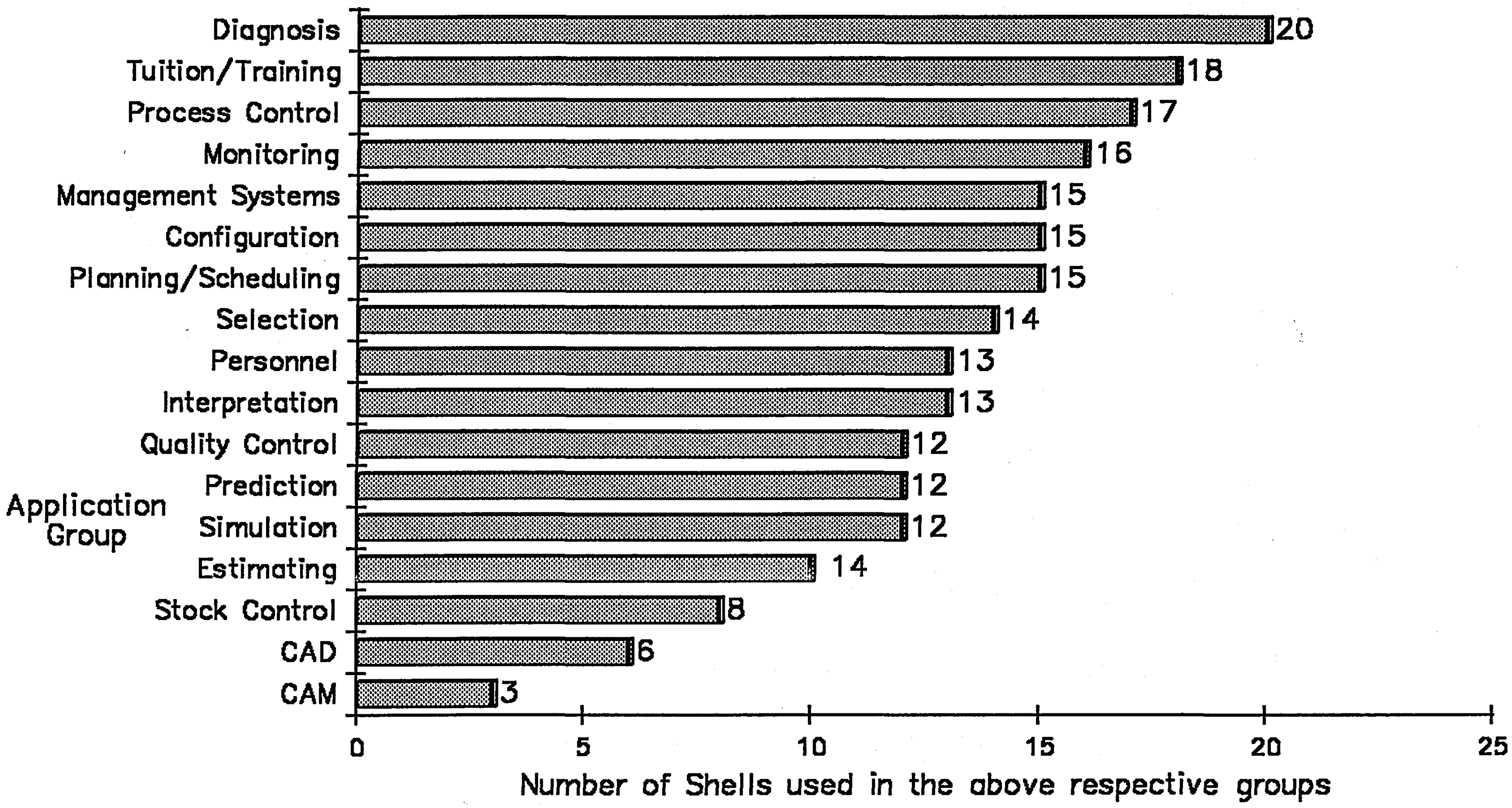


Figure 5:Profile of A.I. Tools & Languages according to application

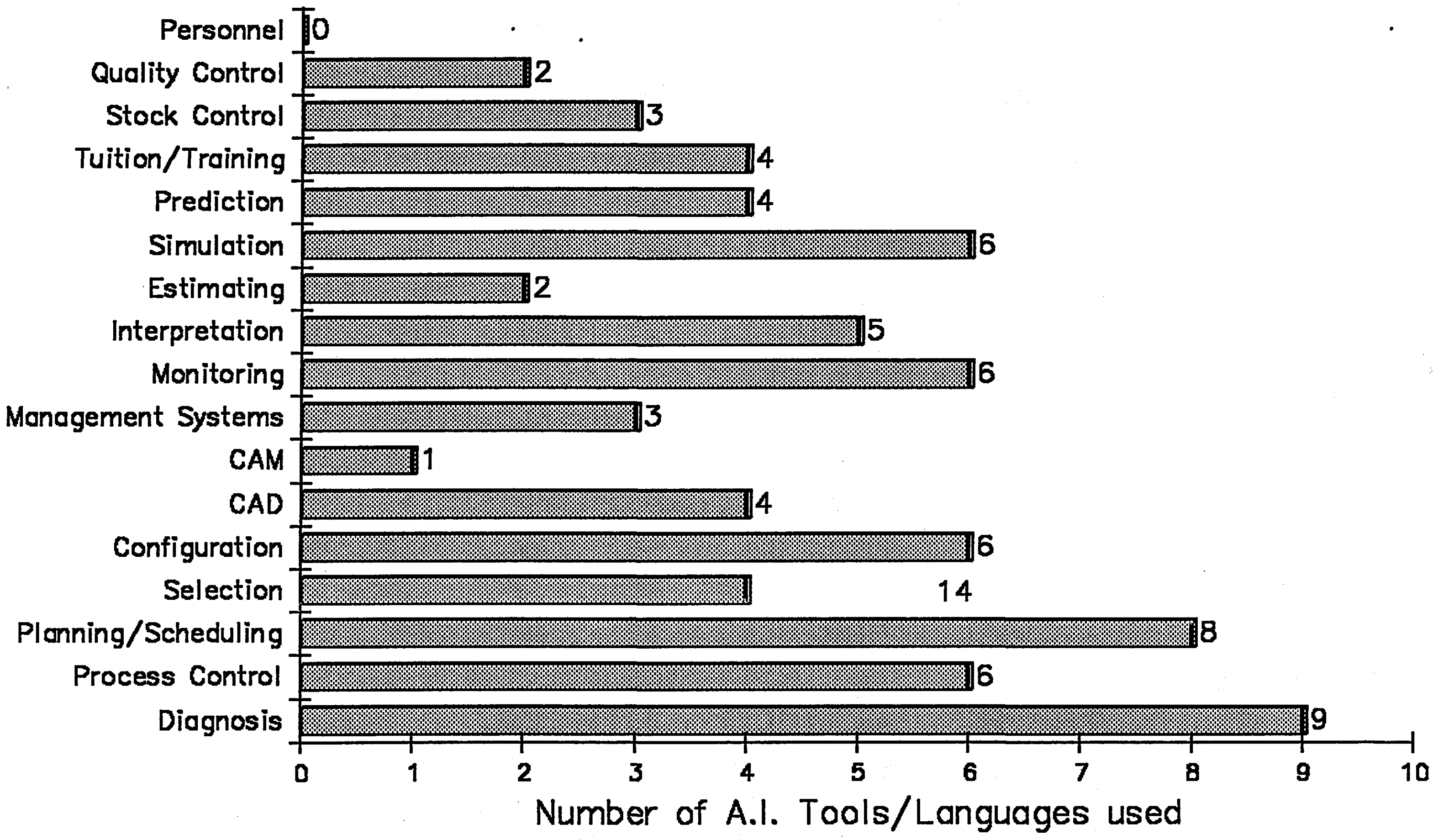


Figure 6: Profile of Environment systems according to application.

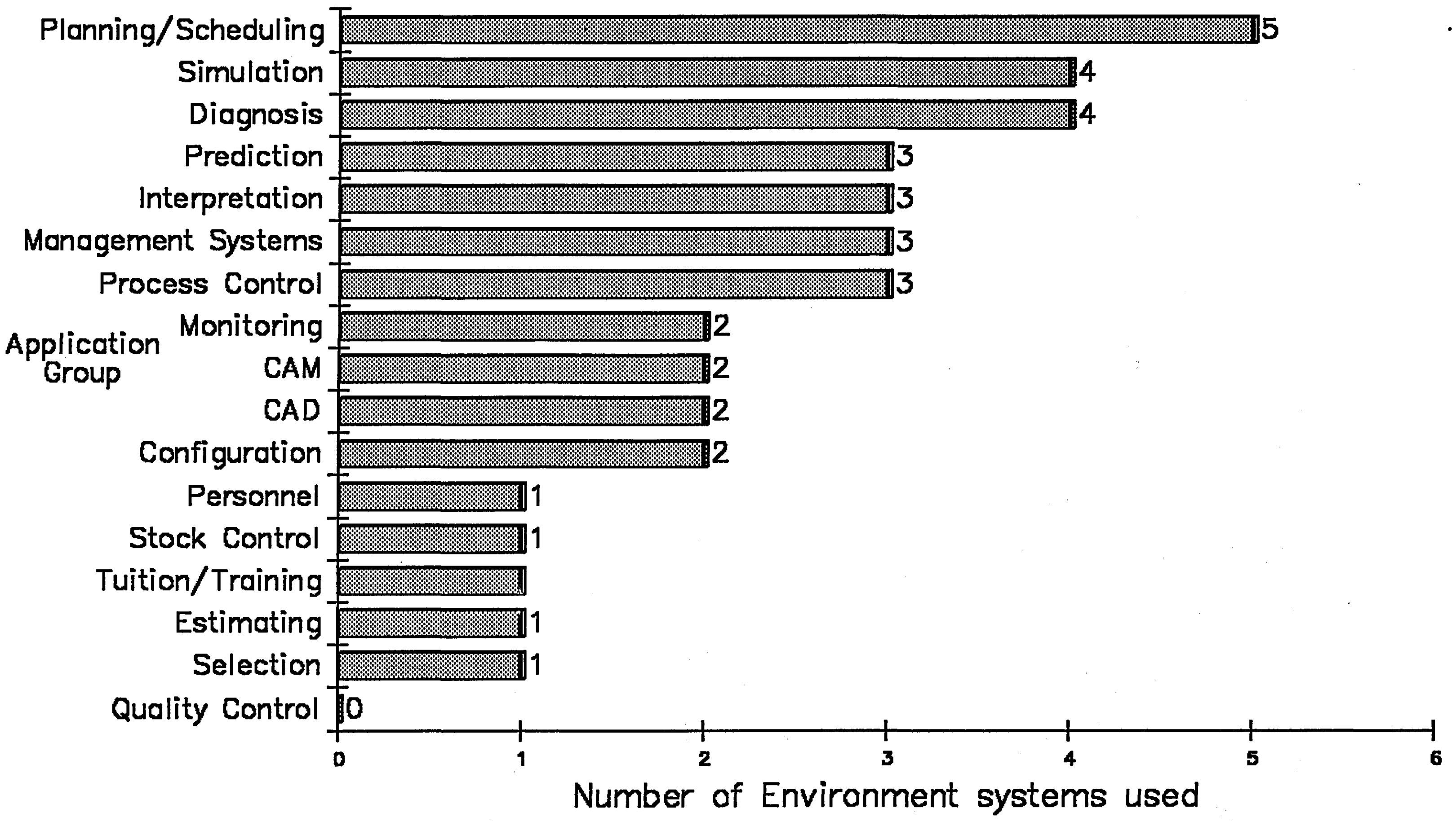


Figure 7: Knowledge representation Techniques used for Shells

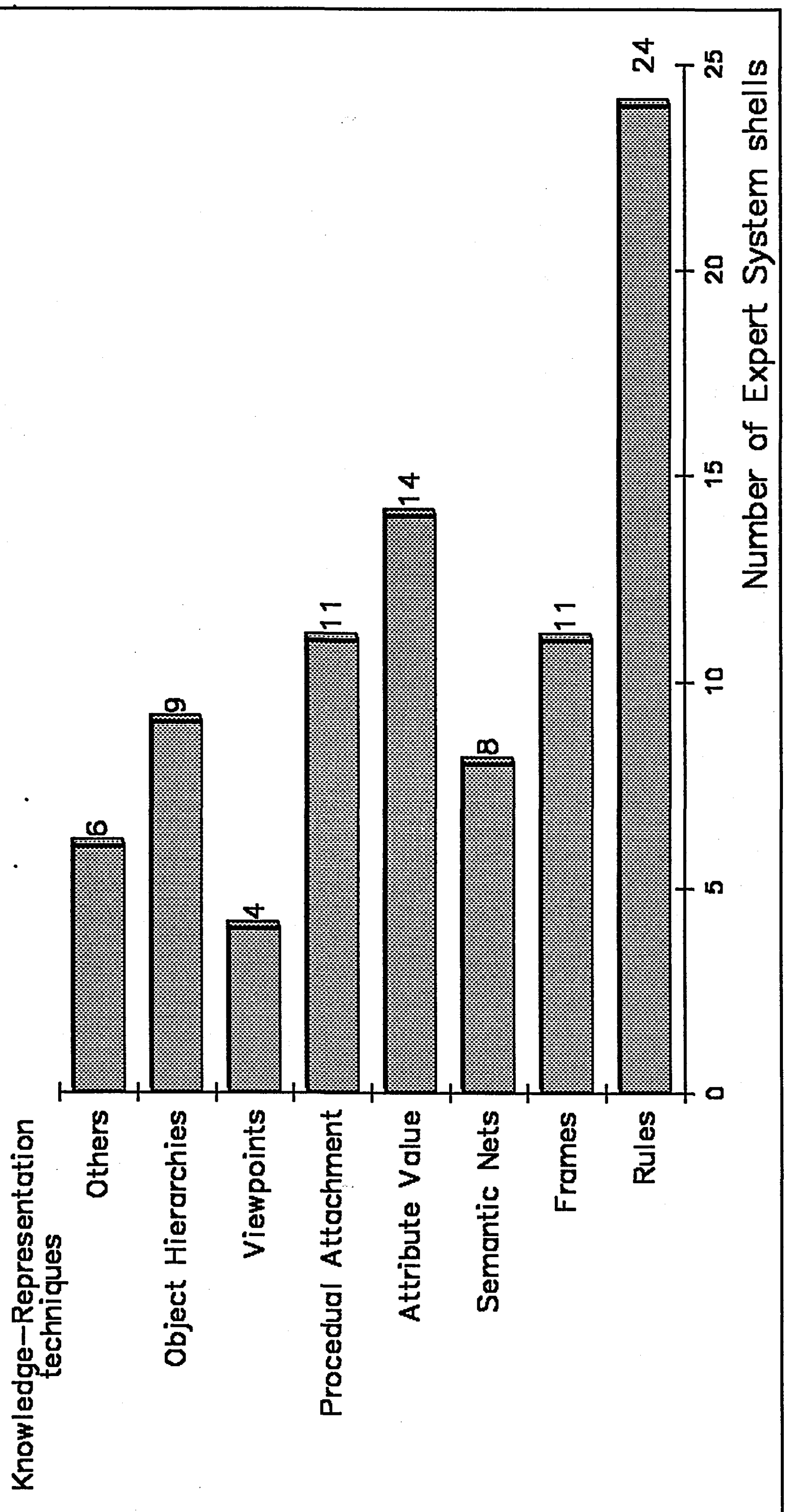


Figure 8: Knowledge-representation techniques used for A.I. Tools

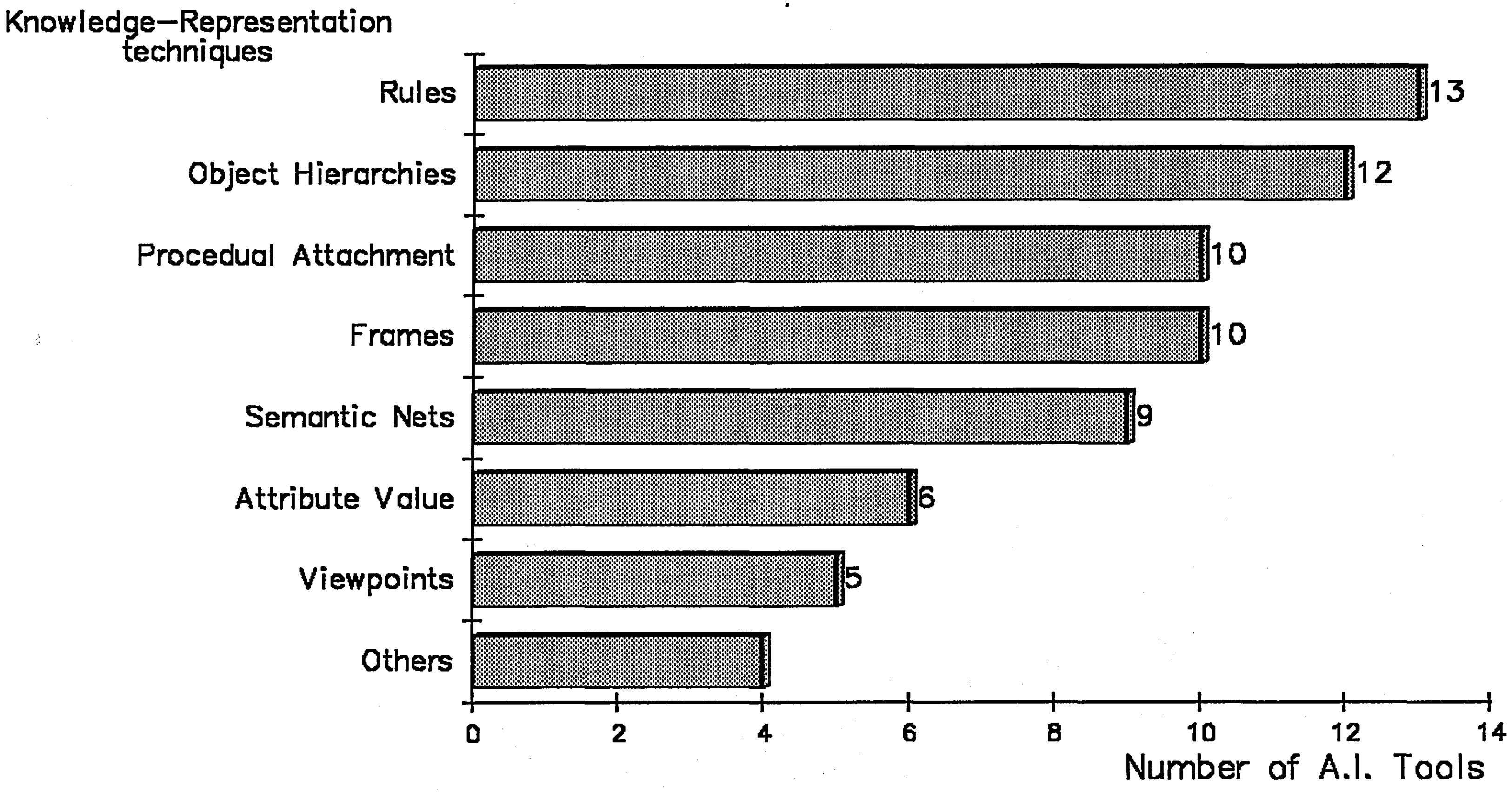
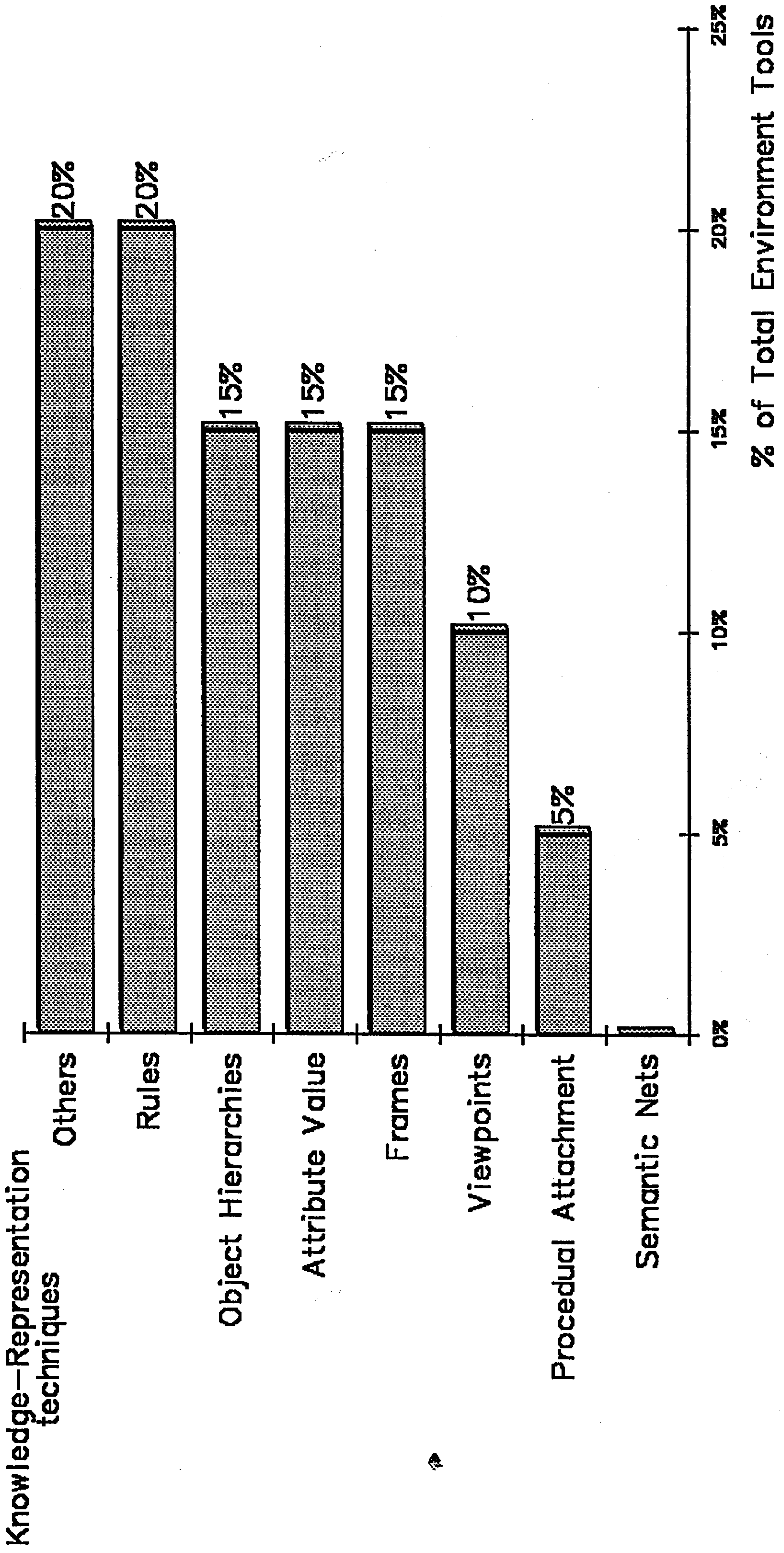


Figure 9: Knowledge-representation techniques for environments



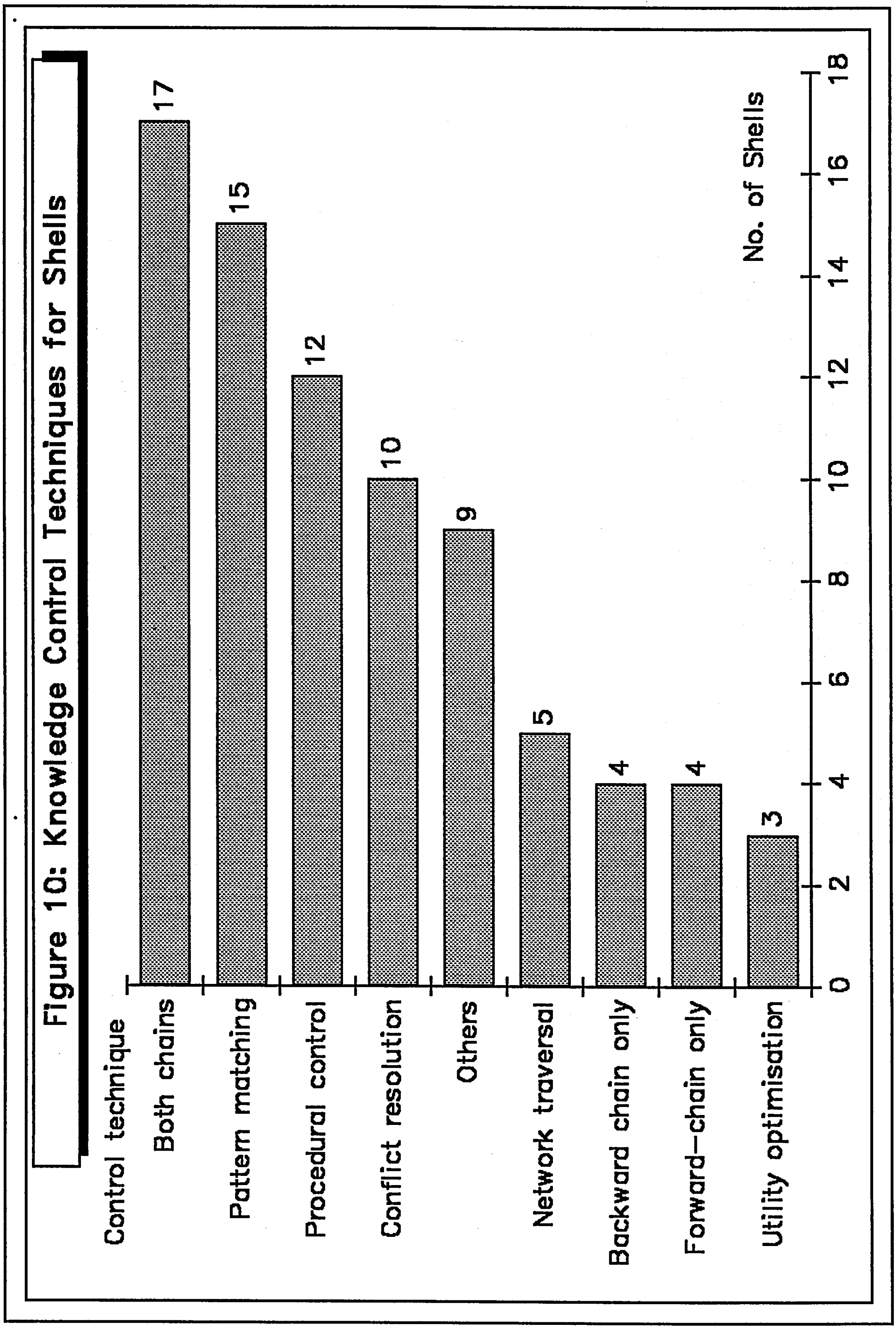


Figure 10: Knowledge Control Techniques for Shells

Figure 11: Knowledge Control Techniques for A.I. Tools

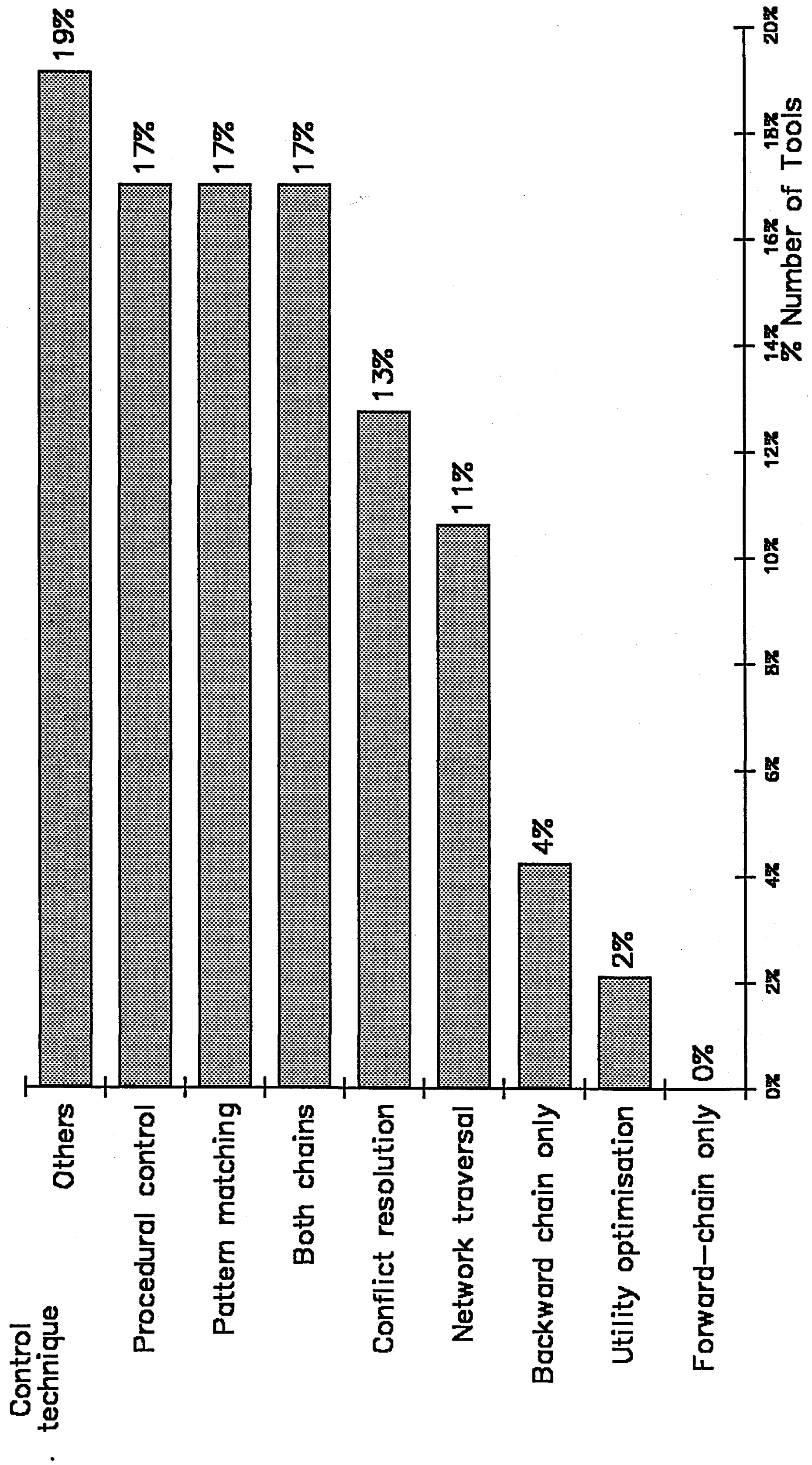
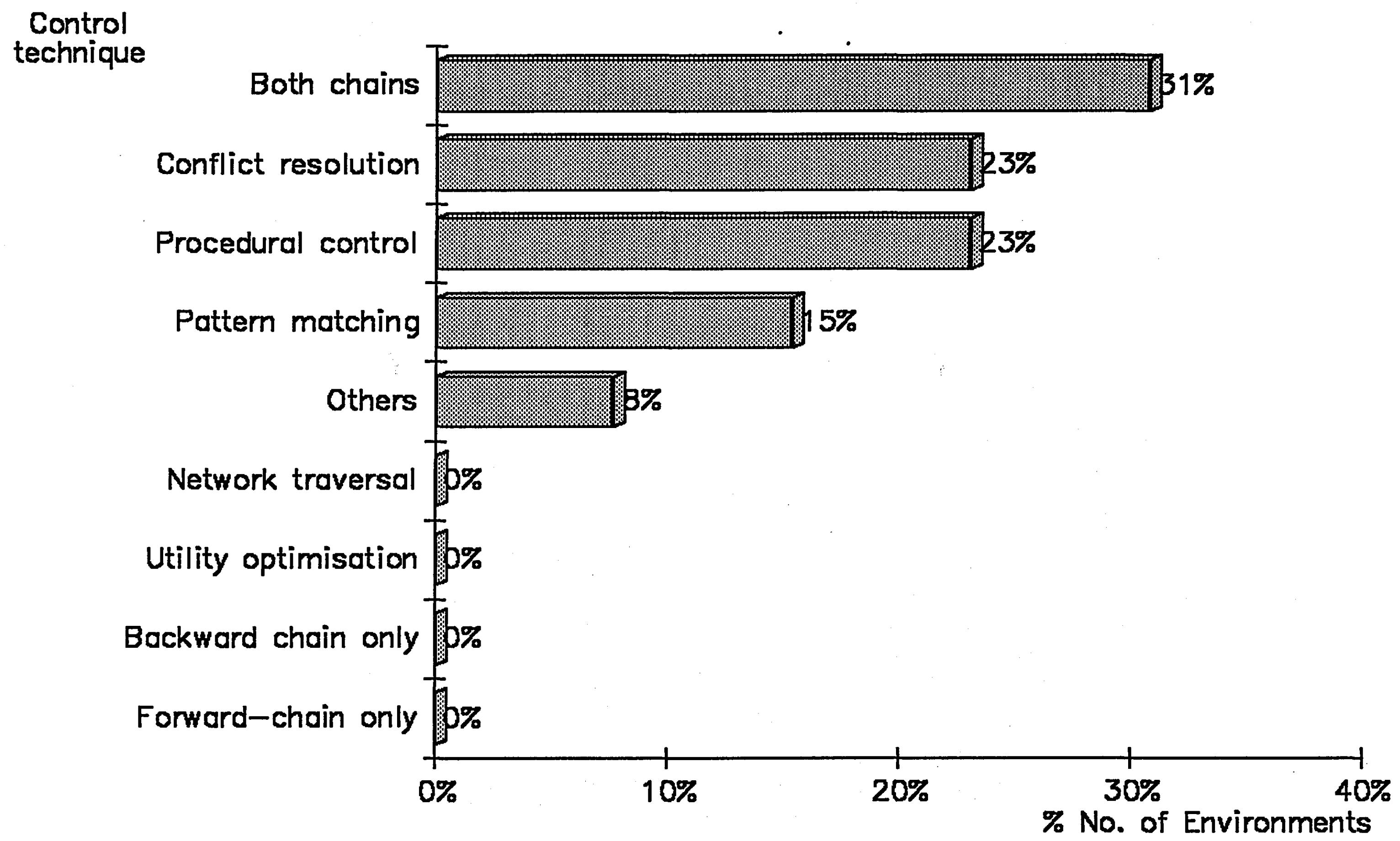
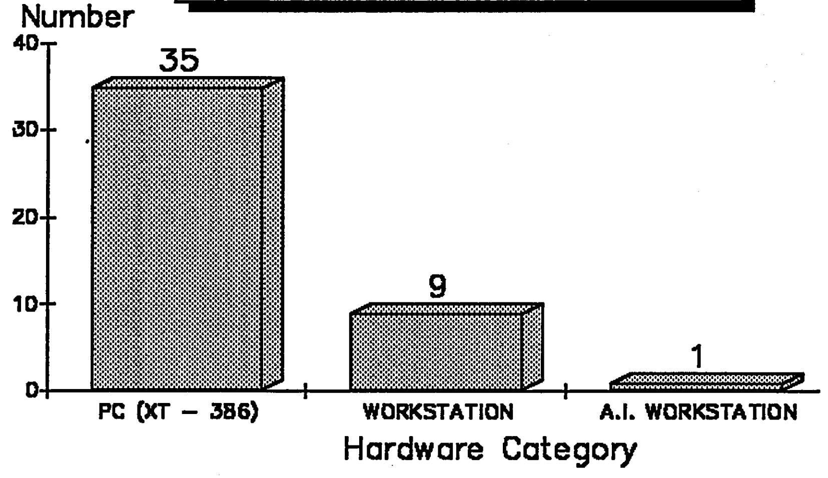


Figure 12: Knowledge Control Techniques for Environments



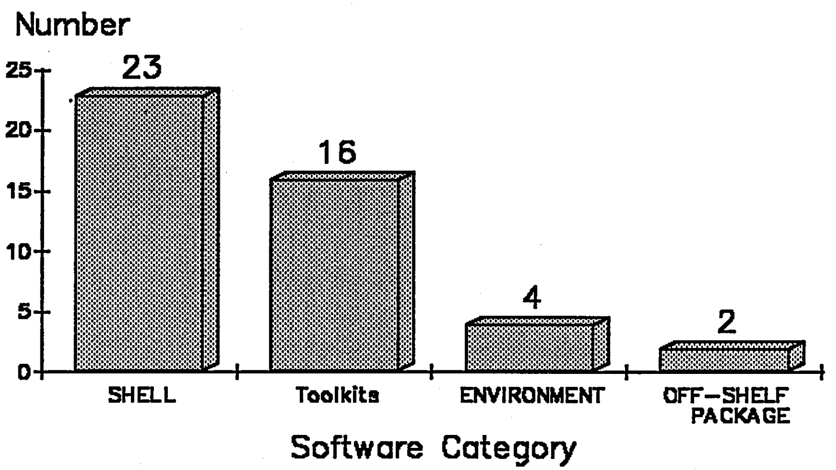
1

Figure 13a: Hardware Requirements



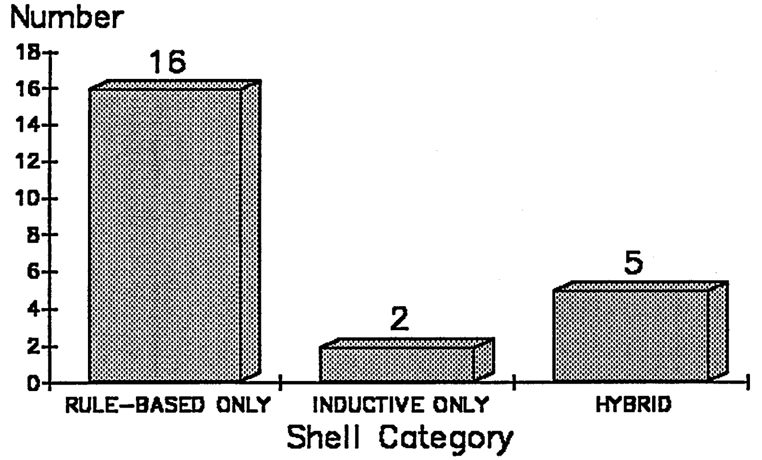
2

Figure 13b: Minimum Software Requirements



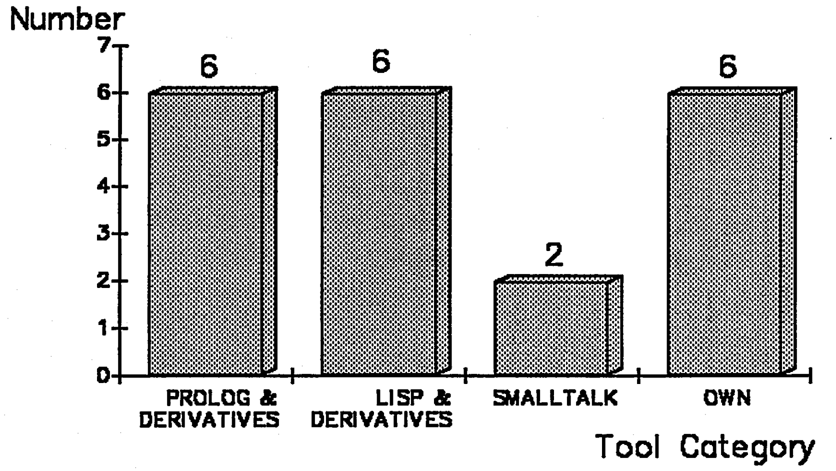
3

Figure 13c: Shell Types



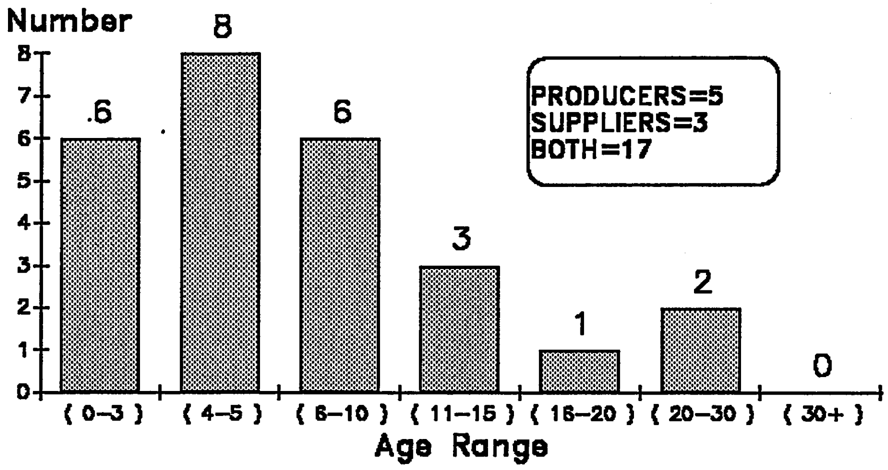
4

Figure 13d: Tool & Environment Languages



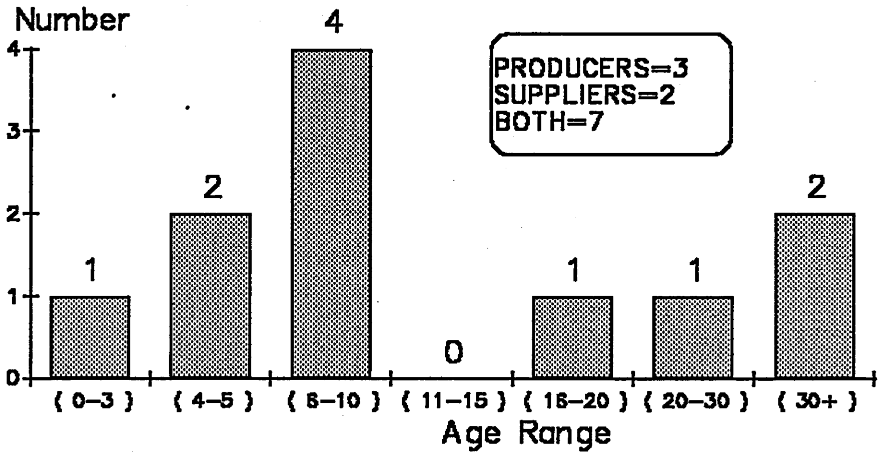
1

Figure 14a:Age of Companies Supplying Shells



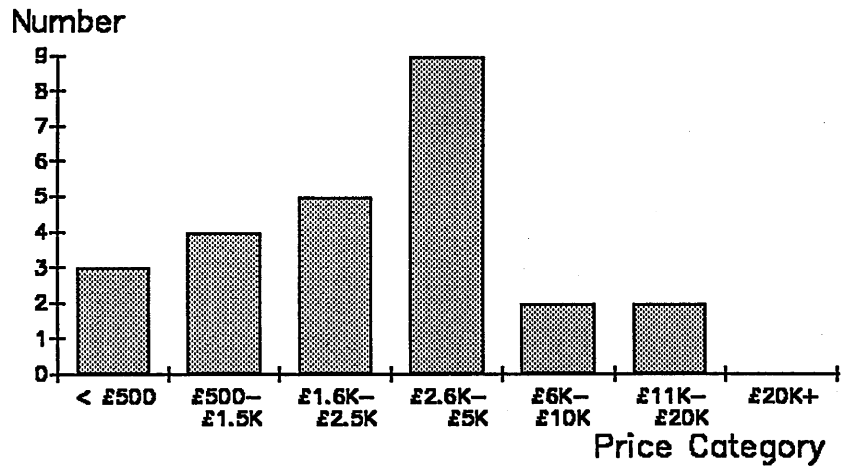
2

Figure 14a: Age of Companies Supplying A.I. Tools



3

Figure 14c: Price Range of Shells



4

Figure 14d: Price Range for Tools & Environments

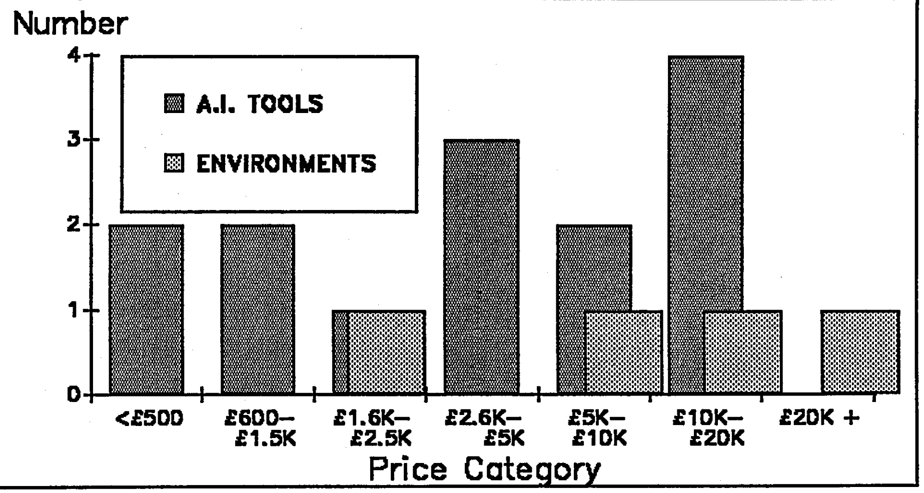


Figure 15: Handling Uncertainty in Shells, Tools & Environments

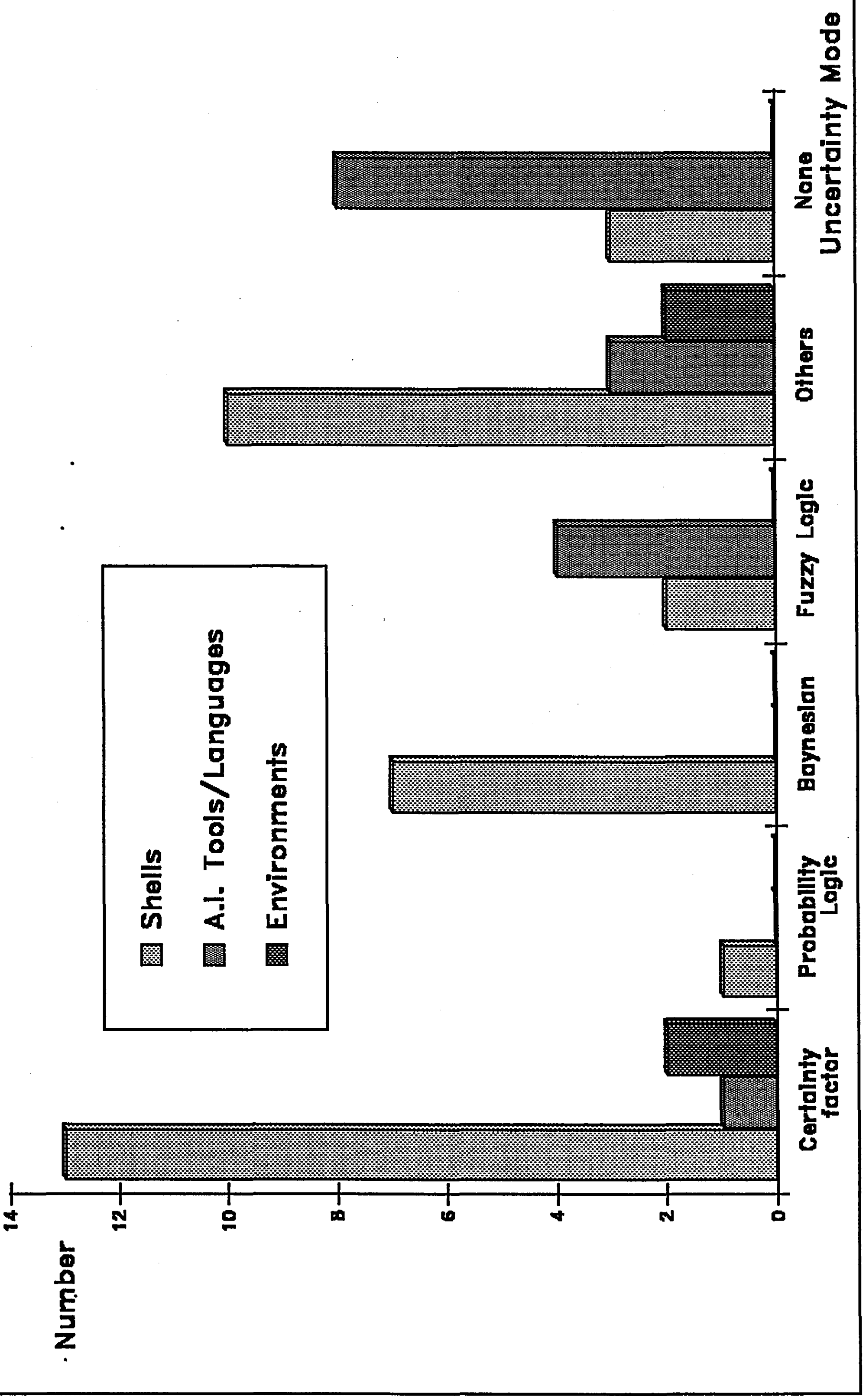


Figure 16: The Use of Shells, A.I. Tools & Environments in Manufacturing

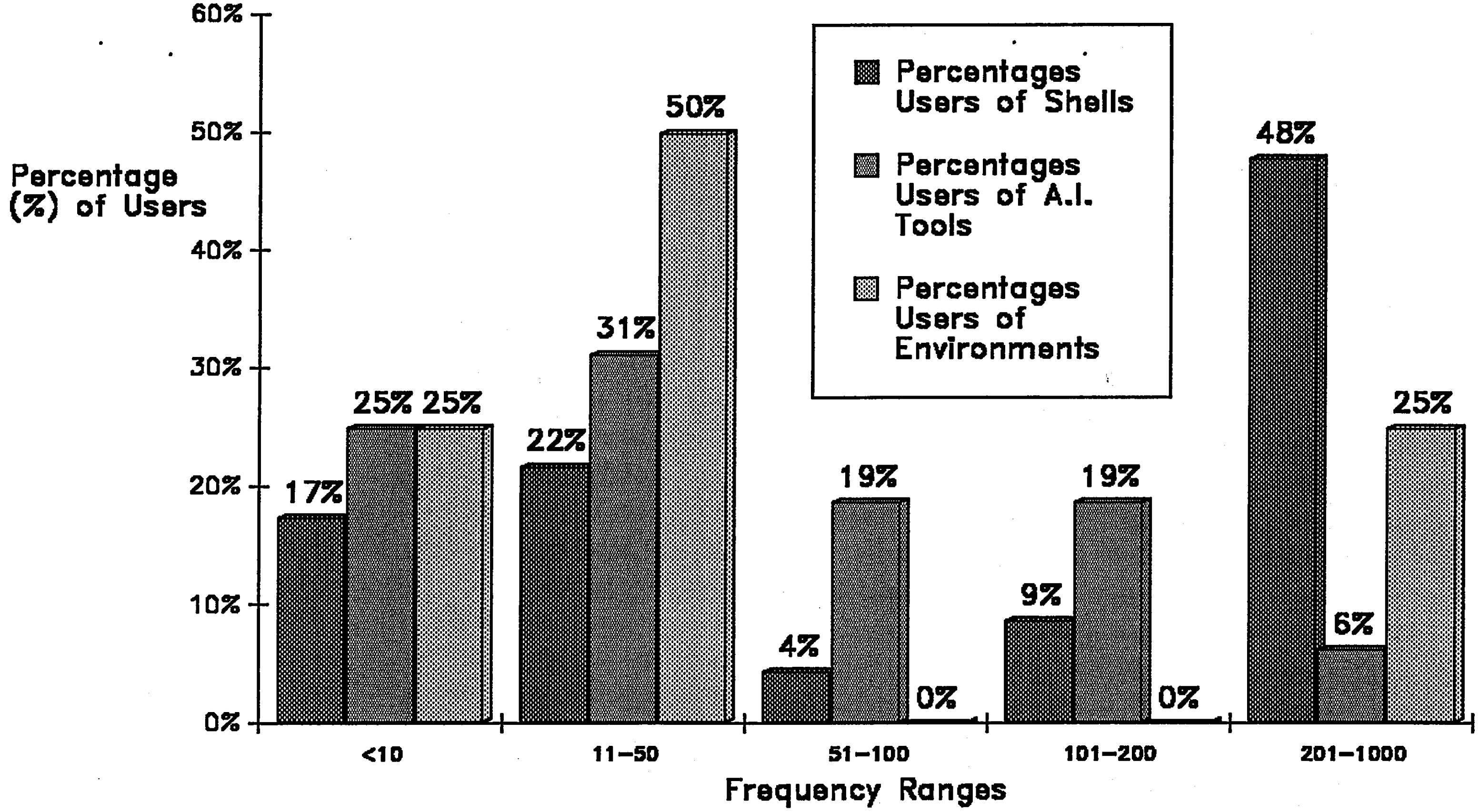
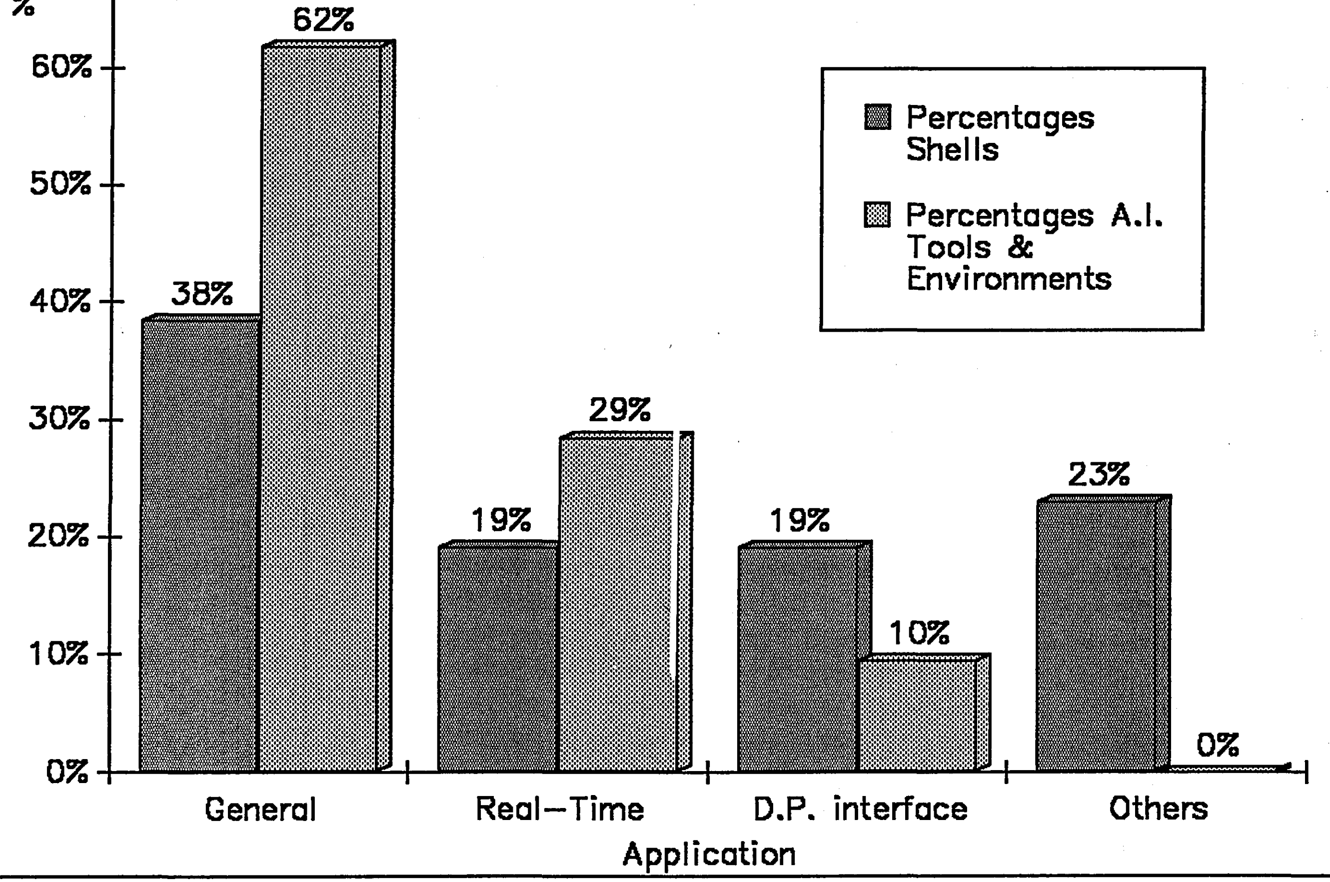


Figure 17: Vendor's Intended Role for Expert Systems



Appendix XII: Section F

Listing of Participating Vendor Organisations and Products

Company Name	Product(s)
1. Artificial Intelligence Limited	ENVOS Goldworks Quintus Prolog Smalltalk 80
2. A.I.I.T. Limited	Genesia I, II, RT Instant Expert
3. Attar Software Limited	XpertRule Advanced X-Rule
4. Automated Reasoning Limited	I-CAT
5. A.I. Corp Limited	KBMS
6. Biosoft	Expert 4
7. B.I.M. Limited	BIM-Prolog
8. Borland UK Limited	Turbo Prolog
9. Bechtel Limited	Lisp
10. Cambridge Consultants Limmited	MUSE
11. Carnegie (UK) Limited	Knowledge Craft
12. Chemical Design	Advisor-2 Prolog-2
13. Cognitive Applications Limited	Alphapop
14. Creative Logic Limited	Leonardo 1 Leonardo 2 Leonardo 3
15. Cullinet Software Limited	Application Expert Enterprise Expert
16. Cognosys Limited	Common Knowledge
17. Concurrent Computer Limited	LISP/Prolog
18. Database Expert Limited	Knowledgeman
19. Digital Equipment Corp(UK)	Lisp/Prolog
20. Electronic Facilities Design Limited	Prolog Design
21. Erros Computer Services Limited	ERROS
22. Expertech Limited	Egeria
23. Export Software International	Expert Ease
24. ETCS	Prolog Xi Plus
25. Fril Systems Limited	FRIL
26. General Research Corp(UK)	Lisp
27. GreyMatters Limited	Prolog
28. Harlequin Limited	Lispworks
29. Helix Technology	Expert edge
30. Hewlett Packard	HP Common Lisp HP Prolog
31. I.E. City	Chrystal City
32. IBM UK Limited	ESE KEE Knowledge Tool

33. ICL	ICL Advisor
	REVEAL
34. Information Builders UK Limited	LEVEL5
35. Integral Solutions Limited	Poplog
	SD-Advisor
	Rules
	Keris
36. Intelligent Systems International	Savior
	Egeria
37. Intellicorp(UK) Limited	KEE
38. Intelligent Applications	Annie
	Synergist
	Violet
39. Intelligent Environments Limited	Crystal
40. Knowledge Garden (UK) Limited	KnowledgeMaker
	Knowledge Pro
41. Logic Programming Associates	Apes
	Flex
	LPA Prolog
42. McGraw-Hill (UK) Limited	Master Expert
43. Machine Reasoning Limited	Orion
44. Megatron Computers	Prolog
45. MDBS (UK) Limited	GURU
46. National Engineering Laboratory	Stimulus
47. Neuron Data	Nexpert Object
48. Nixdorf Computers Limited	Prolisp
49. PAL Software Limited	ES-DEveloper
	Intelligence-1
50. Power Computing Limited	ICAD
51. Procyon Research Limited	Common Lisp
52. Quintec Systems Limited	Quintec-Prolog
53. Scientific Computers Limited	G-Base
54. SD-Scicon	SD-Prolog
55. Software A & E	KES
56. SIRA Limited	G2
	SISS
57. Software Sciences Limited	NEXPERT
	A.S.A.P
58. Software Generation International	AION
59. Southdata Limited	SUPERFILES-ACLS
60. Signal Computing Limited	SIRTES
61. Sun UK Limited	SCLISP
62. Symbolics Limited	Joshua
63. System Sciences Limited	Prolog
64. SW Publishers Limited	Turbo Prolog
65. Telecomputing Plc	Flex
	Top-One
66. Texas Instruments	Personal Consultant
67. Total Systems Plc	EXSYS
68. Unisys UK Limited	KEE
69. Vanilla Flavor Company	Arity
70. Warm Boot Limited	PC/BEAGLE
71. Wisdom Systems (McDermott)	Concept Modeller
72. Work Sciences Limited	PRIORITIES

Appendix XIII

Programming in 'Crystal': The Expert Systems Shell

Contents:-

Appendix XIII: Section A.

An Introduction to Crystal *

This section summarises the main features and programming techniques of Crystal, whilst the next section provides examples of their use in the help-desk.

1. Propositional Logic Structure

Crystal is a rule-based shell and rules are written as propositional logic, that is, the combine a series of statements using connecting words such as AND, OR, NOT and IF. For example,

```
            <CONCLUSION>
IF          <CONDITION 1.1>
AND         <CONDITION 1.2.>
............
AND NOT     <CONDITION 1.n>
OR          <CONDITION 2.1>
AND         <CONDITION 2.2>      etc...
```

The conclusion defines the name of the rule and is set true or false depending on the conditions. Any rule may be broken down into smaller bundles of rules and conditions with each rule having its own subsequent conclusion.

2. Searching and Control Strategy

Crystal is essentially a backward chaining shell (although it can handle forward chaining) in that the line of reasoning is driven backwards from the conclusion or goal stated in the Crystal Master Rule using the rules in an attempt to prove that goal. Crystal uses a depth first procedural method of control whereby each rule calls the rules it needs to test to fulfil its own conditions explicitly by name. The rule and command conditions are executed sequentially until an 'AND' clause fails. At this point, Crystal searches for an alternative ignoring the remaining 'AND' clauses in that set. Alternative conditions preceded by 'OR' are only evaluated when previous alternatives have failed. When the whole rule has succeeded or failed, control is passed back to the rule above.

3. Development Functions

i) Build Function: Rules are entered into the knowledge base through a series of screens. On entering the build system a sub-menu appears with options of rules, variables or unused rules. The rule option is selected in order to create or edit a knowledge base, the variable option is used to add or amend variable validations and unused rules can be amended or deleted via the unused rules option.

---

* *Extracts taken from the Crystal Manual (version 3.1) and MI (1989)*

ii) Master Rule: The master Rule Screen provides a skeletal structure for inputting rules. The keywords of IF, AND and ORD are automatically generated by the Crystal editor depending on the position of the cursor on the screen . The expertise is typed in as a set of test conditions. The rules are incrementally compiled at build time and are input as a hierarchy of rules and variables. Rules at a lower level (rule condition) return Boolean values of true or false as a result of testing their own conditions. At the lowest level the conditions are made up of pre-programmed knowledge base commands. These are held in a sub-menu which may be accessed when a condition is being entered.

iii) Rule Commands: Many commands cause some action to be taken such as to display information to the user, accept user input from the user, assign variables, restart the system, provide explanation or transfer data between Crystal and external commands. Some commands cause the logical result of the knowledge base to be affected. These include: -

Fail:	This function causes the current line of reasoning to fail, and forces any alternative condition to be tested
Succeed:	This function allows the current line of reasoning to continue even if some conditions fail
Test	This is the only command for testing the value of numbers and strings and will succeed or fail depending on the value of the test.
Program:	User program commands allow communication with an interface program and will succeed or fail depending on the actual interface program.
Yes/No:	Yes/No questions present the user with a string of text requiring a yes/no answer. The condition succeeds if the answer is yes and fails when no.

iv) Dictionaries

Two dictionaries are automatically maintained. These are the rule and variable dictionaries (these provided a useful listing of rules and variables for the help-desk, as Appendix 14 shows). The rule dictionary is available at the point where a condition is entered into a rule. Various editing facilities are provided such as copying, altering to create a similar rule and storing in the dictionary under a new name, swapping conditions or alternatives within a rule, and adding intermediate conditions. In this way, rules and their associated structures can be copied to other parts of the knowledge base. Some debugging aids are also available from within the rule dictionary such as expanding the current rule line into its rule conditions, and obtaining a menu of rules from where a particular rule is called up (these are called breakpoints). A list of unused rules is also maintained.

The variable dictionary contains a list of all the variables and arrays used within the knowledge base and is available from within an expression where a variable is to be used, for example inside the test command; or from the top-level Build Menu where validation conditions for each variable or array can be entered and displayed.

4. Help Facilities

Relative to other shells, Crystal has good help and explanation facilities (MI:1989). The developer can get menus of available functions and commands at the press of a key, and syntax is checked as knowledge is coded into the knowledge base. This is achieved by making extensive use of pre-programmed function keys. The meaning of each function key is alters depending on which part of the Crystal system the developer is in at the time. For example, within the knowledge base editor the F6 key provides a list of screen operations, and from the ruletrace facility the text of the current line is displayed.

The Rule Trace facility is the only debugging tool provided within Crystal. It can be activated or de-activated at any point during the running of the knowledge base. Using the Rule Trace, the developer is able to expand conditions at any point in the knowledge base and also run parts of the knowledge base in isolation to see if they succeed or fail and thereby localise the source of bugs.

5. End-User Interfaces

Crystal has easily understood interfaces which is one reason for its use as a front-end to databases and spreadsheets. Input and output is achieved using a combination of yes/no questions, forms, graphics, text screens and menus. Various types of forms are provided within Crystal which have the same operations performed on them using the Screen Painter, but all have slightly different uses. Menus for instance are created using the Menu Question form: these may be text menus, sliding bar menus (see Appendix 14 for an example), or multiple value menus. In addition to menu fields, there are also output fields (these were used to produce data records of user and system profiles). View Forms output variables only and continue whilst Print Forms generate a printed report of output variables (this facility was used to pass consultation information from the operator/helpdesk on to the expert). These forms can be used to create pop-up windows, help boxes, and data entry forms all of which made using the help-desk easier.

6. External Interfaces

Crystal supplies pre-defined interfaces to the DOS operating system and files having Lotus 123, dBaseIII or ASCII formats. The interfaces can be used individually or in combination and are supplied as a series of executable 'C' language programs.

7. Evaluation of Crystal

The strengths of Crystal are that it provides many of the facilities associated with much more expensive shells and toolkits. It has a proven record in diagnostic and trouble-shooting domains where there is decision-tree type logic and therefore is highly suited to the help-desk domain. It was found to be straightforward to use, assisted by the fact that it was menu-driven and had excellent documentation and tutorial support, and had a very good telephone support service which was frequently called upon.

The weaknesses of Crystal are that it is inflexible- it is limited in scope to production rules and principally a backward chaining schema. This was adequate for the help-desk, but other applications may find Crystal limiting. In comparison with other shells, Crystal does not provide 'why' and 'how' or 'what-if' facilities directly although programs may be written to accomplish these tasks.

Appendix 13: Section B

Help-Desk Program Features *

This section provides programming examples from key features in the help-desk. The examples concentrate upon those features which are not standard in the Crystal software or have a special use in the help-desk. Screen prints are used to show how the help-desk presents information to the operator : they can only provide a 'flavour' of the format of consultations .

1. System Profiles

In using the help-desk, it was necessary that the operator had access to information about the user and the systems end-used by the end-user in order to customise the nature of the consultation and thereby avoid laborious menu-driven information-gathering sequences. It also ensured that the operator was sympathetic to the level of competence of the user (although the programming difficulties associated with defining separate levels of consultation dictated that the final form of the help-desk would only adopt system profiles).

Having selected the system profile option (Screen A in Figure B1), the operator is faced with three options as Screen B shows:-

i) Review a Known Record

Here the end-user provides the serial processor number of the computer system being used (variable pcuser$); this loads up the profile pcuser$.pc which is displayed in a format similar to Screen E. The program to execute this sequence is as follows:-

```
IF          TEST ERROFF()                                          EX. SP.
AND         ASSIGN PCUSER$=""
AND         DISPLAY FORM
AND         TEST IMPORT(PCUSER$+".PC")
AND         HELP EXPLAIN
AND         DISPLAY FORM                                     (SEE SCREEN E)
AND         **PRINT_OUT_PROFILE ?
            **      IF          YES/NO QUESTION
                    AND         TEST (OUTPUT "LPT1")
                    AND         PRINT FORM
                    AND         SUCCEED
AND         END HELP
AND         **RETURN TO REST OF CONSULTATION
AND         TEST ERRON()
```

The purpose of the TEST ERROFF()/ERRON() command is to ensure that if programming logic faults occur then they are logged for the developer rather than presented to the operator which may be confusing. The 'Ex.' function denotes that the rule is an export rule and may therefore be stored to a file called pcuser$.pc. The Sp. command ensures that it is a special rule and therefore new records may be called up by wiping the old name of pcuser$.pc and associated record values. The asterisks imply that the rule has a sub-rule or sub-rules as for example in the

---

* *This section refers primarily to the features included in the revised Office Systems Helpdesk.*

command ** print_out_file. The Help Explain/End Help sequence makes it possible to provide context sensitive help facilities by relating guide-lines to specific rules.

ii) <u>Search for the End User Serial Number</u>

In the event that the end-user fails to provide the correct serial number after the third attempt or the operator wishes to view a record independently, a file listing of serial numbers is useful. In the example shown in Figure B1, the operator requests a profile of a PC user (Screen C); all files from this category are listed in Screen D. the program which enables this is as follows:-

```
IF          TEST ERROFF()                                   1
AND         TEST DIR(DECEX$[#], "*.PC",0)                   2
AND         TEST SORT (DECEX$[#],0)                         3
AND         MENU DECEX$                                     4
AND         TEST IMPORT(FILE DECEX$+ ".PC")                 5
AND         DISPLAY FORM                                    6
AND         TEST ERRON()                                    7
AND         SUCCEED
```

Line 2 above locates all files ending with the extension *.pc and places then in a single dimension array. Line 3 then sorts the file numbers in ascending order for ease of location. All the records are displayed to the operator as menu options in Line 4 from which a selection, DECEX$ is made. Line 5 then calls up this file from records and displays it as in Screen E using the Display Form command in Line 6.

Upon selecting a file for import, a screen card is called up (for example Screen E) which provides factual information about the user and location of the user; a configuration of the computer system and associated peripherals; and an inventory of the software used including a support rating (a rating of 1 indicates that the computer department fully supports the software, while a rating of 3 implies that the software is specialised and for use by the end-user alone in which case there is no company wide support). It is useful to provide a print-out of a system when a fault is escalated to the expert and therefore a print-screen facility is provided (Screen F) as a YES/NO question.

iii) <u>Input a New Record</u>

Information may be updated by amending the records in screen cards or defining new records.In this case, a Display Form is used to define input fields as Screen G shows. Programming begins by initialising text and numeric variables from which the operator adds records to the screen card and saves it under a new filename:

```
IF          INITIALISE VARIABLES                            1
                    ASSIGN PCUSER$=""                      1A
                    ASSIGN EXT:= 0                         1B
                    ASSIGN ALT$=""                         1C
                    ASSIGN SP1:=0 ETC....               .....1N
AND         DISPLAY FORM        (I.E. SCREEN G)             2
AND         DISPLAY FORM                                    3
AND         TEST EXPORT(PCUSER$ +".PC")                     4
AND         YES/NO QUESTION                                 5
            AND RESTART RULE                                6

OR          SUCCEED                                         7
```

Lines 4-6 allow the operator to input further records if desirable within the same consultation. These records are screen cards rather than databases in the sense that they cannot be interrogated (for example, list all users with a PC sharing adaptor etc.). A useful enhancement therefore would be to link Crystal to Dbase or use Crystal's own ASCII based data handling facility.

2. Prioritising

A requirement of the help-desk was that it should be able to distinguish between different levels of users and users' queries. Both required a rating system which would channel high priority (critical) faults directly to the expert as a fault report and, in the case of user profiles, indicate the level of expertise held by the user.In both cases, the expert had difficulty in quantifying levels and so a qualitative rating system was used. An example of this is shown in Figure B2 for defining the criticality of a device in terms of high medium or low importance across a sliding scale (Screen A). This information is passed onto the expert from the operator who then decides in this case whether outside specialist repair help is required. The score takes the form of an array (screen B) made up of a score rating (which is provided by the expert at an earlier stage using screen D - in this case the expert preferred a numeric value) and a bar value: both variables are displayed in Screen C. This information is processed as records and therefore is defined as an export rule and stored in ASCII format.

3. Saving Consultations

At the end of each consultation, the operator is given the option to save the consultation or refer back up to higher level knowledge bases without saving (see Screen A of Figure B3). If a consultation is saved the operator is asked to provide a filename (Screen B) before being given the option to transfer to another knowledge base or quit the system (Screen C). The filename is usually in the format of the end-user's initials and date. The program command is as follows:-

```
IF          MENU SAVEPQ         (i.e. Screen A)                      1
AND         TEST SAVEPQ=1                                            2
AND         ASSIGN ZAP$=""                                           3
AND         DISPLAY FORM        (i.e. Screen B)                      4
AND         TEST EXPORT(ZAP$+'.IV5')                                 5
AND         **CHANGE LEVEL                                           6
                IF          MENU WHEREPQ        (Screen C)          6A
                AND         TES WHEREPQ=1                           6B
                AND         KBS RE-RUN

                OR          TEST WHEREPQ=2                          6C
                AND         VIEW FORM                                .
                AND         TEST LOAD("GENSUB2)                      .
                .                                                    .
                .                                          (continued)
                OR          TEST WHEREPQ=5                          6F
                AND         VIEW FORM
                AND         TEST LOAD("MAIN")

                OR          QUIT                                    6G

OR          **CHANGE LEVEL                                           6
```

The file extension used in Line 5, .IV5, indicates which knowledge base the consultation ended up at. The TEST LOAD() function calls up a knowledge base and since this may take some time to load up, informs the operator using a VIEW FORM DISPLAY that knowledge-bases are being transferred.

4. Recalling Consultations

An operator may wish to recall a consultation at a later date, for example in the case where the end-user is cut-off the phone and contacts the operator at a later time. Files are displayed as a menu option and are listed in sequence according to the time and date. The operator selects a file and specifies the file extension (e.g. IV5) which determines which knowledge bases should be loaded first. The program which delivers this operation is below:

```
IF              TEST ERROFF()
                AND **INITIALISE VARIABLES

                     IF        TEST ARRCLR(FILEM$[##]
                     AND       ASSIGN EX$[0] = ""
                     AND       ASSIGN FILEM$ = ""
AND             TEST DIR(EX$[#],"*.IV*",1)
AND             TEST SORT(EX$[#],1)
AND             ASSIGN FE$ = ""
AND             DISPLAY FORM
AND             **IDENTIFY MODULE

                     IF        TESTFE$ = ".IV1"
                     AND       TEST IMPORT(NAM$ + ".IV1")
                     AND       TEST LOAD("GENSUB1")....ETC

                     OR        SUCCEED.
AND             TEST ERRON()
```

5. Changing Answers: 'What Ifs'

At the end of a particular consultation, the Operator may also wish to change the line of reasoning to see what effect it has upon a solution. This what-if facility was achieved in the Help-desk by defining the suite of rules associated with the consultation as being special and then using the WIPE command to erase previous answers. This process takes the operator back to the beginning point of the line of reasoning or to specified points indicated by wipe flags. The program example below shows how the wipe command was used:

```
IF              HELP EXPLAIN
AND             TEST START(" CHANGE ONE OF THE ANSWERS, RESULT$)
AND             MENU: WHAT_IF_MENU
AND             END EXPLAIN
AND FAIL

OR              TEST WHAT_IF_MENU = 1
AND             WIPE PRINT_QUALITY$
AND             GLOBAL RESTART

OR              TEST WHAT_IF_MENU = 2
AND             WIPE PRINT_ORIENTATION$
AND             GLOBAL RESTART              ETC....

OR SUCCEED
```

6. Using YES/NO Questions

Where the problems were highly structured, it was sufficient to define rules as a series of YES/NO consultations between the end-user and operator using the F1 Help key to clarify questions or provide a default response when the user was uncertain about a question. The basic programming format of a YES/No question is:

```
IF        YES/NO QUESTION ( Screen display with a question)
          THEN YES OPTION

OR        THEN NO OPTION
```

A typical YES/NO consultation is shown in Figure B4.In this example, the end-user can provide no symptom which might suggest why a printer is faulty and so it is necessary for the operator to start from fundamental trouble-shooting routines and work down a YES/NO type decision-tree, becoming progressively more complex. This process of interaction with the end-user continues until the operator has sufficient information to conclude that, in this case, the printer switch is at fault and suggests three possible solutions starting from the most likely.

YES/NO questions are very time consuming both in terms of programming and during trouble-shooting. Moreover, they are unsuited to multiple response or uncertainty scenarios. However, they can be highly effective in systematically reducing the number of possible options, especially where there is little initial information provided.

7. Error Messages

A quick and simple indication of faults was from error codes and error messages provided by the computer hardware equipment itself. However to be of use to the end-user it was necessary to interpret the error message and define simple guidelines which would clear the fault. This function was performed by the help-desk for terminal diagnostics and one such case is highlighted in Figure B5. Upon defining the scope of the problem in Screens A,B & C, the operator establishes the error code (this may be alpha-numeric, numeric or textual) from the user and enters it into the help-desk which subsequently provides advice to the operator on checks to make. In many cases, the problem is too complex to be resolved by the end-user and is therefore escalated to the expert, as in the case of Screen E.

8. Configuring Problems Using Menu-Driven Techniques

An important feature of the help-desk is to configure the problem situation on the basis of menu alternatives. At a high level menus are used to select knowledge-bases; while at a lower level menu alternatives are an integral part of the decision-making process (for example, 'which of these alternatives best describes the problem' etc). Menus can take the operator from an abstract level where symptoms of faults are known to within the boundaries of the fault itself. This process is accelerated through the use of System Profiles and improved further if the operator is able to volunteer information.

An example of menu decomposition is given in Figure B6. The problem is that an end-user cannot print a particular character on the laser printer. From the main menu, the operator has to decide, on the information provided by the end-user, in which broad category of fault the problem is likely to be in. In this case it is a 'print style/orientation' type fault. From this, the operator elicits more information from the end-user in order to define the exact boundary of the problem (again using menu screens). In the example for instance, this includes details about the type of printer and software being used at the time of the problem and .the type style number selected to print out the character. At a lower level of decision-making menus are then used to establish precisely what the end-user was trying to do -underline, embolden or print a character. In the example, the latter option is chosen and the operator establishes that for the character set used for this configuration a certain character is not supported. This information is passed on to the end-user with a list of alternative character options.

A problem in using menus is that many are often required in order to solve even basic problems. This is time consuming and involves a high degree of interaction with the end-user to be successful. However because menus were a well structured and systematic means of gathering information and narrowing down the experts' line of reasoning, they were used extensively in the help-desk.

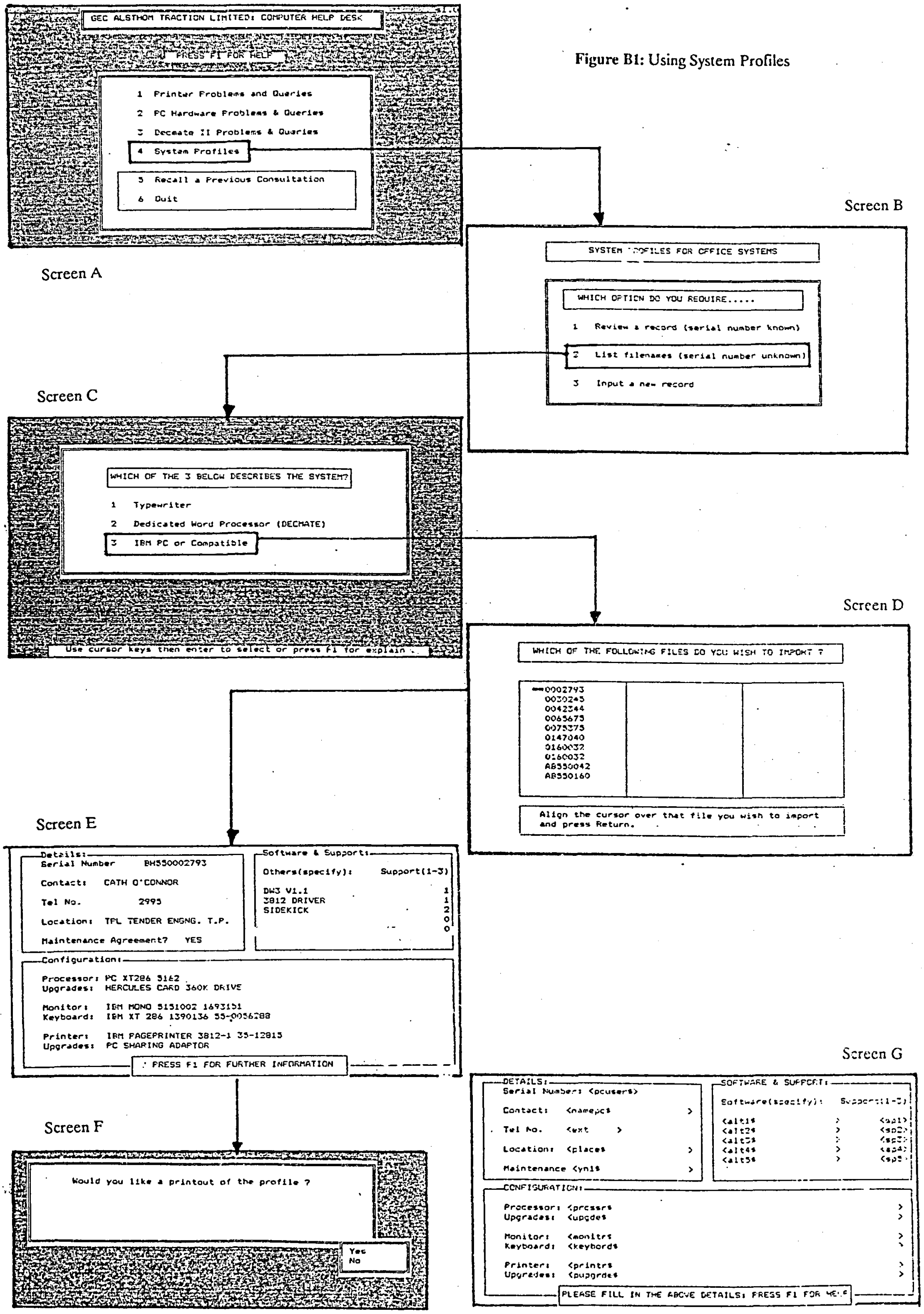


Figure B1: Using System Profiles

Screen A

```
GEC ALSTHOM TRACTION LIMITED: COMPUTER HELP DESK

PRESS F1 FOR HELP

1  Printer Problems and Queries
2  PC Hardware Problems & Queries
3  Decmate II Problems & Queries
4  System Profiles

5  Recall a Previous Consultation
6  Quit
```

Screen B

```
SYSTEM PROFILES FOR OFFICE SYSTEMS

WHICH OPTION DO YOU REQUIRE.....

1   Review a record (serial number known)
2   List filenames (serial number unknown)
3   Input a new record
```

Screen C

```
WHICH OF THE 3 BELOW DESCRIBES THE SYSTEM?

1   Typewriter
2   Dedicated Word Processor (DECMATE)
3   IBM PC or Compatible

Use cursor keys then enter to select or press F1 for explain .
```

Screen D

```
WHICH OF THE FOLLOWING FILES DO YOU WISH TO IMPORT ?

0002793
0030245
0042344
0065675
0075375
0147040
0160032
0160032
AB550042
AB550160

Align the cursor over that file you wish to import
and press Return.
```

Screen E

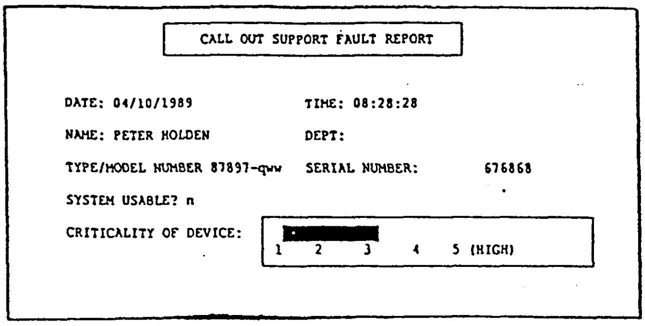
```
Details:
Serial Number      BH550002793
Contact:   CATH O'CONNOR
Tel No.          2995
Location:  TPL TENDER ENGNG. T.P.
Maintenance Agreement?   YES

Software & Support:
Others(specify):         Support(1-3)
DW3 V1.1                            1
3812 DRIVER                         1
SIDEKICK                            2
                                    0
                                    0

Configuration:
Processor: PC XT286 5162
Upgrades:  HERCULES CARD 360K DRIVE

Monitor:   IBM MONO 5151002 1693151
Keyboard:  IBM XT 286 1390136 55-0056288

Printer:   IBM PAGEPRINTER 3812-1 35-12815
Upgrades:  PC SHARING ADAPTOR

PRESS F1 FOR FURTHER INFORMATION
```

Screen F

```
Would you like a printout of the profile ?

Yes
No
```

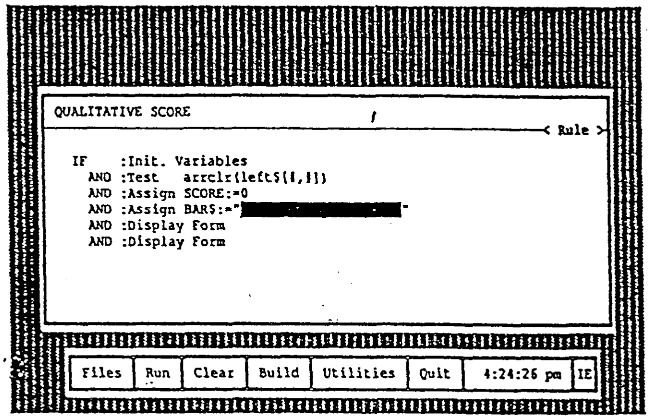
Screen G

```
DETAILS:
Serial Number: <pcusers>
Contact:    <namepc$                   >
Tel No.     <ext      >
Location:   <places$                   >
Maintenance <yn1$                      >

SOFTWARE & SUPPORT:
Software(specify):    Support(1-3)
<alt1$          >       <sp1>
<alt2$          >       <sp2>
<alt3$          >       <sp3>
<alt4$          >       <sp4>
<alt5$          >       <sp5>

CONFIGURATION:
Processor: <prcssr$                                  >
Upgrades:  <upgde$                                   >

Monitor:   <monitr$                                  >
Keyboard:  <keybord$                                 >

Printer:   <printr$                                  >
Upgrades:  <pupgrde$                                 >

PLEASE FILL IN THE ABOVE DETAILS: PRESS F1 FOR HELP
```



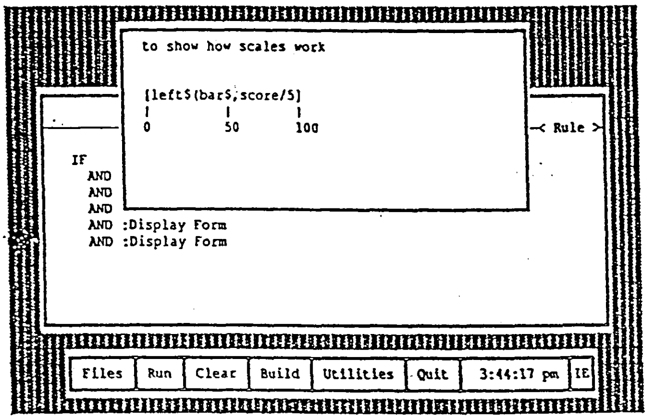
Screen A

```
CALL OUT SUPPORT FAULT REPORT

DATE: 04/10/1989                TIME: 08:28:28
NAME: PETER HOLDEN              DEPT:
TYPE/MODEL NUMBER 87897-qww     SERIAL NUMBER:        676868
SYSTEM USABLE? n
CRITICALITY OF DEVICE:   ████████
                         1    2    3    4    5 (HIGH)
```



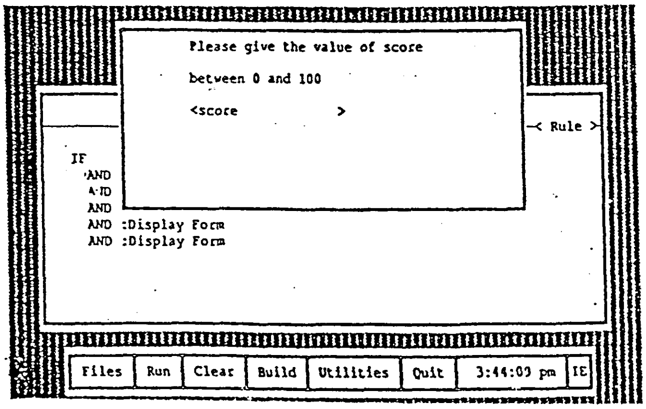
Screen B

```
QUALITATIVE SCORE
                                          < Rule >
  IF    :Init. Variables
    AND :Test   arrclr(left$(1,1))
    AND :Assign SCORE:=0
    AND :Assign BAR$:="████████████"
    AND :Display Form
    AND :Display Form
```



Screen C

```
to show how scales work

[left$(bar$,score/5)]
|          |         |
0          50        100

                                          < Rule >
  IF
    AND
    AND
    AND
    AND :Display Form
    AND :Display Form
```

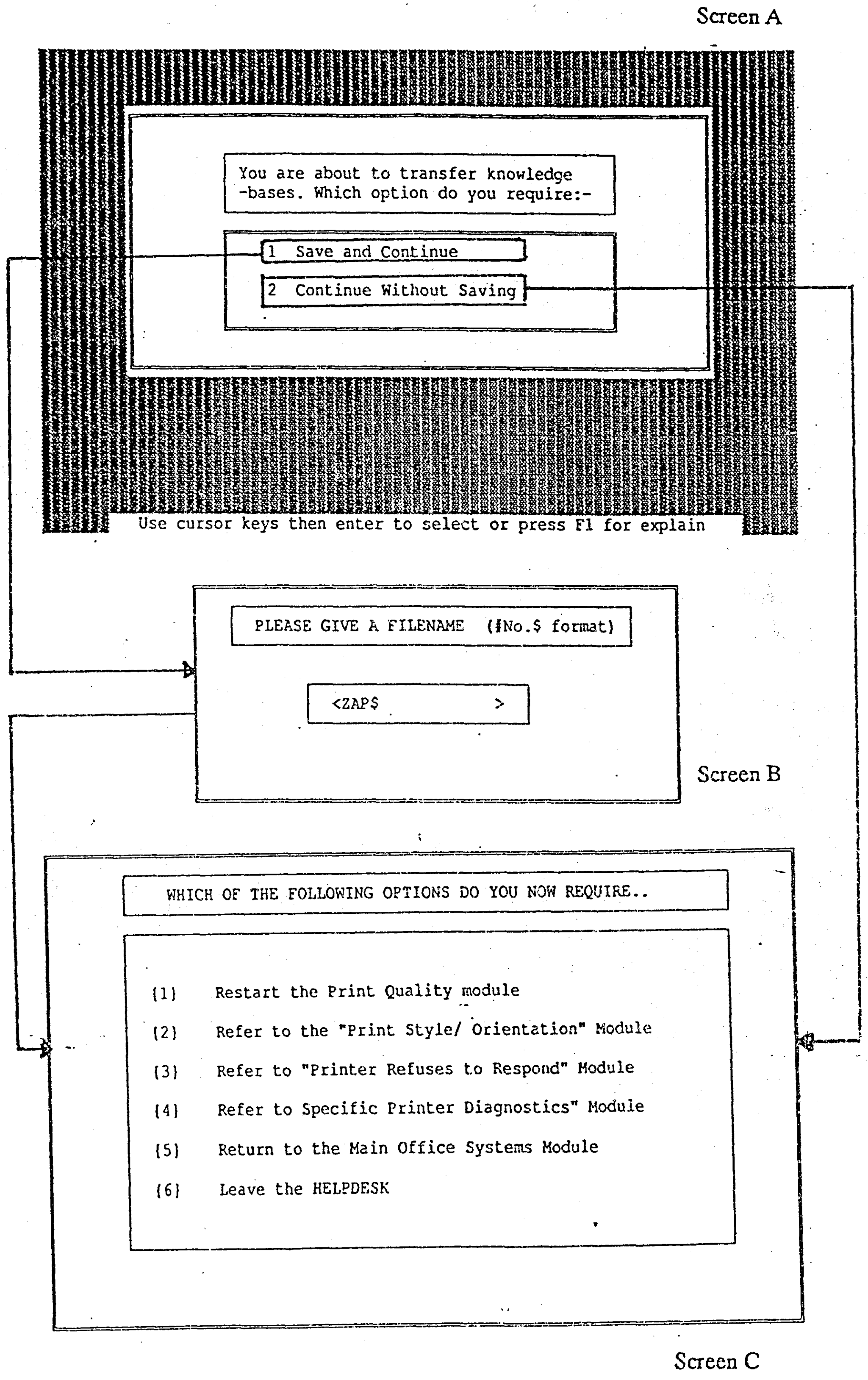


Screen D

```
Please give the value of score

between 0 and 100

<score              >
                                          < Rule >
  IF
    AND
    AND
    AND
    AND :Display Form
    AND :Display Form
```

Figure B2: An Example of Qualitative Scoring in Prioritising Faults

Figure B3: Saving Consultations



You are about to transfer knowledge -bases. Which option do you require:-

1 Save and Continue

2 Continue Without Saving

Use cursor keys then enter to select or press F1 for explain

PLEASE GIVE A FILENAME (#No.$ format)

<ZAP$ >

WHICH OF THE FOLLOWING OPTIONS DO YOU NOW REQUIRE..

(1) Restart the Print Quality module

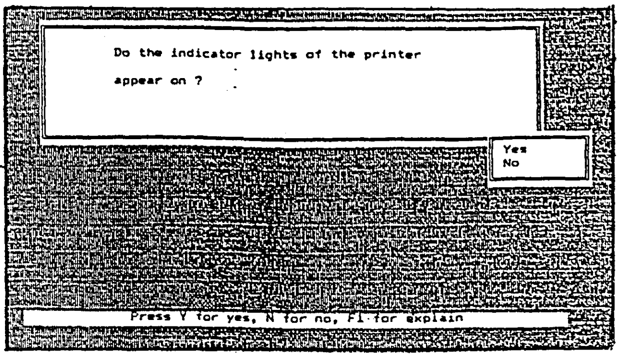
(2) Refer to the "Print Style/ Orientation" Module

(3) Refer to "Printer Refuses to Respond" Module

(4) Refer to Specific Printer Diagnostics" Module

(5) Return to the Main Office Systems Module

(6) Leave the HELPDESK

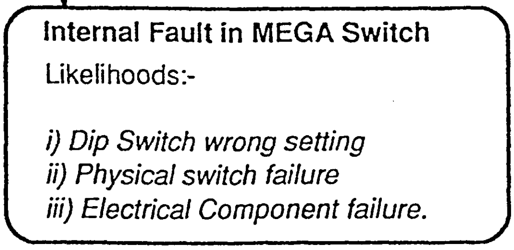


Do the indicator lights of the printer appear on ?

Press Y for yes, N for no, F1 for explain

Screen A

```
IF      DISPLAY FORM
        AND YES/NO QUESTION
        AND SWITCH_PRINTER_ON/OFF

OR      CHECK_MAINS_LEAD

        IF      YES/NO QUESTION
                AND CHECK_PRINTER_PC_CABLE

        OR      TEST_FOR_ERROR_MESSAGE

                IF      YES/NO QUESTION
                AND     CHECK_SOFTWARE

                OR      RETURN TO MAIN MENU

                        IF      YES/NO QUESTION
                        AND     CHECK_AUTOSWITCH

                        OR      ESTABLISH_PRINTER_SPECIFICS

                                IF      YES/NO QUESTION
                                AND     MANUAL_OR_AUTO_OPERATION?

                                OR      LOOK_CLOSER_AT_PC_HARDWARE

                                        IF      YES/NO QUESTION
                                        AND     MENU MorA$
                                                AND     TEST MorA$="M"
                                                AND     SELECT_MANUAL_OPTION

                                                OR      TEST MorA$="A"
                                                AND     SELECT_ AUTO_OPTION

                                                OR      RETURN_TO_MAIN_MEN

                                                        IF      YES/NO QUESTION
                                                        AND     LED_LIGHT_ON

                                                        OR      LED_LIGHT_NOT_ON
                                                                AND DISPLAY FORM
                                                                AND RETURN TO MAIN_MEN
```



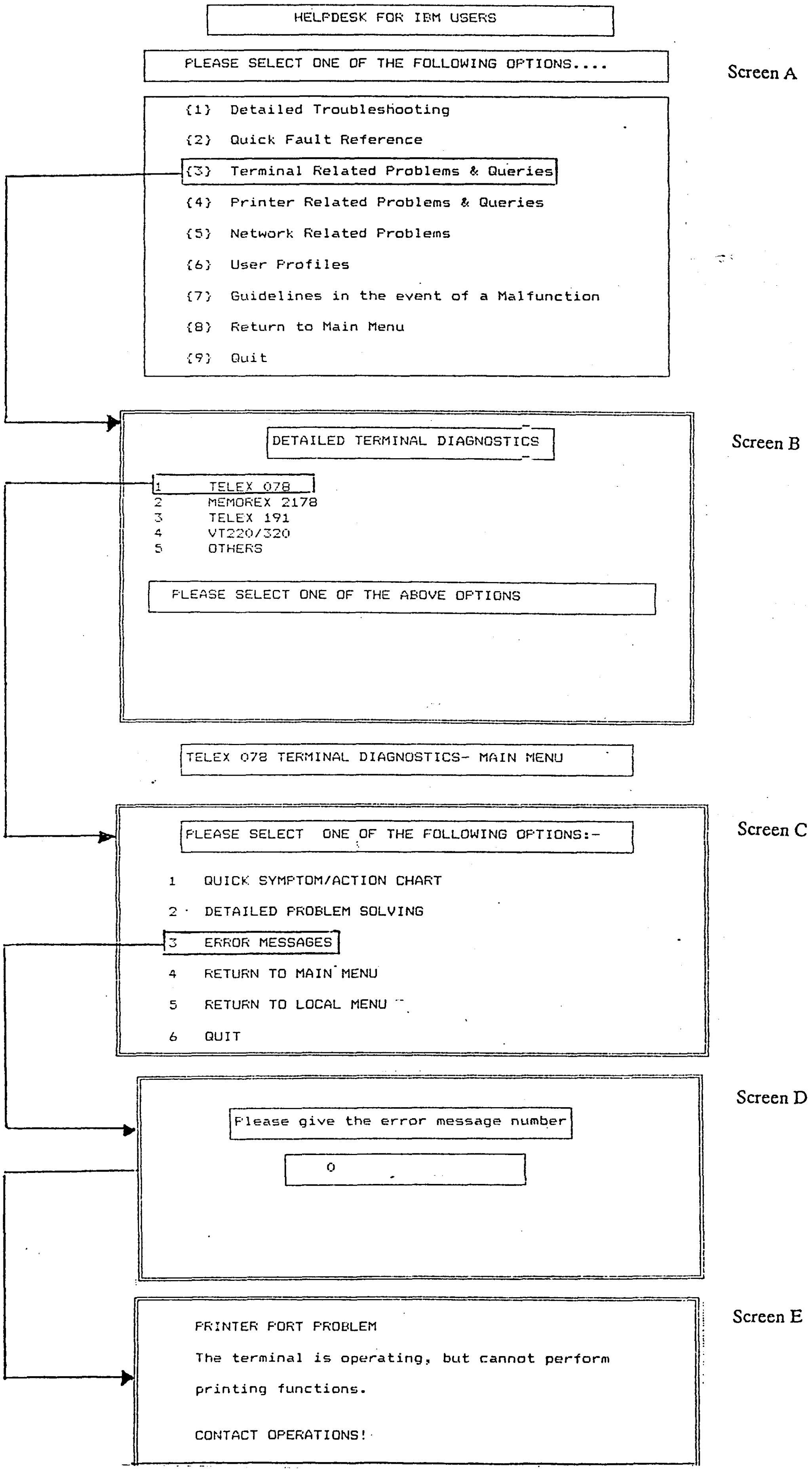
Internal Fault in MEGA Switch

Likelihoods:-

i) Dip Switch wrong setting
ii) Physical switch failure
iii) Electrical Component failure.

Screen B

Figure B4: Example Consultation Showing the YES/NO Question Format.



```
HELPDESK FOR IBM USERS

PLEASE SELECT ONE OF THE FOLLOWING OPTIONS....

{1}  Detailed Troubleshooting
{2}  Quick Fault Reference
{3}  Terminal Related Problems & Queries
{4}  Printer Related Problems & Queries
{5}  Network Related Problems
{6}  User Profiles
{7}  Guidelines in the event of a Malfunction
{8}  Return to Main Menu
{9}  Quit
```

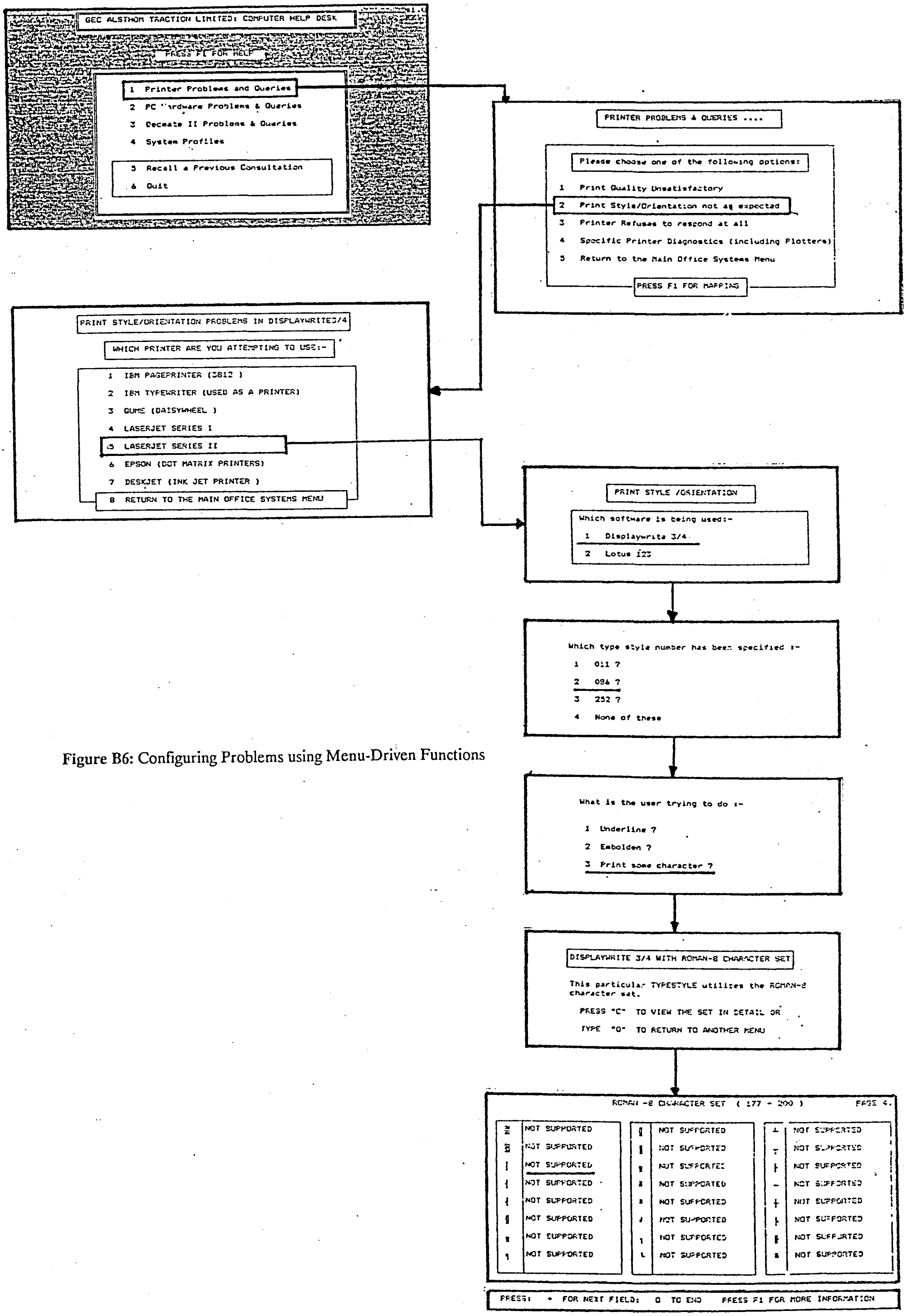
```
DETAILED TERMINAL DIAGNOSTICS

1      TELEX 078
2      MEMOREX 2178
3      TELEX 191
4      VT220/320
5      OTHERS

PLEASE SELECT ONE OF THE ABOVE OPTIONS
```

```
TELEX 078 TERMINAL DIAGNOSTICS- MAIN MENU

PLEASE SELECT  ONE OF THE FOLLOWING OPTIONS:-

1   QUICK SYMPTOM/ACTION CHART
2   DETAILED PROBLEM SOLVING
3   ERROR MESSAGES
4   RETURN TO MAIN MENU
5   RETURN TO LOCAL MENU
6   QUIT
```

```
Please give the error message number

     0
```

```
PRINTER PORT PROBLEM

The terminal is operating, but cannot perform

printing functions.

CONTACT OPERATIONS!
```

Figure B5: Example of Error Handling Capabilities in the Helpdesk



```
GEC ALSTHOM TRACTION LIMITED: COMPUTER HELP DESK

PRESS F1 FOR HELP

1  Printer Problems and Queries
2  PC "ardware Problems & Queries
3  Decmate II Problems & Queries
4  System Profiles

5  Recall a Previous Consultation
6  Quit
```

```
PRINTER PROBLEMS & QUERIES ....

Please choose one of the following options:

1  Print Quality Unsatisfactory
2  Print Style/Orientation not as expected
3  Printer Refuses to respond at all
4  Specific Printer Diagnostics (including Plotters)
5  Return to the Main Office Systems Menu

PRESS F1 FOR MAPPING
```

```
PRINT STYLE/ORIENTATION PROBLEMS IN DISPLAYWRITE3/4

WHICH PRINTER ARE YOU ATTEMPTING TO USE:-

1  IBM PAGEPRINTER (3812 )
2  IBM TYPEWRITER (USED AS A PRINTER)
3  QUME (DAISYWHEEL )
4  LASERJET SERIES I
5  LASERJET SERIES II
6  EPSON (DOT MATRIX PRINTERS)
7  DESKJET (INK JET PRINTER )
8  RETURN TO THE MAIN OFFICE SYSTEMS MENU
```

```
PRINT STYLE /ORIENTATION

Which software is being used:-
1  Displaywrite 3/4
2  Lotus 123
```

```
Which type style number has been specified :-
1  011 ?
2  086 ?
3  252 ?
4  None of these
```

```
What is the user trying to do :-
1  Underline ?
2  Embolden ?
3  Print some character ?
```

```
DISPLAYWRITE 3/4 WITH ROMAN-8 CHARACTER SET

This particular TYPESTYLE utilizes the ROMAN-8
character set.

PRESS "C"  TO VIEW THE SET IN DETAIL OR
TYPE  "O"  TO RETURN TO ANOTHER MENU
```

```
                    ROMAN -8 CHARACTER SET  ( 177 - 200 )           PAGE 4.

 ≡ | NOT SUPPORTED        ║ | NOT SUPPORTED        ┴ | NOT SUPPORTED
 ▣ | NOT SUPPORTED        ▌ | NOT SUPPORTED        ┬ | NOT SUPPORTED
 │ | NOT SUPPORTED        ▪ | NOT SUPPORTED        ├ | NOT SUPPORTED
 ┤ | NOT SUPPORTED        ▪ | NOT SUPPORTED        ─ | NOT SUPPORTED
 ┤ | NOT SUPPORTED        ▪ | NOT SUPPORTED        ┼ | NOT SUPPORTED
 ╢ | NOT SUPPORTED        ┘ | NOT SUPPORTED        ├ | NOT SUPPORTED
 ▪ | NOT SUPPORTED        ┐ | NOT SUPPORTED        ╟ | NOT SUPPORTED
 ┐ | NOT SUPPORTED        └ | NOT SUPPORTED        ▪ | NOT SUPPORTED

PRESS:   ◆  FOR NEXT FIELD:   Q  TO END    PRESS F1 FOR MORE INFORMATION
```

Figure B6: Configuring Problems using Menu-Driven Functions

Appendix XIV.

An Account of the Office Systems Help Desk

Contents: -

Appendix XIV: Part A

Office Systems Help Desk: Structure and Operations

1. Overview of the Help-desk

Knowledge for the Help-desk has come from two sources. Firstly from the specialists in the form of rules-of-thumb on how problems have been solved in the past; and secondly from technical documentation such as Vendor Manuals and company Codes of Practice which provide regulatory information to complement the specialist's knowledge.

The Help-desk has three main functions. The first and most important function is to trouble-shoot all printers (including typewriters and plotters), PC hardware, and Dedicated Word Processor problems and queries in the company. The second function is to provide on-line access during consultation of systems configuration and User details, both of which are contained in a database linked to the help-desk. The final function of the Help-desk is to record the types of faults which arise for future analysis (frequency of calls, recurring problems, allocation of resources etc). This is made possible through the design of Save & Recall facilities which allocate a particular consultation to a user filename. These primary functions are outlined in **Figure 1.**

2. Help-desk Structure

This section briefly describes the main components of the Help-desk and how they interrelate.

2.1 Top-level or Meta Control

The help-desk is made up of a number of medium sized knowledge bases controlled by a top-level knowledge base called "Meta-control". The Meta-control function decides when to load or exit a particular knowledge-base; when to refer to external sources- in the case of the Help-desk, this is presently restricted to an ASCII file-handling system; and when to import and export data-files. This multi-level structure is outlined in **Figure 2.** The benefits of adopting a meta-control structure are that it improves the "usability" and logic of the Help-desk and also makes programming and maintenance tasks easier. However, there are also practical constraints in using Crystal which limit the size of each knowledge base to approximately 200Kb in size. When this limit is reached, the knowledge-base is partitioned into a number of smaller and independent knowledge-bases: the split being made on the basis of problem category.

When using the Help-desk, you will frequency see screens indicating that new knowledge-bases at different levels in the problem hierarchy are being loaded and you will be given the option to save the consultation in the previous knowledge-base before proceeding to the next.

The average consultation will take the user through 3-4 different knowledge-bases before arriving at the required solution. At the end of a consultation you may wish to restart the same knowledge-base, to perform "WHAT-IF" type scenarios for instance, or return to a different knowledge base which is perhaps at a different problem level. This is achieved by the use of referral screens, as shown for example in **Screen 2,** which allow the user to proceed to other areas of the Help-desk without the need to ascend to higher level screens.

2.2 Printer Problems and Queries

This is the largest module in the help-desk and is broadly divided into four main sections as shown in **Figure** 3.and **Screen 2.** These sections are described overleaf.

i) Print Quality Unsatisfactory: This section includes characteristic print quality problems for each of the printers and typewriters identified in **Figure 3.1.** A typical consultation will first check that the correct paper is being used and that "consumables", such as ink cartridges, ribbons, daisywheels and so on, are operable and properly connected to the printer. Should the fault persist, then more detailed and systematic questions are asked in which the Help-desk tries to identify the exact nature of the print defect through the elimination of possible symptoms (complete image is dark, blurred stripes across the top edge of the paper, dark vertical lines running parallel to the long edge of the paper etc.). When there are enough symptoms to indicate a particular fault, then the corresponding corrective measures are recommended.

ii) Print Style / Orientation Not as Expected: In order to provide full support for the various combinations of software and printers, it is necessary to correctly configure the software (Displaywrite 3/4 or Lotus 123), typestyle(011, 018 etc), printer(see **Figure 3.2**), character set(IBM PC, Roman-8, LIC PC, Qume Bilingual and IBM Typewriter) and cartridge (for the Laserjet series only), ·together with the text function required (emboldening or underlining) and the print orientation (portrait/landscape).

For each particular permutation, the Help-desk states whether the default printer set-up will support a character or text function given the above constraints. Where a character is not supported, the Help-desk may define an alternative character option using the ALT Key and NUM LOCK key pad.

iii) Printer Refuses to Print At All: The basic structure of this section is given in **Figure 3.3.** The Help-desk consultation begins by asking a number of "broadsweep" questions about the printer itself; for instance whether it is on-line and the indicators lights are on etc. Once satisfied that the basic conditions for the printer to work have been met, the Help-desk enquires about peripheral equipment, such as cables and switches, which link the printer to the PC. This is proceeded by top-level checks at the PC end-whether the printer is configured correctly for the software for example. If the problem persists, then the consultation goes back to the printer where more detailed and technical questions are asked about the operations of the printer.

iv) Specific Printer Diagnostics: These are diagnostic procedures specific to each of the printers shown in **Figure 3.4.** It also includes troubleshooting guide-lines for Typewriters (see **Screen 3.**) and Plotters. A large amount of the technical information used in this section originated from Manufacturers' manuals and other documented information, which was then structured by the specialist Support Engineers to present this information in a form which was consistent with their own troubleshooting procedures.

2.3 PC Hardware Problem & Queries

This module covers only top-level or shallow problems because of the complexity of the domain and therefore the specialist skills required to undertake even basic troubleshooting tasks. The module is divided into four basic sections, as outlined in **Figure 4.**

> **i) Keyboard Problems:** This section combines broad-sweep top-level troubleshooting with more detailed manufacturers' tests for IBM, Amstrad and Compaq machines.
>
> **ii) Monitor Problems:** Although manufacturers' tests are available, it was considered potentially hazardous to expect non-specialists to undertake other than fundamental broad-sweep tests. Should the fault persist then the specialist should be contacted immediately.
>
> **iii) System Unit Problems:** This is divided into three components. Firstly, device error messages which, upon identifying the model of machine and DOS version, will list the most frequently occurring faults and suggest appropriate remedial action.

The second component addresses floppy disk drive problems. In response to either "not ready" or "general failure" type error, the Help-desk suggests various tests which should be carried out on the basis of simple YES/NO format consultations with the user.

The third component offers the user a selection of fault symptoms from a menu. The user is asked to select a combination of symptoms, with each combination signifying a likely fault source.

> **iv) Power-up Error Codes:** This section provides assistance in the interpretation of power-on self-test messages for IBM, COMPAQ and AMSTRAD machines and offers advice on possible solution requirements.

2.4 Decmate II: Problems & Queries

This module is divided into four main sections, as shown in **Figure 5.** As with PC Hardware, the first section tests all "boot-up" error codes and recommends subsequent remedial action; it is also cross-referenced to all other sections in the module.

The second section looks specifically at disk related problems and the third section addresses printer related problems. As **Screen 4** shows, the information is presented to the user as a list of symptoms from which the selection of a particular group will indicate the likely cause of a fault.

Power related faults are covered in the final section which also provides advice on action to take in the event of power loss.

2.5 System Profiles

The System Profile provides you with information on Help-desk callers, the system configuration of their machines and the software they use. This information is presented in the form of look-up Screen displays (see **Screen 6** for example). Additional information is also supplied and this includes the level of software support and whether maintenance agreements exist for each system used. Three basic classes of machine are covered by the Profile: these are IBM PCs, Typewriters and Dedicated Word Processors (DWP).

3. Help-Desk Operations

3.1. Using Crystal

It is recommended that Crystal applications are run on an IBM PC AT or compatible minimum, and preferably a 286 specification machine, especially since there are 10 knowledge-bases.

Minimum RAM memory is 192K RAM, single or twin floppy machine, although again a hard disk configuration is recommended. Crystal will run on a colour or monochrome screen.

The Help-desk requires DOS 2.0 and above to operate and may also run on OS/2 in "compatibility box". If it is intended to use the graphics upgrade, then CGA, EGA or Hercules monitors are required.

Most printers may be connected to Crystal on the standard default setting of LPT1 on COMMS1. Note that Dot-Matrix printers do not provide an adequate resolution for printing from Crystal (including screen dumps) and it is recommended that the HP Deskjet, HP Laserjet (s.II) or IBM Pageprinter or Postscript Laser printer be used.

3.1.1 Installation & Starting the Help-desk: The Help-desk may be loaded onto your machine in two ways according to the intended role of the user:-

a) Run-time Only: A run-time version of the Help-desk is loaded onto your machine by a member of the Computer Department. Using this version, you will not have access to the program or to any of the development functions. In return, the machine requires half the memory and is significantly faster to operate.

b) Development Version: For users involved in development, it is important that the application operates in both run-time and development modes so that prototyping and maintenance can be undertaken. After Crystal is installed (see the Crystal manual A-7 to A-11) the Help-desk program may be loaded.

To start the Help-desk, switch on the machine and wait for the **C >** DOS prompt. Then type **HDESK** and press ENTER of Carriage return. Help-desk is loaded as a batch file in Autoexec.bat and will take you immediately to the main menu screen of the Help-desk application, as shown in **Screen 1.**

To leave the Help-desk at any time, choose the **QUIT** option from the menu and this will take you back to the DOS prompt.

3.1.2 Using Function & Cursor Keys

Nearly all options in the Help-desk are menu-driven, with each option being placed on a separate line and ascribed a sequence number. Menus are generally preferred as a means of presenting information to the user because of their clarity, ease of use and speed. There are two ways of making a selection from the menu layout: either typing in the sequence number directly from the central keyboard (this will automatically register the option chosen), or by using the arrow keys (known as cursor keys). These are the four keys together to the right of the keyboard with short arrows on them facing up, down, left and right. They move to the next or previous menu option up or down the screen. Make sure that the NUM LOCK light is OFF otherwise Crystal will record the first menu option nearest to the cursor. Once an option has been chosen, you know at which option the cursor is at because the number sequence will be highlighted, press return and this option will be recorded and the consultation will proceed in a similar way.

Although there will be some variations among machines, the following Keys are used in the Help-desk application:-

a) Main Character Keys: The letters and digits on the central part of the keyboard should be used to type in words and numbers. Although you may have a separate numeric keypad, you are advised to use this for cursor manipulation and therefore the "NUM LOCK" should be left OFF.

b) SHIFT Key: To enter capital letters or the characters above the numbers press the appropriate key with the shift key depressed. There are usually two shift keys, one each side of the central keyboard, with a large arrow pointing upwards on them.

c) ENTER Key: In a number of cases in the Help-desk, the user is asked to enter information. For instance, when giving a serial number in the System Profile section or an error code in Printer Diagnostics. In either case, simply type over the highlighted bar and press ENTER (or RETURN) when typing is complete. If the data you have entered is incorrect, the system will allow you a second attempt to re-input the data after which you are taken back to the immediate higher level menu.

d) TAB Key: The TAB key is the one with a left and right arrow on it. It is used to jump on to the next field on a form going from left to right. If it is pressed when the SHIFT key is depressed, then it will jump back to the previous field.

e) Backspace Key: This key has a single long left facing arrow on it and will delete the character to the left of the cursor and moves the cursor to that position. It is useful when editing text.

f) CTRL key: The control key is used in combination with other keys.

g) DEL key: The delete key deletes the character at the cursor position.

h) INS key: The insert key enables a character to be inserted between two others.

j) Reset: Occasionally, temporary problems can prevent the Help-desk from working properly, making it not possible to control the system from the keyboard. Such problems can be resolved by performing a reset. To do this, press the CTRL ALT DEL keys simultaneously and then reload the help-desk.

k) F1 function key: If there is an explanation screen it can be displayed by pressing the key labelled F1. If no explanation key has been prepared, then Crystal automatically displays the rule that was last executed and thereby substantiates the premises behind the question. This default help process provides a limited understanding and it advised that for more important questions, help screens are prepared. By pressing the ESCAPE key, the user is taken back to the presentation.

l) Esc Key: Depending upon whether this function has been enabled in the batch file (refer to the technical reference manual), pressing the Escape key will take the user out of the run-time mode of the Help-desk into development mode. For users not involved in development, it is preferable to disengage the Escape key.

m) HOME key: Positions the cursor in the home position, as defined by the application in use.

n) END: Positions the cursor in the end position, as defined by the application in use.

3.1.3 Responding to Crystal Prompts

These are prompts which are provided automatically by Crystal in response to certain command structures:

a) Press Y for YES, N for NO: This prompt appears at the bottom of the screen when a YES/NO reply is required. You should select the answer by pressing the "Y" or "N" key. Alternatively, you can place the highlight bar on the correct option using the cursor keys and press the Enter key.

b) Press any Key to CONTINUE: This prompt appears at the bottom of the screen when you only need to read the information on the screen. When you have read the information press any key to continue.

c) Please Type a Number: This prompt appears at the bottom of the screen when you must enter a number. The number will be entered in the field marked with the highlight bar. When you have typed in the number press ENTER to select another field in order to pass the number to Crystal for processing. If the number is unacceptable the screen will not change. If the number is accepted, the highlight bar will move to the next field on the same screen. If you press the ENTER key on the last field of the screen , Crystal will go to the next screen.

Numbers can be entered with decimal places following a decimal full stop. If a value is displayed then it will remain unless overtyped. If you are entering a number with only decimal places you must type a zero before the decimal point.

Fields can be edited using the INS (for inserting characters), DEL (for deleting characters) and Backspace Keys. You may move around the screen editing values until they are correct. Then complete the screen by pressing the END key followed by the ENTER key.

d) Please Type Your Response: This prompt appears at the bottom of the screen when you must type general information. The information you type will be entered in the field marked with the highlight bar. When you have entered the information, press the ENTER key. You may move around the screen typing fields, as described in the last section. In the same way, pressing ENTER on the last field will complete the screen.

e) Press F1 to Explain: This prompt will be added to the prompt at the bottom of the screen whenever explanation or help facilities are available. To see the explanation press the F1 function key. The explanation will either list the logic of the rules being used or will be a general text screen. To return to the screen you were on at any time in the explanation press the escape (ESC) key.

f) Other Screens: There are a number of prompts which are specific to the Help-desk application. These may either complement or replace the screens provided by Crystal:-

i) *Transferring Knowledge-Bases. Please Wait !* As **Figure 8** shows, the Help-desk has ten main knowledge-bases. For ease of programming and future maintenance, each knowledge-base contains information about a particular problem category with the ability to cross-reference to other knowledge-bases. Cross-referencing takes place when the line of reasoning has been exhausted in a particular knowledge-base and by deduction the logic is transferred to another knowledge-base. Transferring knowledge-bases may take up to twenty seconds and therefore the user is advised to wait.

ii) In some areas of the Help-desk, the user is asked:" *You are about to transfer knowledge-bases, do you wish to save* ?" If the user responds yes, then he or she will be asked to give a filename (see later sections) before the transfer takes place. Otherwise, transfer occurs without saving the consultation in the previous knowledge-base.

iii) Loading Knowledge-base. Please Wait !: This prompt simply confirms to the user that knowledge base transfer is underway.

iv) Please Select one of the Following Responses: This prompt is used for screen manipulation in the printer troubleshooting module. It refers to the selection of one of four options:-

P - refer to the previous screen
N - refer to the next screen
C - continue with the display
Q - leave the display and return to the main menu.

3.1.4 Error Messages

There are two types of error which occur during operations. Those that are generated automatically by Crystal and indicate a logic problem or data/file transaction problem; and those that are created by the applications developer to respond to errors in data input by the user. The main errors that are likely to occur in the Help-desk are:-

" File does not exist"
" Knowledge-base does not exist"
" Cannot open file- press any key to continue"
" Export Failure"
" Import Failure"

A full listing of errors is given in the Crystal Manual.In all these cases, you are advised to repeat the operation taking care to type in the correct data in the correct field. If the error message persists contact the Computer Department.

Error messages are created by the developer in cases where Crystal fails to provide an error message and instead forces the user out of the application with no explanation. This usually arises when the user is asked to type in one of a number of options and by mistake (spelling, omissions etc) fails to make a selection from the list. A "safety net" is therefore programmed into the Help-desk which first indicates to the user of an error, erases the value attributed to the menu variable, and loops back to repeat the original question. This procedure is repeated until an acceptable response is made.

3.2. Using the System Profiles

The Profile performs three operations and these are shown in **Figure 6** and **Screen 5.** The first two are for information retrieval whereby you will be asked to type in the last seven digits of the serial number for the system processor of the caller. If the correct number is given then the respective screen display is called up automatically. Otherwise you will be prompted to retry typing in the serial number. In the event that the serial number is not known, or you fail to type in the correct number sequence, then you will be referred to a list of all filenames and be able to call up a screen by placing the highlighted bar over the file required and pressing the ENTER key. In either event, you will be given the option to receive a hard copy of the screen display for records.

The third facility allows new records to be added. First of all, you will be asked to specify which class of machine is required (IBM PC, Typewriter or DWP). Next you will be taken to an empty display screen and asked to insert the relevant information to each of the field headings specified by the position of the highlighted bar. Some fields are numeric and will not accept text in any form. Once an insertion has been made, press ENTER and you will be taken to another field indicated by the highlighted bar. When all fields have been completed, you will be asked to provide a filename, which, as before, is the last seven digits of the serial number for the machine processor.

3.3 Saving and Recall Facilities

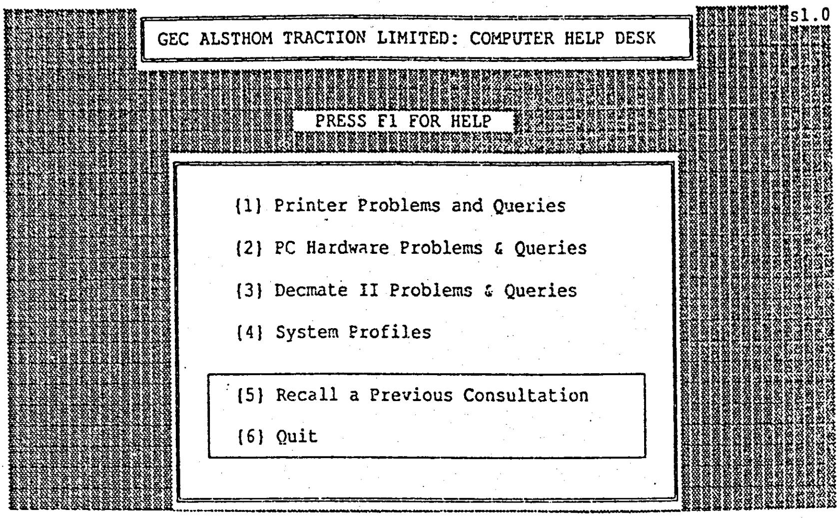
At the end of most consultations of the Help-desk, you will be given the option to "Continue without Saving" or " Save and Continue". In either case you will have the option to restart the consultation in the knowledge-base you are presently in, refer to any other knowledge-base, quit or return to the main help-desk menu (see **Screen 7**). Should you decide to save the consultation, then you will be asked to save it under a filename. This should be no more than seven characters in length and for ease of recall, should be in the format:-

INITIALS-DAY/MONTH (e.g PH1012, IES0104 etc)

All files and records are saved in an ASCII interface held within the main Crystal program. However, upon request files can be loaded and unloaded from the A: floppy disk drive.

In order to *recall a previous consultation*, you should first choose the option so named from the main menu. This will take you to a list of filenames. Align the highlighted bar over that file which you wish to "import" and press the RETURN key. This will take you to the beginning of the last knowledge-base you consulted during this particular problem-solving session.

Once within the knowledge-base, the previous path taken is indicated by the position of the highlighted bar. You are asked to confirm this by pressing the RETURN key. This process of retracing responses is repeated through all subsequent screens until the point is reached where a solution to the problem was found.



GEC ALSTHOM TRACTION LIMITED: COMPUTER HELP DESK

s1.0

PRESS F1 FOR HELP

{1} Printer Problems and Queries

{2} PC Hardware Problems & Queries

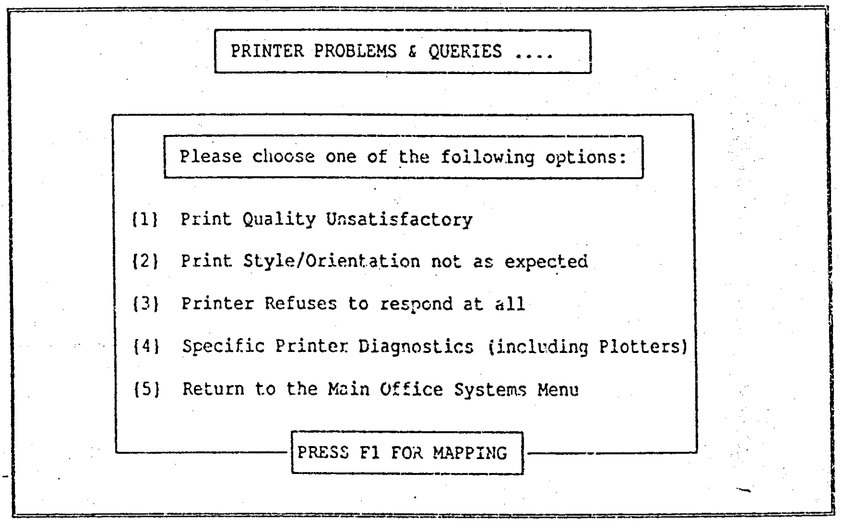
{3} Decmate II Problems & Queries

{4} System Profiles

{5} Recall a Previous Consultation

{6} Quit

Screen 1: Title Screen & Main Menu



PRINTER PROBLEMS & QUERIES ....

Please choose one of the following options:

{1} Print Quality Unsatisfactory

{2} Print Style/Orientation not as expected

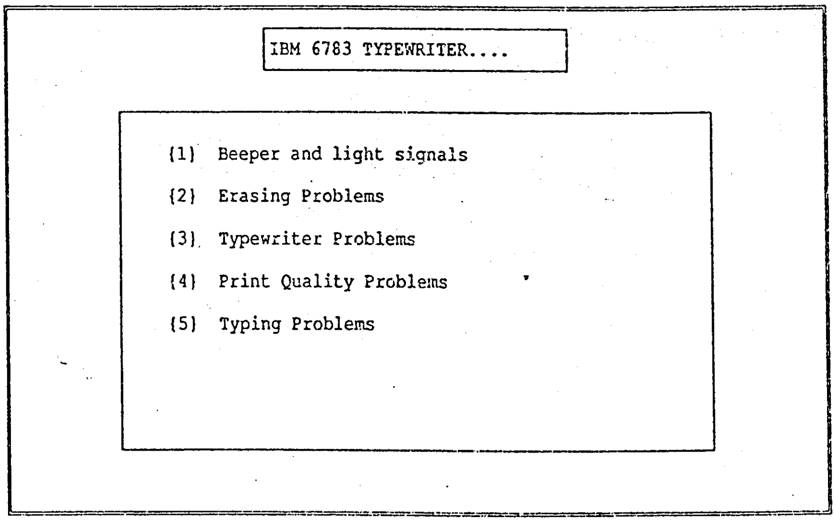
{3} Printer Refuses to respond at all

{4} Specific Printer Diagnostics (including Plotters)

{5} Return to the Main Office Systems Menu

PRESS F1 FOR MAPPING

Screen 2: Top-level Menu for the Printer Section



IBM 6783 TYPEWRITER....

{1} Beeper and light signals

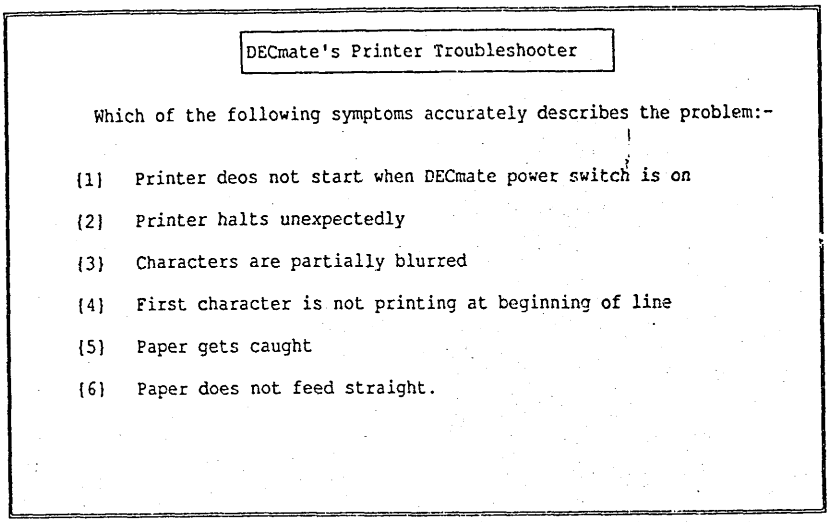
{2} Erasing Problems

{3} Typewriter Problems

{4} Print Quality Problems

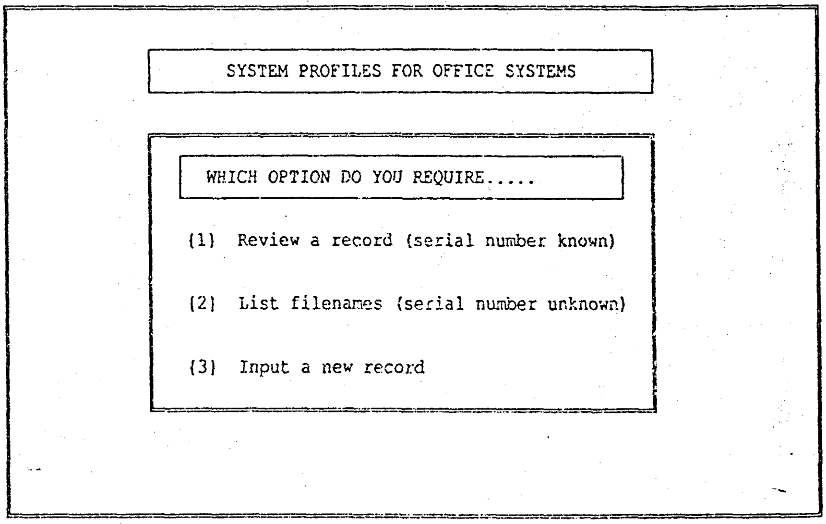
{5} Typing Problems

Screen 3: Top-level Menu for the Typewriter Section



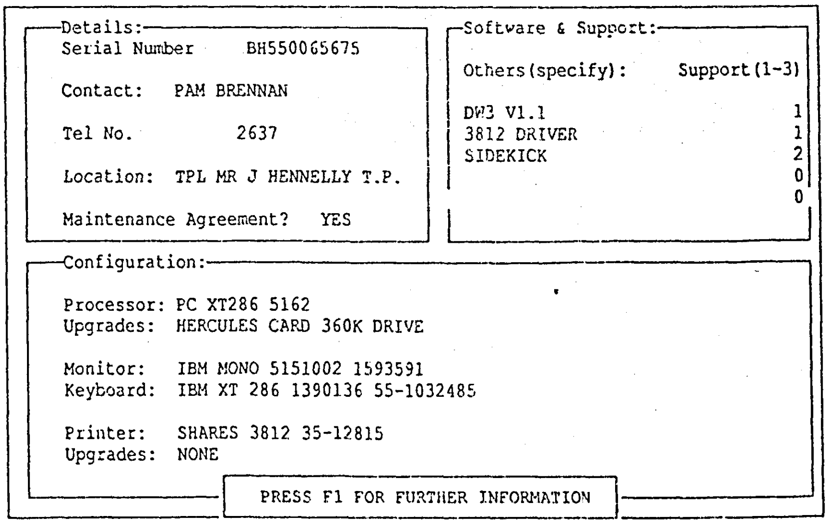
```
DECmate's Printer Troubleshooter

Which of the following symptoms accurately describes the problem:-

{1}  Printer deos not start when DECmate power switch is on

{2}  Printer halts unexpectedly

{3}  Characters are partially blurred

{4}  First character is not printing at beginning of line

{5}  Paper gets caught

{6}  Paper does not feed straight.
```

Screen 4: Top-level Menu for the Decmate II Section



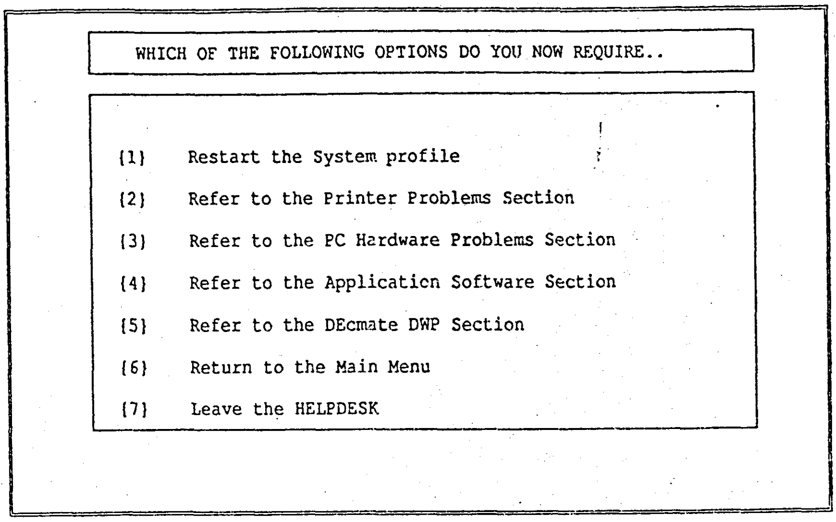
```
SYSTEM PROFILES FOR OFFICE SYSTEMS

WHICH OPTION DO YOU REQUIRE.....

{1}  Review a record (serial number known)

{2}  List filenames (serial number unknown)

{3}  Input a new record
```

Screen 5: Top-level Menu for the Systems Profile



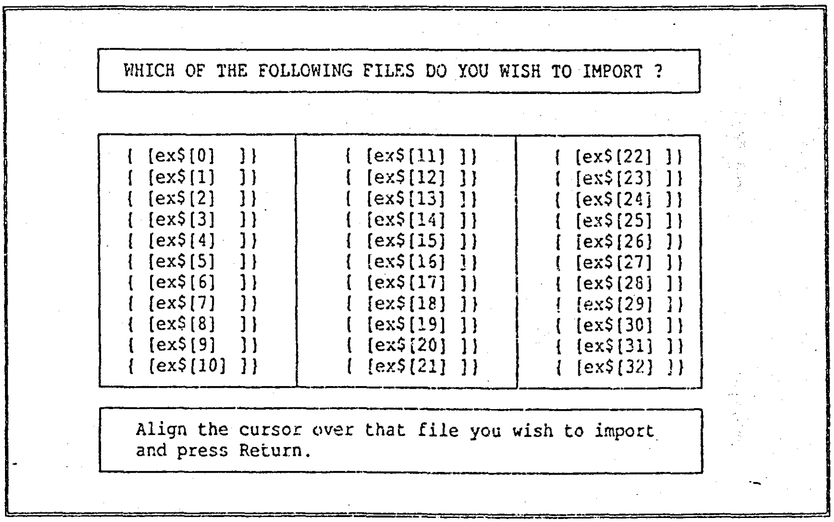
```
Details:                                   Software & Support:
Serial Number     BH550065675
                                           Others(specify):     Support(1-3)
Contact:   PAM BRENNAN
                                           DW3 V1.1                        1
Tel No.          2637                      3812 DRIVER                     1
                                           SIDEKICK                        2
Location:  TPL MR J HENNELLY T.P.                                          0
                                                                           0
Maintenance Agreement?   YES

Configuration:

Processor: PC XT286 5162
Upgrades:  HERCULES CARD 360K DRIVE

Monitor:   IBM MONO 5151002 1593591
Keyboard:  IBM XT 286 1390136 55-1032485

Printer:   SHARES 3812 35-12815
Upgrades:  NONE

           PRESS F1 FOR FURTHER INFORMATION
```

Screen 6: An Example Screen Profile



```
WHICH OF THE FOLLOWING OPTIONS DO YOU NOW REQUIRE..

{1}   Restart the System profile
{2}   Refer to the Printer Problems Section
{3}   Refer to the PC Hardware Problems Section
{4}   Refer to the Application Software Section
{5}   Refer to the DEcmate DWP Section
{6}   Return to the Main Menu
{7}   Leave the HELPDESK
```

Screen 7: End of Consultation Knowledge-Base Transfer



```
WHICH OF THE FOLLOWING FILES DO YOU WISH TO IMPORT ?

{ [ex$[0]  ]}    { [ex$[11] ]}    { [ex$[22] ]}
{ [ex$[1]  ]}    { [ex$[12] ]}    { [ex$[23] ]}
{ [ex$[2]  ]}    { [ex$[13] ]}    { [ex$[24] ]}
{ [ex$[3]  ]}    { [ex$[14] ]}    { [ex$[25] ]}
{ [ex$[4]  ]}    { [ex$[15] ]}    { [ex$[26] ]}
{ [ex$[5]  ]}    { [ex$[16] ]}    { [ex$[27] ]}
{ [ex$[6]  ]}    { [ex$[17] ]}    { [ex$[28] ]}
{ [ex$[7]  ]}    { [ex$[18] ]}    { [ex$[29] ]}
{ [ex$[8]  ]}    { [ex$[19] ]}    { [ex$[30] ]}
{ [ex$[9]  ]}    { [ex$[20] ]}    { [ex$[31] ]}
{ [ex$[10] ]}    { [ex$[21] ]}    { [ex$[32] ]}

Align the cursor over that file you wish to import
and press Return.
```

Screen 8: File Transfer & Recall Facility

Figure 1: The Office Systems Helpdesk: Structure & Operations

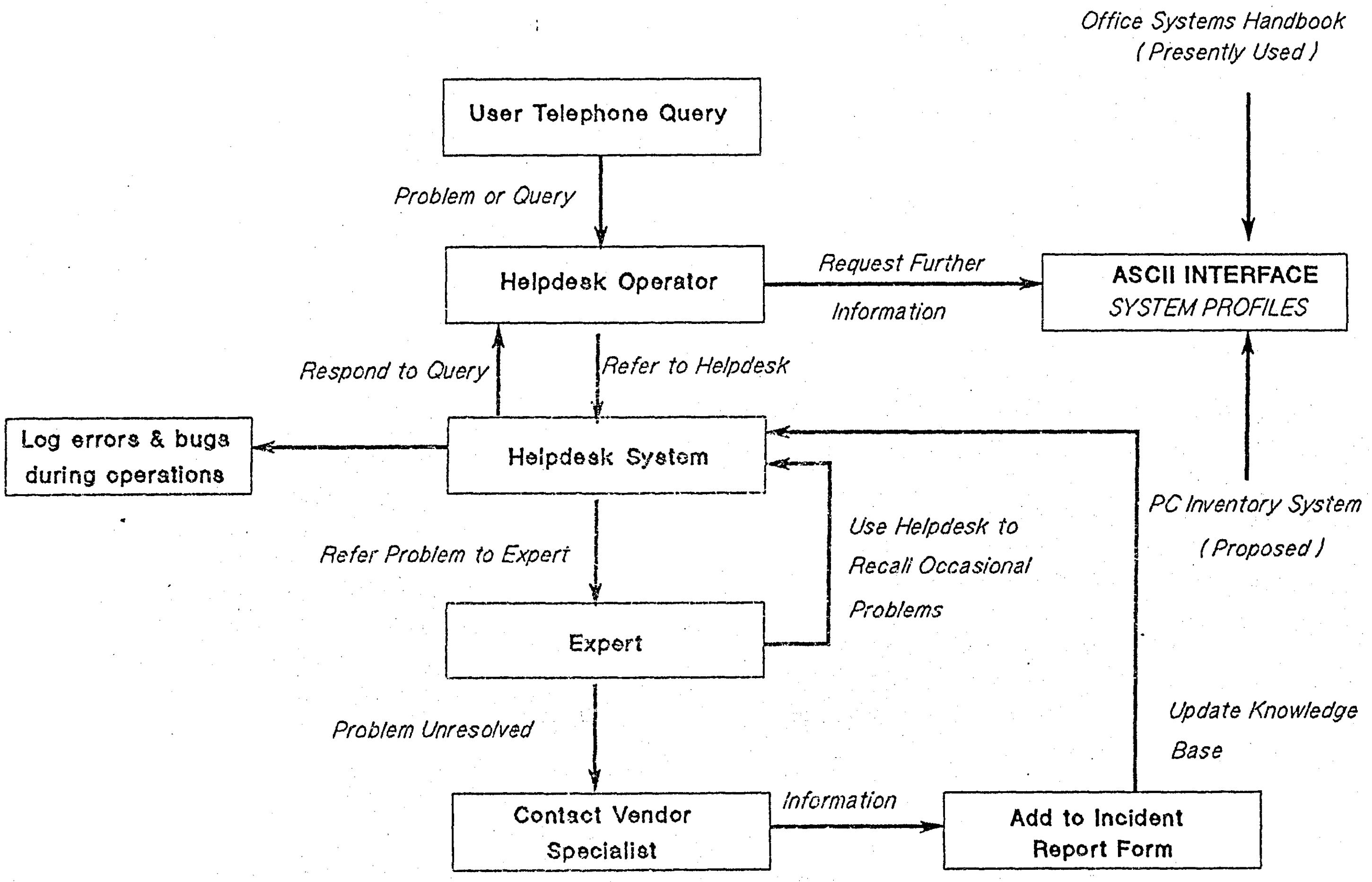
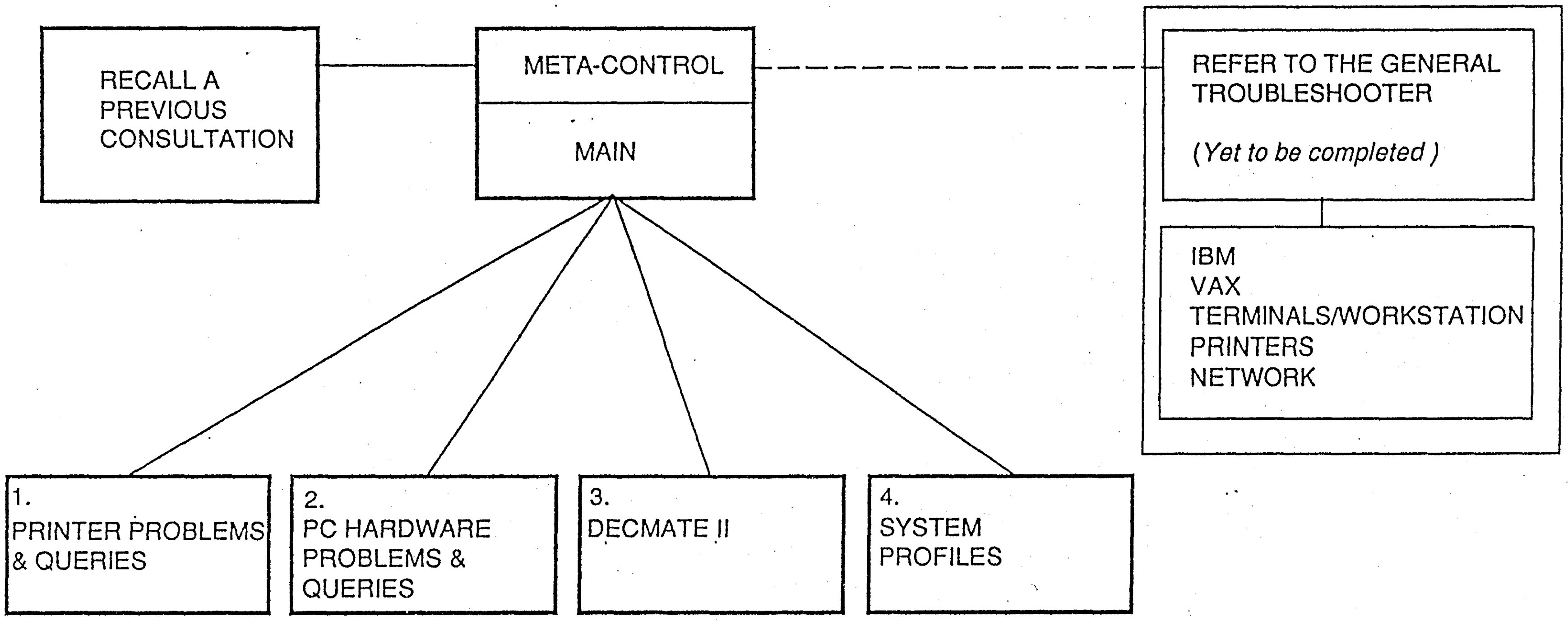


Figure 2: Top-Level Structure of the Helpdesk



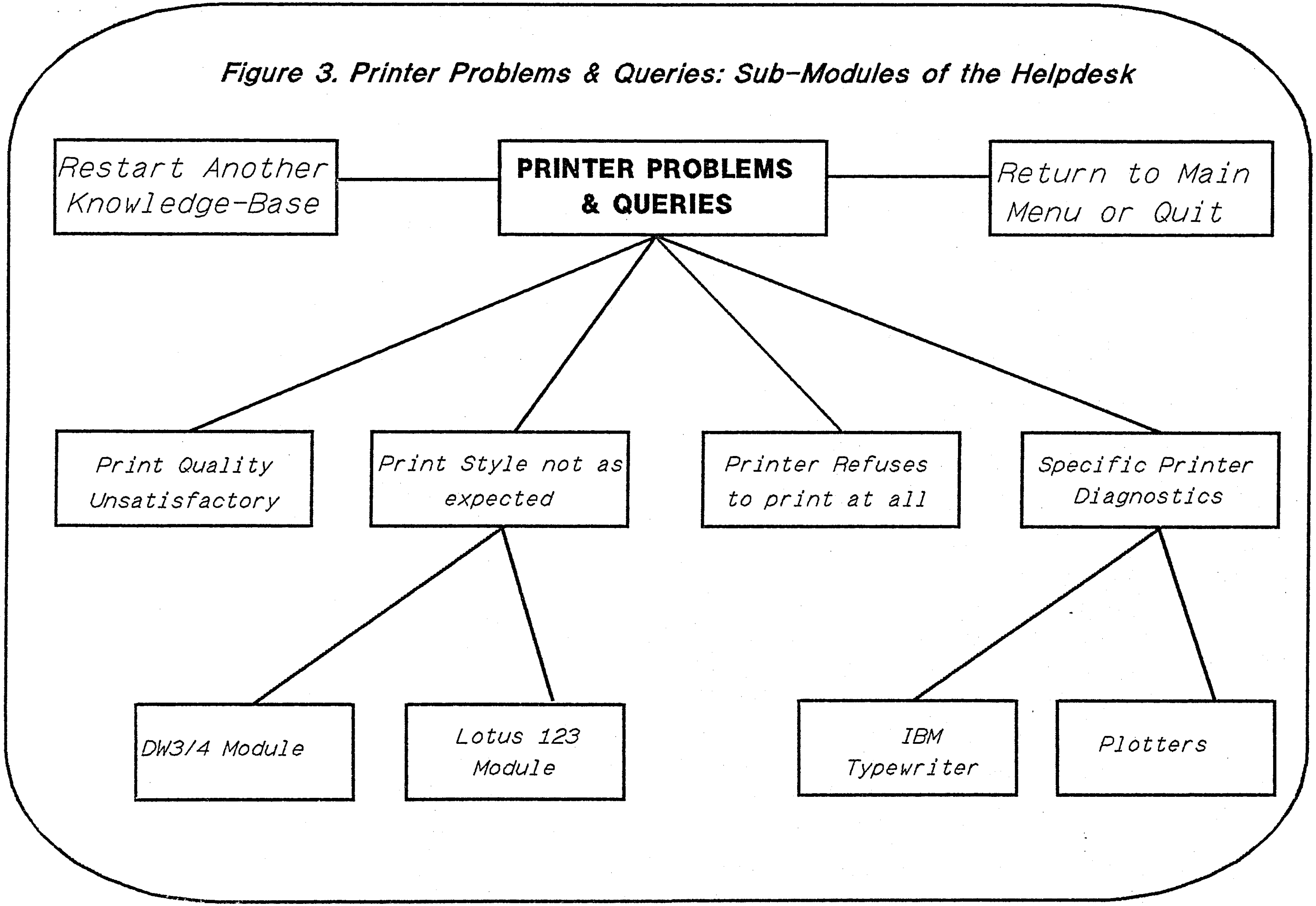


Figure 3. Printer Problems & Queries: Sub-Modules of the Helpdesk

Figure 3.1 Print Quality Unsatisfactory: Sub-Module of Printer problems

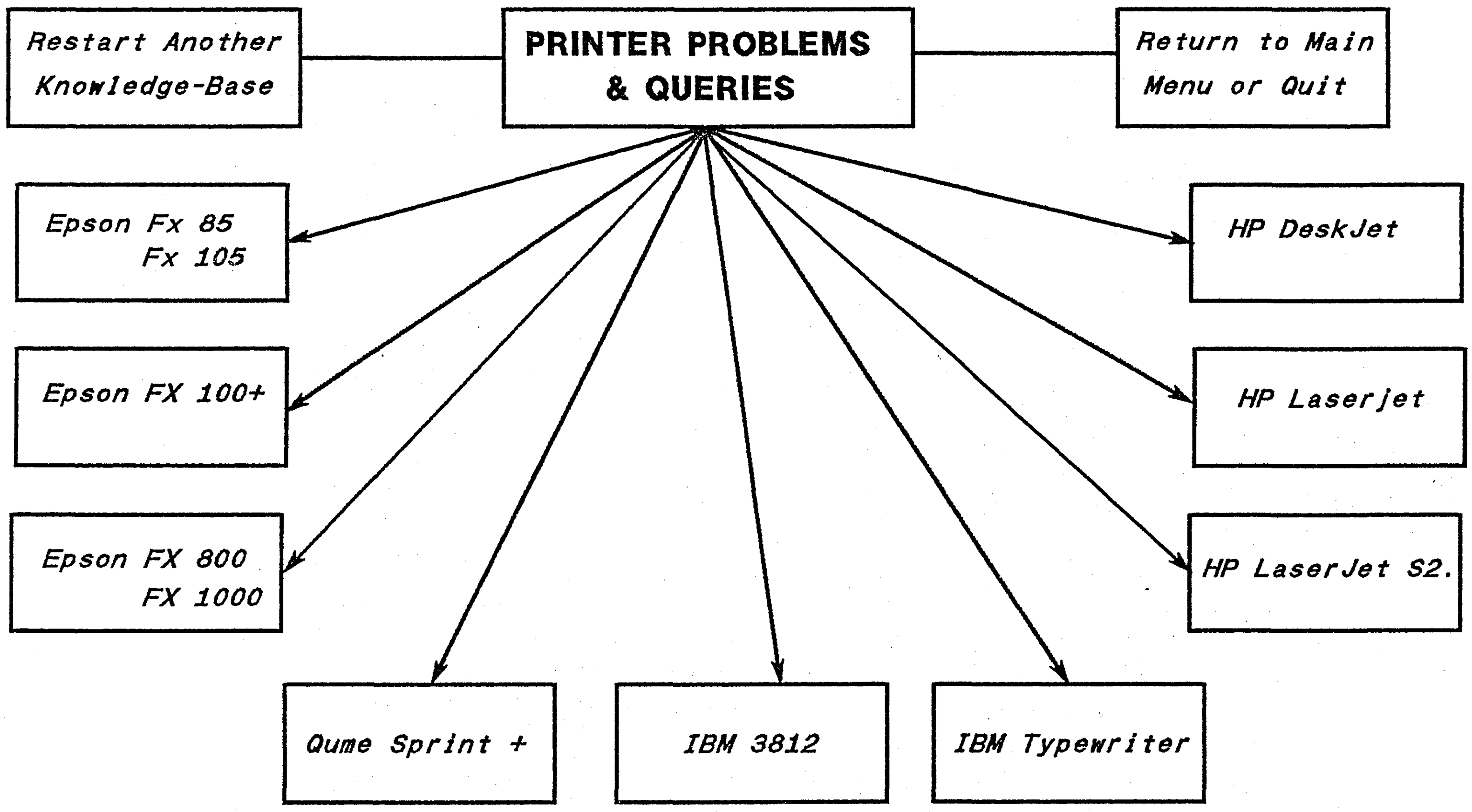
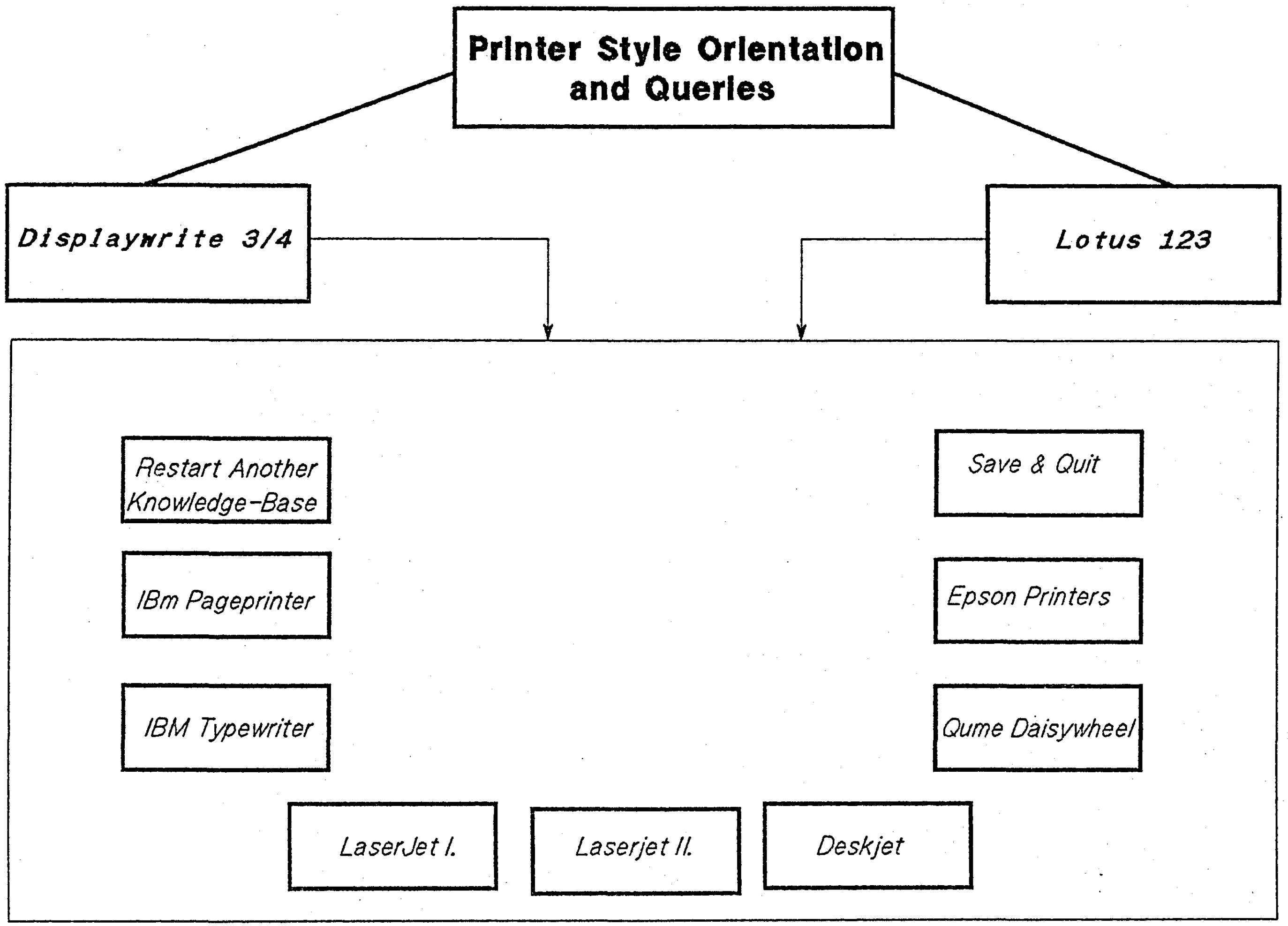


Figure 3.2 Print Style Not as Expected: Sub-Module of Printer Problems



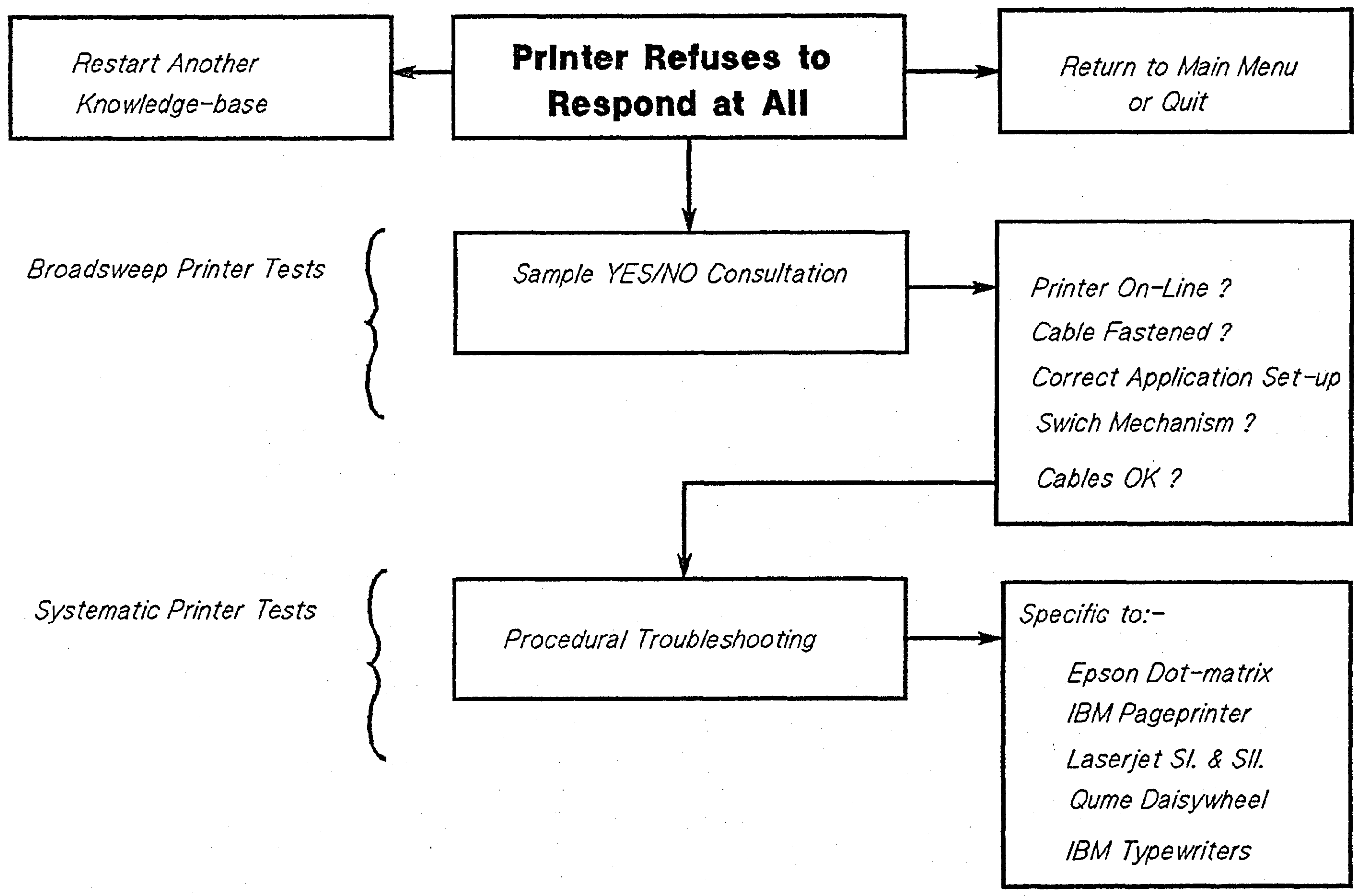


Figure 3.3 Printer Refuses to Respond at All: Sub-Module of Printer Problems

Figure 3.4 Specific Printer Diagnostics: Sub-module of Printer Problems

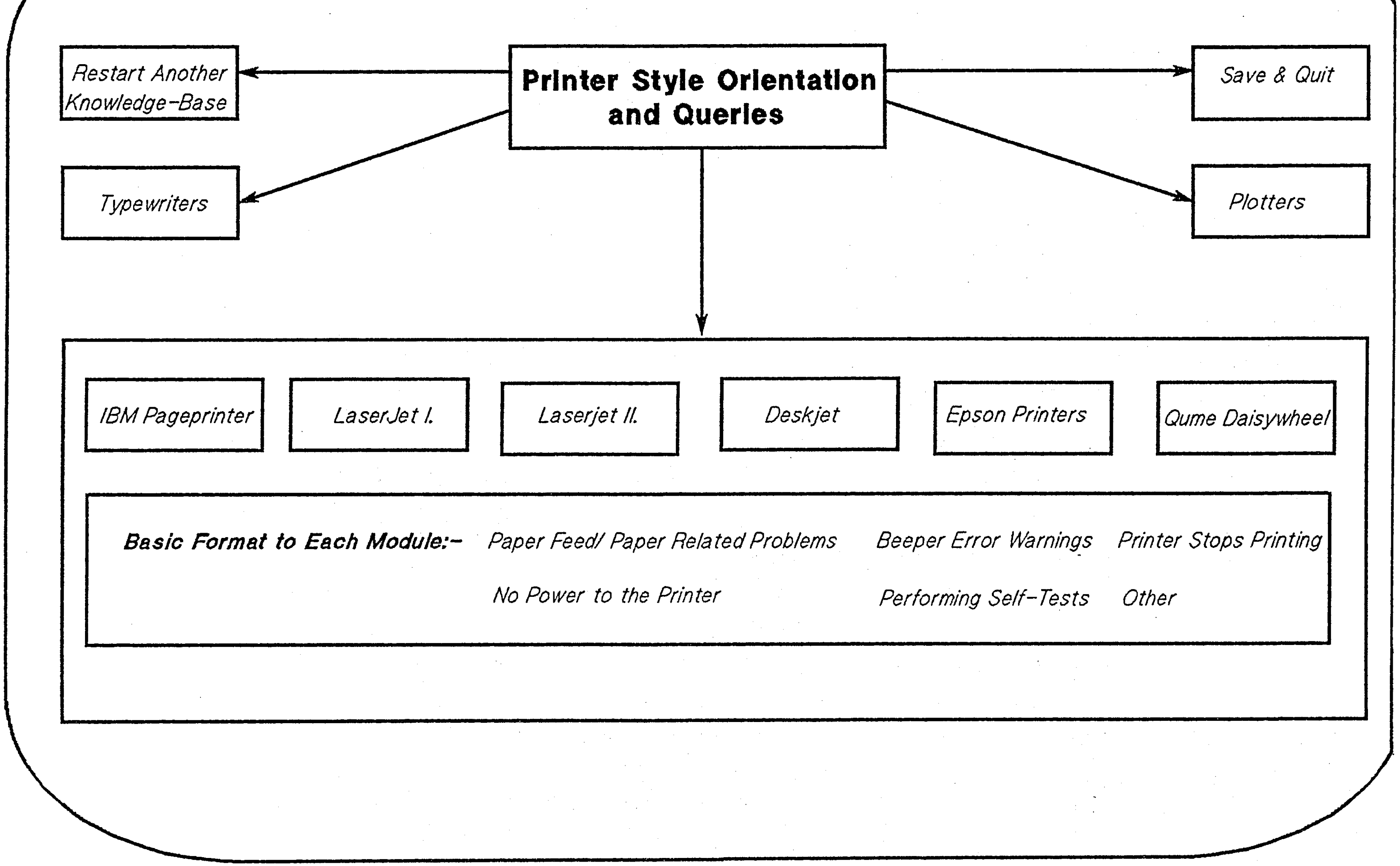


Figure 4: PC Hardware Problems & Queries

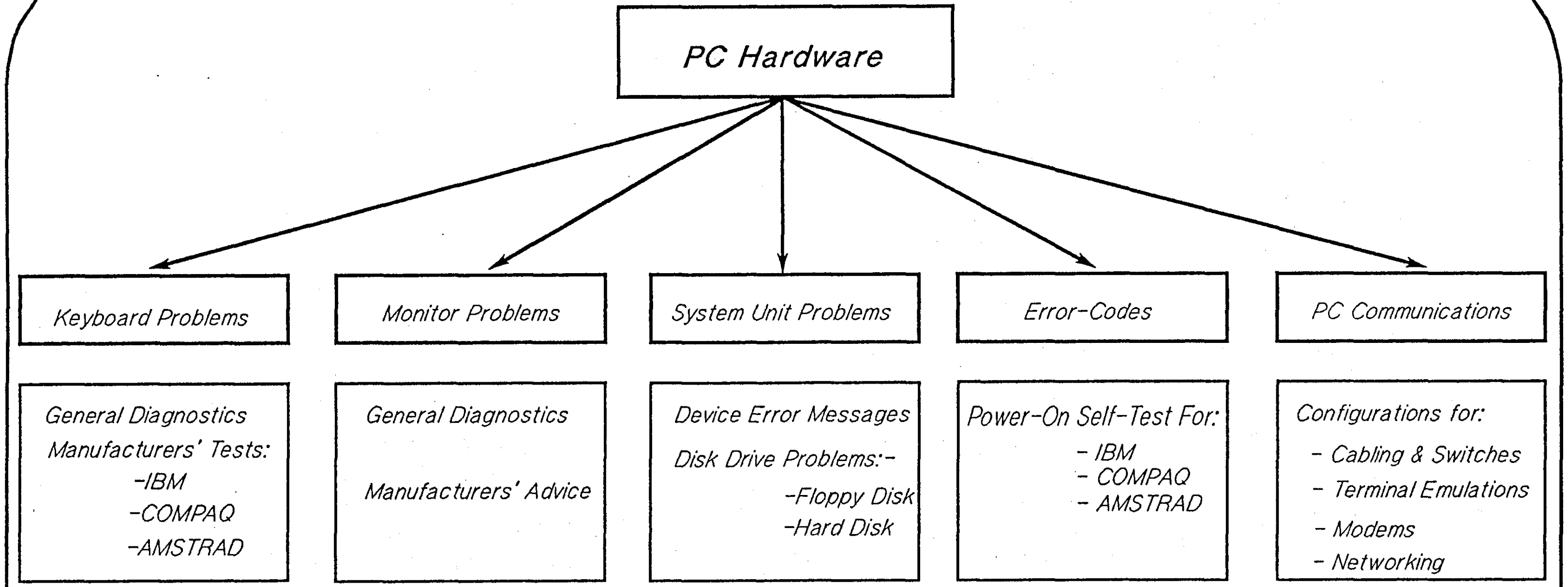


Figure 5: DECMATE II Troubleshooting Module

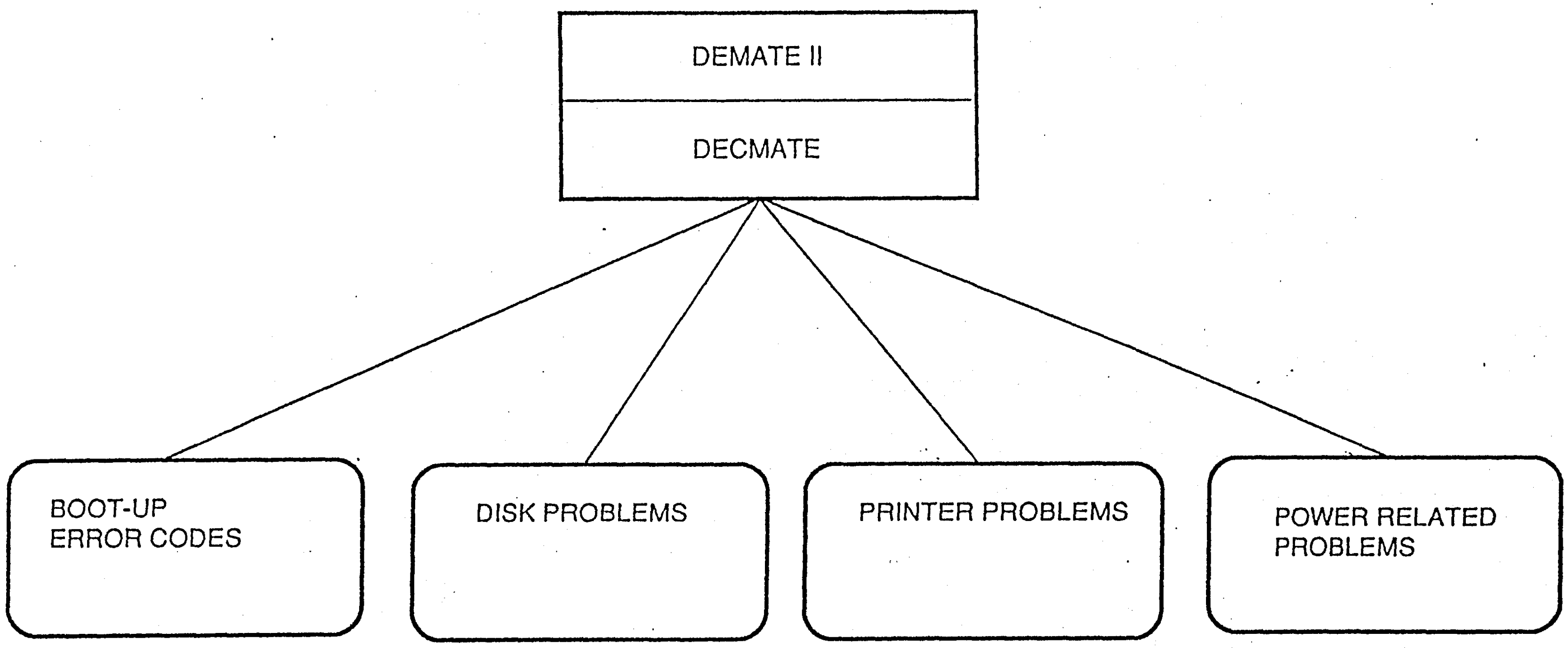


Figure 6: Systems Profile

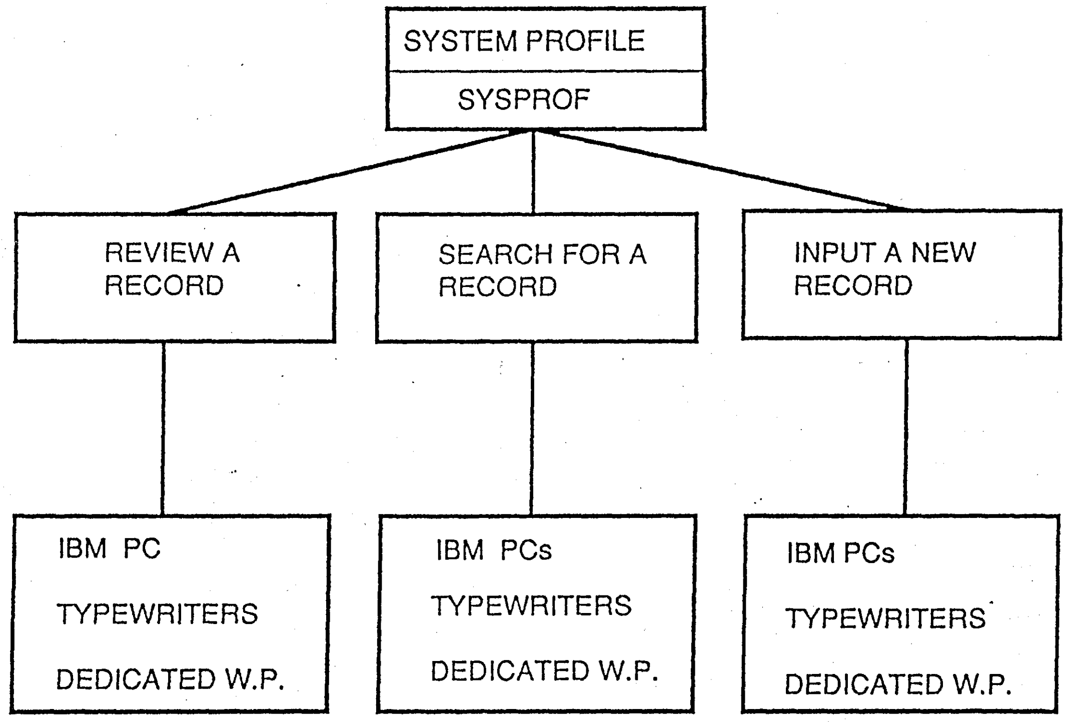
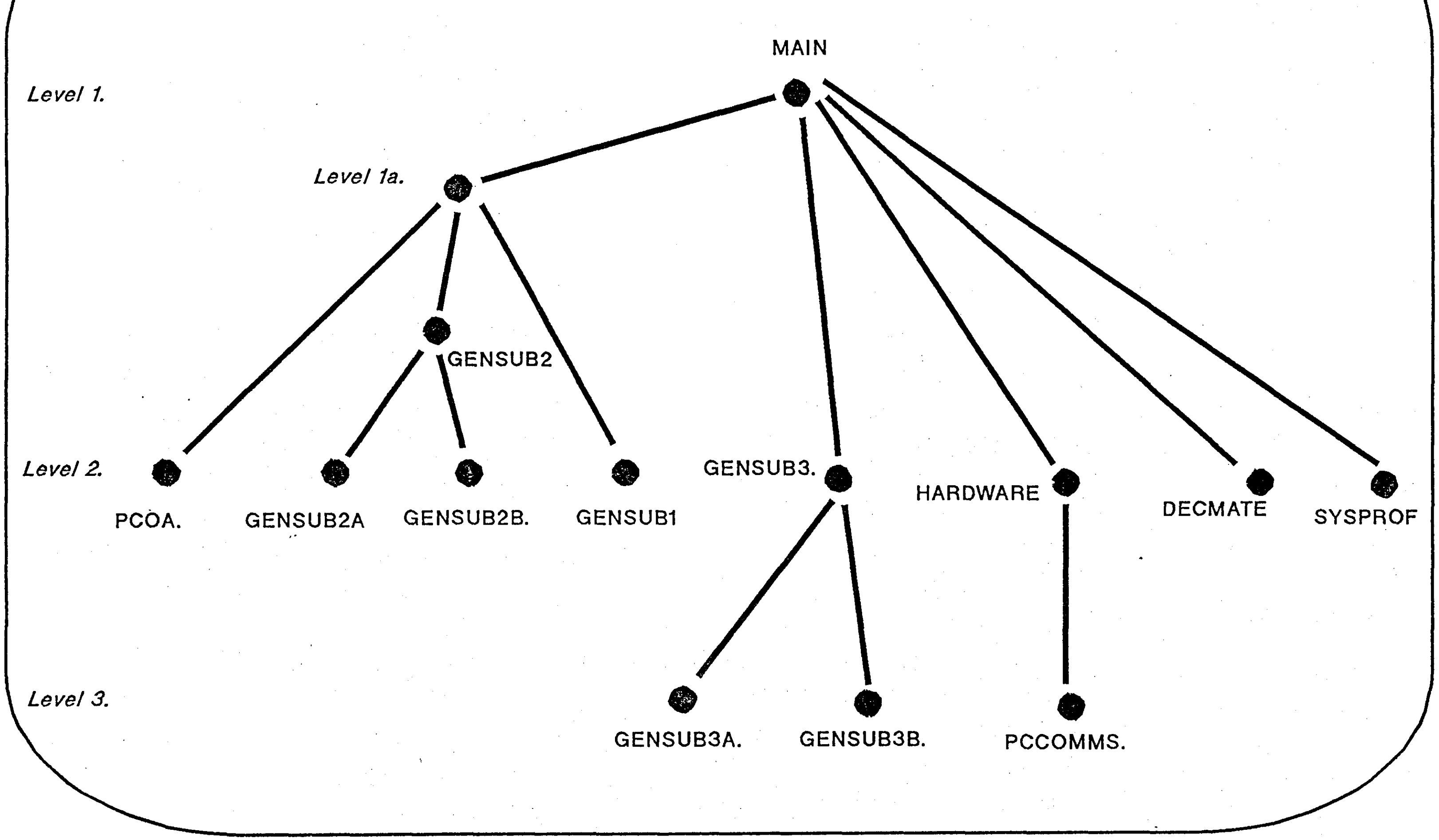


Figure 7: Knowledge Base Modular Design



Appendix XIV: Part B.

Rule & Variable Listings for the Office Systems Helpdesk

```
*
*  RULE DICTIONARY                  Wed Mar 28 14:33:26 1990  Page:
*  ---------------
  ♦ *yes/nol                                              Sp
  ♦ Auto option has been chosen                        Ex Sp
  ♦ check correct port                                 Ex Sp
  ♦ check device control for PC                        Ex Sp
  ♦ check switch to printer link                       Ex Sp
  ♦ check the manual / auto switch                     Ex Sp
    DBASE3 OPTION FROM OTHERS                          Ex Sp
  ♦ did this work                                      Ex Sp
    displaywrite 3 option                              Ex Sp
    displaywrite 370 option                            Ex Sp
    displaywrite4 option                               Ex Sp
  ♦ Do top-level check of the software                 Ex Sp
  ♦ DOS OPTION FROM OTHERS                             Ex Sp
    e2$                                                Ex Sp
    ER$                                                Ex Sp
  ♦ error messages LJ2                                 Ex Sp
  ♦ Exhausted option therefore try systematic tests    Ex Sp
  ♦ fx100 beeper error warnings                        Ex Sp
  ♦ go into user profile to look at printer type       Ex Sp
  ♦ Has error code & consequent action solved problem? Ex Sp
  ♦ instructions on how to specify the printer         Ex Sp
  ♦ laserjet error codes                               Ex Sp
  ♦ LED light on                                       Ex Sp
    LEX86 OPTION FROM OTHERS                           Ex Sp
  ♦ look closer at the PC                              Ex Sp
  ♦ LOTUS 123 option                                   Ex Sp
  ♦ Manual option has been chosen                      Ex Sp
  ♦ manual or auto option                              Ex Sp
  ♦ manual switch correct                              Ex Sp
  ♦ manual switch incorrect                            Ex Sp
  ♦ others option                                      Ex Sp
    plop                                               Ex Sp
  ♦ Possible application software problem              Ex Sp
  ♦ problem persist3                                   Ex Sp
  ♦ problem persists2                                  Ex Sp
  ♦ problem prevails                                   Ex Sp
  ♦ remove switch and test                             Ex Sp
  ♦ RETURN TO GENERAL TROUBLESHOOTER                   Ex Sp
  ♦ RETURN TO REST OF CONSULTATION-GENSUB1             Ex Sp
  ♦ switch printer off                                 Ex Sp
  ♦ test beep error warnings option                    Ex Sp
  ♦ test blinking error problems                       Ex Sp
  ♦ test for error message                             Ex Sp
  ♦ TEST FX85/105 BEEPER ERROR WARNINGS                Ex Sp
  ♦ test manual or auto option                         Ex Sp
  ♦ test print from the operating system               Ex Sp
  ♦ test that cable between printer & switch is placed Ex Sp
  ♦ TEST THE ERROR CODE                                Ex Sp
  ♦ test the error option for the laserjetII menu      Ex Sp
  ♦ test the fx100 buzzer menu.                        Ex Sp
  ♦ test the menu options for the fx85 / fx105         Ex Sp
  ♦ TEST THE SECONDARY APPLICATION SOFTWARE OPTION.    Ex Sp
  ♦ test the software option of the IBBM PC            Ex Sp
  ♦ test where to go back to-gensub1                   Ex Sp
  ♦ TEST WHETHER IN LOTUS 123 OR DW3/4-GENSUB1         Ex Sp
  ♦ using software correctly?                          Ex Sp
  ♦ what to do now?                                    Ex Sp
  ♦ yes, the printer does make an error response       Ex Sp
```

```
*
*  RULE DICTIONARY                      Wed Mar 28 14:27:44 1990  Page:
*  ---------------
  ♦ ACCY VARIABLE SCREEN                                  Sp
  ♦ carry on with qume/dw3/4 diagnostics
  ♦ carry on2-dw3/4                                    Ex Sp
  ♦ DESKJET LANDSCAPE CHOSEN                              Sp
  ♦ DESKJET PORTRAIT CHOSEN                               Sp
  ♦ DESKJET SUPPORT TESTS-dw3/4                        Ex Sp
  ♦ DISPLAYWRITE 3/4 CHARACTER SET                     Ex Sp
    DJ$
  ♦ DJ:DW3/4: WHAT TRYING TO DO-1                         Sp
  ♦ DJ:DW3/4: WHAT TRYING TO DO-2                         Sp
  ♦ DJ:DW3/4: WHAT TRYING TO DO-3                         Sp
  ♦ DJ:DW3/4: WHAT TRYING TO DO-4
  ♦ DJ:TEST WHAT DOING IN LANDSCAPE2-DW3/4                Sp
  ♦ DJ:WHAT DOING LSCAPE1-DW3/4                           Sp
  ♦ DJ:WHAT DOING LSCAPE2-DW3/4                           Sp
  ♦ DON'T KNOW- IE Y CARTRIDGE LIKELY                  Ex Sp
  ♦ DOT MATRIX SUPPORT TESTS-dw3/4                     Ex Sp
  ♦ FONT$ VARIABLE SCREEN                              Ex
  ♦ IBM PAGEPRINTER SUPPORT TESTS-dw3/4                Ex Sp
  ♦ IBM TYPEWRITER SUPPORT TESTS-dw3/4                 Ex Sp
  ♦ L CARTRIDGE                                        Ex Sp
  ♦ L landscape chosen                                 Ex Sp
  ♦ L portrait chosen                                  Ex Sp
  ♦ landscape chosen                                   Ex Sp
  ♦ LASERJET SUPPORT TESTS-dw3/4                       Ex Sp
  ♦ LASERJETII SUPPORT TESTS-dw3/4                     Ex Sp
  ♦ look at fx1000                                        Sp
  ♦ LOOK AT FX1000DW34                                    Sp
  ♦ LOOK AT FX85                                          Sp
  ♦ LOOK AT FX85DW34                                   Ex Sp
  ♦ M LANDSCAPE CHOSEN-DW3/4                           Ex Sp
  ♦ M PORTRAIT CHOSEN-DW3/4                            Ex Sp
    OPTIONPC$ VARIABLE SCREEN
  ♦ PCMAIN$ VARIABLE SCREEN                            Ex
  ♦ portrait chosen                                    Ex Sp
  ♦ PSDW: TEST WHICH PRINTER YOU TRYING TO USE         Ex Sp
  ♦ PSDW:DESKJET SUPPORT TESTS-DW3/4                      Sp
  ♦ PSDW:DOT MATRIX SUPPORT TESTS-DW3/4                Ex Sp
  ♦ PSDW:IBM PAGEPRINTER SUPPORT TEST-DW3/4               Sp
  ♦ PSDW:IBM TYPEWRITER SUPPORT TESTS-DW3/4            Ex Sp
  ♦ PSDW:LASERJET SUPPORT TESTS-DW3/4
  ♦ PSDW:LASERJETII SUPPORT TESTS-DW3/4                Ex Sp
  ♦ PSDW:QUME SUPPORT TESTS-DW3/4                         Sp
  ♦ Qume /dw34 orientation problem                     Ex Sp
  ♦ Qume /Lotus 123 orientation problem                Ex Sp
  ♦ QUME SUPPORT TESTS-dw3/4                           Ex Sp
  ♦ return to main menu                                   Sp
  ♦ return to the PC local menu                           Sp
  ♦ RETURN TO THE REST OF THE CONSULTATION-GENSUB2A    Ex Sp
  ♦ right size of font- dw3/4                             Sp
  ♦ save file routine                                     Sp
    SOFT$ VARIABLE SCREEN
  ♦ TEST DJ- WHAT DOING-2                                 Sp
  ♦ TEST DJ- WHAT DOING-3                                 Sp
  ♦ TEST DJ-WHAT DOING -4                                 Sp
  ♦ TEST DJ-WHAT DOING 1                                  Sp
  ♦ TEST LASERJET FONT CARTRIDGE TYPE                     Sp
  ♦ TEST THE APPLE QUIT OPTION                            Sp
  ♦ test the dw3/4 save file routine option               Sp
  ♦ TEST THE FIFTH PAGE OPTION-IBM PC                     Sp
```

```
*
*  RULE DICTIONARY                     Wed Mar 28 14:27:47 1990  Page:
*  ---------------
  ♦ TEST THE FIFTH PAGE OPTION-LICS                      Sp
  ♦ TEST THE FIFTH PAGE OPTION-LICS WITH PAGEPRINTER     Sp
  ♦ TEST THE FIFTH PAGE OPTION-LICS WITH QUME            Sp
  ♦ TEST THE FIFTH PAGE OPTION-QUME PRINTER DW3/4     Ex Sp
  ♦ TEST THE FIFTH PAGE OPTION-ROMAN                     Sp
  ♦ TEST THE FIRST PAGE OPTION-IBM PC                    Sp
  ♦ TEST THE FIRST PAGE OPTION-IBM TYPEWRITER         Ex Sp
  ♦ TEST THE FIRST PAGE OPTION-LICS                      Sp
  ♦ TEST THE FIRST PAGE OPTION-LICS WITH PAGEPRINTER     Sp
  ♦ TEST THE FIRST PAGE OPTION-LICS WITH QUME            Sp
  ♦ TEST THE FIRST PAGE OPTION-QUME PRINTER DW3/4     Ex Sp
  ♦ TEST THE FIRST PAGE OPTION-ROMAN                     Sp
  ♦ TEST THE FOURTH PAGE OPTION-IBM PC                   Sp
  ♦ TEST THE FOURTH PAGE OPTION-IBM TYPEWRITER        Ex
  ♦ TEST THE FOURTH PAGE OPTION-LICS                     Sp
  ♦ TEST THE FOURTH PAGE OPTION-LICS WITH PAGEPRINTER    Sp
  ♦ TEST THE FOURTH PAGE OPTION-LICS WITH QUME           Sp
  ♦ TEST THE FOURTH PAGE OPTION-QUME PRINTER DW3/4    Ex Sp
  ♦ TEST THE FOURTH PAGE OPTION-ROMAN                    Sp
  ♦ test the options to wipe with special                Sp
  ♦ TEST THE PAGEPRINTER MENU OPTION-DW3/DW4          Ex Sp
  ♦ test the save file routine option                    Sp
  ♦ TEST THE SECOND PAGE OPTION- IBM TYPEWRITER       Ex Sp
  ♦ TEST THE SECOND PAGE OPTION- QUME PRINTER DW3/4   Ex Sp
  ♦ TEST THE SECOND PAGE OPTION-IBM PC                   Sp
  ♦ TEST THE SECOND PAGE OPTION-LICS                     Sp
  ♦ TEST THE SECOND PAGE OPTION-LICS WITH PAGEPRINTER    Sp
  ♦ TEST THE SECOND PAGE OPTION-LICS WITH QUME           Sp
  ♦ TEST THE SECOND PAGE OPTION-ROMAN                    Sp
  ♦ TEST THE THIRD PAGE OPTION-IBM PC                    Sp
  ♦ TEST THE THIRD PAGE OPTION-IBM TYPEWRITER         Ex Sp
  ♦ TEST THE THIRD PAGE OPTION-LICS                      Sp
  ♦ TEST THE THIRD PAGE OPTION-LICS WITH PAGEPRINTER     Sp
  ♦ TEST THE THIRD PAGE OPTION-LICS WITH QUME            Sp
  ♦ TEST THE THIRD PAGE OPTION-QUME PRINTER DW3/4     Ex Sp
  ♦ TEST THE THIRD PAGE OPTION-ROMAN                     Sp
  ♦ TEST WHAT DOING IN DESKJET LANDSCAPE 1-dw3/4      Ex Sp
  ♦ test what doing in landscape 1-dw3/4                 Sp
  ♦ Test what doing in landscape L1                      Sp
  ♦ test what doing in landscape L2                      Sp
  ♦ test what doing in landscape M1-lotus                Sp
  ♦ test what doing in landscape M2-lotus                Sp
  ♦ test what doing in landscape M3-LOTUS                Sp
  ♦ test what doing in landscape2-dw3/4                  Sp
  ♦ test what trying to do -one                          Sp
  ♦ test what trying to do -three
  ♦ test what trying to do- two                          Sp
  ♦ test what trying to do-L1                            Sp
  ♦ test what trying to do-L2                            Sp
  ♦ test what trying to do-L3                            Sp
  ♦ test what trying to do-M1DW3/4                       Sp
  ♦ test what trying to do-M2                            Sp
  ♦ test what trying to do-M2DW3/4                       Sp
  ♦ test what trying to do-M3                            Sp
  ♦ test what trying to do-M3DW3/4                       Sp
  ♦ test where to go back to-GENSUB2A                 Ex Sp
  ♦ TEST WHETHER DESKJET IN PORTRAIT OR LANDSCAPE        Sp
  ♦ Test whether in portrait or landscape                Sp
  ♦ Test whether L cartridge in portrait or landscape    Sp
  ♦ Test whether M cartridge is in portrait or l'scape   Sp
```

```
*
*  RULE DICTIONARY                      Wed Mar 28 14:27:52 1990  Page:
*  ---------------
  ♦ TEST WHETHER TO VIEW IBM PC CHARACTER SET-DESKJET     Sp
  ♦ TEST WHETHER TO VIEW IBM/LICS CHARACTER SET           Sp
  ♦ TEST WHETHER TO VIEW LICS CHARACTER SET OR END        Sp
  ♦ TEST WHETHER TO VIEW ROMAN-8 CHARACTER SET OR END     Sp
  ♦ test which Deskjet portrait style has been chosen     Sp
  ♦ test which dw3/4 variables to wipe                    Sp
  ♦ test which L portrait typestle used                   Sp
  ♦ test which landscape type style is used-DW/3/4        Sp
  ♦ test which landscape type style used for L cart.      Sp
  ♦ test which landscape type style used for M cart.      Sp
  ♦ TEST WHICH LSCAPE TYPESTYLE IS USED WITH DESKJET
  ♦ TEST WHICH M LANDSCAPE TYPESTYLE CHOSEN-DW3/4         Sp
  ♦ TEST WHICH M PORTRAIT TYPESTYLE CHOSEN-DW3/4          Sp
  ♦ TEST WHICH OPTION TO WIPE                             Sp
  ♦ test which option to wipe-dw3/4                       Sp
  ♦ test which portrait typestyle has been used           Sp
  ♦ TEST WHICH PRINTER YOU ARE ATTEMPTING TO USE       Ex Sp
  ♦ test which tw/dw3 typestyle                        Ex Sp
  ♦ TEST WHICH WAY TO GO-DW3
  ♦ turn to general troubleshooter                        Sp
  ♦ VIEW IBM PC CHARACTER SET                             Sp
  ♦ VIEW IBM TYPEWRITER CHARACTER SET-DW3/4               Sp
  ♦ VIEW LICS CHARACTER SET                               Sp
  ♦ VIEW LICS CHARACTER SET AGAINST THE IBM CHAR. SET     Sp
  ♦ VIEW QUME BILINGUAL CHARACTER SET-DW3/4            Ex Sp
  ♦ VIEW QUME BILINGUAL CHARACTER SET-LOTUS 123           Sp
  ♦ VIEW ROMAN-8 CHARACTER SET                            Sp
  ♦ what are you trying to do- one?                       Sp
  ♦ what are you trying to do- three?                     Sp
  ♦ what are you trying to do- two?                       Sp
  ♦ what doing landscape 1-dw3/4                       Ex Sp
  ♦ what doing landscape 2-dw3/4                          Sp
  ♦ what doing landscape L1                               Sp
  ♦ what doing landscape L2                               Sp
  ♦ what doing landscape M1                               Sp
  ♦ what doing landscape M2                               Sp
  ♦ what doing landscape M3                               Sp
  ♦ what doing M landscape with dw3/4-M1                  Sp
  ♦ what doing M landscape with dw3/4-M2                  Sp
  ♦ what doing M landscape with dw3/4-M3                  Sp
  ♦ what to do now-dw3/4?                              Ex Sp
  ♦ what trying to do with L cartridge-1                  Sp
  ♦ what trying to do with L cartridge-2                  Sp
  ♦ what trying to do with L cartridge-3                  Sp
  ♦ what trying to do with M cartridge with dw3/4-1
  ♦ what trying to do with M cartridge with dw3/4-2
  ♦ what trying to do with M cartridge with dw3/4-3       Sp
  ♦ what trying to do with M cartridge-1                  Sp
  ♦ what trying to do with M cartridge-2                  Sp
  ♦ what trying to do with M cartridge-3                  Sp
  ♦ Y CARTRIDGE                                        Ex Sp
    ynd
  ♦ ZLIB$ VARIABLE SCREEN                              Ex
```

```
*
*  RULE DICTIONARY                       Wed Mar 28 14:22:42 1990  Page:
*  ---------------
  ♦ carry on with qume/lotus123 diagnostics                  Sp
  ♦ carry on2-lotus 123                                      Sp
    CQ123$
  ♦ DON'T KNOW 123- I.E. Y CARTRIDGE LIKELY                  Sp
    end$
    EP123$
  ♦ FONT$ VARIABLE SCREEN LJETI WITH 123
  ♦ L-LJ1-L123: Landscape chosen                          Ex Sp
  ♦ L-LJ1-L123: Portrait chosen                           Ex Sp
  ♦ L-LJ1-L123:Test what doing in landscape L1               Sp
  ♦ L-LJ1-L123:test what doing in landscape L2               Sp
  ♦ L-LJ1-L123:test what trying to do-L1                     Sp
  ♦ L-LJ1-L123:Test what trying to do-L2                     Sp
  ♦ L-LJ1-L123:test what trying to do-L3                     Sp
  ♦ L-LJ1-L123:Test which L portrait typestyle used          Sp
  ♦ L-LJ1-L123:test which landscape type style used          Sp
  ♦ L-LJ1-L123:what doing landscape L1                       Sp
  ♦ L-LJ1-L123:what doing landscape L2                       Sp
  ♦ L-LJ1-L123:what trying to do with L cartridge-1          Sp
  ♦ L-LJ1-L123:what trying to do with L cartridge-2          Sp
  ♦ L-LJ1-L123:what trying to do with L cartridge-3          Sp
  ♦ L-LJ1:123 CARTRIDGE                                      Sp
  ♦ L123: M LANDSCAPE CHOSEN-LII                             Sp
  ♦ L123: M PORTRAIT CHOSEN-LII                              Sp
    LCQTY$
    LEC$
  ♦ LOOK AT FX1000-LOTUS 123                                 Sp
  ♦ LOOK AT FX85-LOTUS 123                                   Sp
  ♦ LOTUS 123 CHARACTER SET                               Ex Sp
  ♦ Qume /Lotus 123 orientation problem                   Ex Sp
  ♦ RETURN TO THE REST OF THE CONSULTATION-GENSUB2B          Sp
  ♦ right size of font-lotus 123                             Sp
  ♦ TEST 123 WITH LJETI FOR CARTRIDGE TYPE                Ex Sp
  ♦ Test Epson printers with Lotus 123                       Sp
  ♦ test IBM typewriter with Lotus                           Sp
  ♦ Test Laserjet I with Lotus 123                           Sp
  ♦ TEST LASERJET II WITH M CARTRIDGE USING LOTUS 123        Sp
  ♦ test pageprinter with Lotus 123                          Sp
  ♦ Test qume printer with Lotus 123                         Sp
  ♦ TEST THE FIFTH PAGE OPTION-LICS WITH IBM TYPEW.          Sp
  ♦ TEST THE FIFTH PAGE OPTION-LICS WITH PAGEPRINTER         Sp
  ♦ TEST THE FIFTH PAGE OPTION-LICS WITH QUME                Sp
  ♦ TEST THE FIFTH PAGE OPTION-LICS WITH ROMAN-8             Sp
  ♦ TEST THE FIRST PAGE OPTION-LICS WITH IBM TYPEW.       Ex Sp
  ♦ TEST THE FIRST PAGE OPTION-LICS WITH PAGEPRINTER         Sp
  ♦ TEST THE FIRST PAGE OPTION-LICS WITH QUME                Sp
  ♦ TEST THE FIRST PAGE OPTION-LICS WITH ROMAN-8             Sp
  ♦ TEST THE FOURTH PAGE OPTION-LICS WITH IBM TYPEW.         Sp
  ♦ TEST THE FOURTH PAGE OPTION-LICS WITH PAGEPRINTER        Sp
  ♦ TEST THE FOURTH PAGE OPTION-LICS WITH QUME               Sp
  ♦ TEST THE FOURTH PAGE OPTION-LICS WITH ROMAN-8            Sp
  ♦ test the pageprinter menu option-LOTUS 123               Sp
  ♦ TEST THE SECOND PAGE OPTION-LICS WITH IBM TYPEW.      Ex Sp
  ♦ TEST THE SECOND PAGE OPTION-LICS WITH PAGEPRINTER        Sp
  ♦ TEST THE SECOND PAGE OPTION-LICS WITH QUME               Sp
  ♦ TEST THE SECOND PAGE OPTION-LICS WITH ROMAN-8            Sp
  ♦ TEST THE THIRD PAGE OPTION-LICS WITH IBM TYPEW.       Ex Sp
  ♦ TEST THE THIRD PAGE OPTION-LICS WITH PAGEPRINTER         Sp
  ♦ TEST THE THIRD PAGE OPTION-LICS WITH QUME                Sp
  ♦ TEST THE THIRD PAGE OPTION-LICS WITH ROMAN-8             Sp
```

```
*
*  RULE DICTIONARY                      Wed Mar 28 14:22:46 1990  Page:
*  ---------------
  ♦ TEST WHAT DOING IN LANDSCAPE M2-LII                   Sp
  ♦ TEST WHAT DOING IN LANDSCAPE M3-LII                   Sp
  ♦ TEST WHAT TO DO-LOTUS 123                             Sp
  ♦ test what trying to do-M1LII                          Sp
  ♦ test what trying to do-M2LII                          Sp
  ♦ test what trying to do-M3LII                          Sp
  ♦ test what trying to do-Y-LJ1:L123 ONE                 Sp
  ♦ test what trying to do-Y-LJ1:L123 THREE               Sp
  ♦ test what trying to do-Y-LJ1:L123 TWO                 Sp
  ♦ test where to go back-GENSUB2B                     Ex Sp
  ♦ Test whether L cartridge in portrait or landscape     Sp
  ♦ TEST WHETHER M CARTRIDGE IN P OR L- LOTUS 123         Sp
  ♦ TEST WHETHER TO VIEW IBM/LICS CHARACTER SET           Sp
  ♦ TEST WHETHER TO VIEW IBM/LICS CHARACTER SET-EPSON     Sp
  ♦ TEST WHETHER TO VIEW ROMAN-8 WITH LICS OR END         Sp
  ♦ TEST WHICH LANDSCAPE STYLE CHOSEN-LII                 Sp
  ♦ TEST WHICH M PORTRAIT TYPESTYLE CHOSEN-LII            Sp
  ♦ TEST WHICH WAY TO GO-LOTUS 123
  ♦ VIEW IBM TYPEWRITER CHARACTER SET-LOTUS 123           Sp
  ♦ VIEW LICS CHARACTER SET AGAINST ROMAN-8 CHAR. SET     Sp
  ♦ VIEW LICS CHARACTER SET AGAINST THE IBM CHAR. SET     Sp
  ♦ VIEW QUME BILINGUAL CHARACTER SET-LOTUS 123           Sp
  ♦ WHAT DOING IN LANDSCAPE M1-LII                        Sp
  ♦ WHAT DOING IN LANDSCAPE M2-LII                        Sp
  ♦ WHAT DOING IN LANDSCAPE M3-LII                        Sp
  ♦ what trying to do with M cartridge-1LII               Sp
  ♦ what trying to do with M cartridge-2LII               Sp
  ♦ what trying to do with M cartridge-3LII               Sp
  ♦ what trying to do-Y-LJ1:L123 ONE                      Sp
  ♦ what trying to do-Y-LJ1:L123 THREE                    Sp
  ♦ what trying to do-Y-LJ1:L123 TWO                      Sp
  ♦ Y-LJ1:123 CARTRIDGE                                   Sp
  ♦ Y-LJ1:LANDSCAPE 123 CHOSEN                            Sp
  ♦ Y-LJ1:PORTRAIT 123 CHOSEN                             Sp
  ♦ Y-LJ1:Test what doing in landscape 1-lotus 123        Sp
  ♦ Y-LJ1:TEST WHAT DOING IN LANDSCAPE 2- LOTUS 123       Sp
  ♦ Y-LJ1:TEST WHETHER IN P OR L 123                      Sp
  ♦ Y-LJ1:Test which landscape typestyle is used-LOTUS    Sp
  ♦ Y-LJ1:TEST WHICH P 123 TYPESTYLE USED                 Sp
  ♦ Y-LJ1:what doing landscape 1-lotus 123                Sp
  ♦ Y-LJ1:what doing landscape 2-lotus 123                Sp
```

```
*
*  RULE DICTIONARY                      Wed Mar 28 14:18:12 1990  Page:
*  ---------------
  ♦ #problem persists?
  ♦ @beeper error warnings
  ♦ @computer hangs up when trying to print               Sp
  ♦ @does problem still exist1
  ♦ @does problem still exist4
  ♦ @does the problem still exist2                     Ex
  ♦ @does the problem still exist3
  ♦ @does the problem still exist5
  ♦ @does the problem still exist6
  ♦ @does the problem still exist7
  ♦ @error messages LJ2                                   Sp
  ♦ @fx100 beeper error warnings
  ♦ @general troubleshooting for laserjetII
  ♦ @graphics problems
  ♦ @HP laserjet main menu                                Sp
  ♦ @laserjet error codes                                 Sp
  ♦ @laserjet self-tests
  ♦ @laserjetII yes/no2
  ♦ @LaserjetII yes/no3
  ♦ @LaserjetII yes/no4
  ♦ @LaserjetII yes/no5
  ♦ @LaserjetII yes/no6
  ♦ @LaserjetII yes/no7
  ♦ @PAPER FEEDING PROBLEMS
  ♦ @paper jam problems with LaserjetII
  ♦ @power problems with the laserjet
  ♦ @print changes during printing
  ♦ @print quality problems
  ♦ @PRINTER DOES NOT PRINT                               Sp
  ♦ @printer fails to print with no reason
  ♦ @Printing is garbled
  ♦ @Problem remain1
  ♦ @problem remain2
  ♦ @problem remain3
  ♦ @problem remain4
  ♦ @problem remain5
  ♦ @problem still exist?
  ♦ @Problems with the papar-end defector
  ♦ @Qume diagnostics with sheet feeder
  ♦ @Qume diagnostics without sheet feeder
  ♦ @Qume Sprint 11 Plus(11/40) systematic tests
  ♦ @resort to the self tests
  ♦ @self test for the fx100
  ♦ @still losing data
  ♦ @Tabbing problems with printer
  ♦ @test beep error warnings option
  ♦ @test blinking error problems
  ♦ @test commonly recurring problems
  ♦ @test for printing quality problems
  ♦ @test for the print quality problem with LjetII.
  ♦ @test fx800 / fx1000 printer options
  ♦ @test garbled printing option
  ♦ @test graphics optionsd
  ♦ @test hp deskjet main menu options
  ♦ @test if we can refer to error section
  ♦ @Test Laserjet series two main menu options
  ♦ @test menu not printing
  ♦ @test paper feeding options
  ♦ @test paper feeding problems
  ♦ @test paper related problems
```

```
*
*  RULE DICTIONARY                      Wed Mar 28 14:18:15 1990  Page:
*  ---------------
  ♦ @test print not what you expect
  ♦ @test print quality options
  ♦ @test printer does not print options               Ex Sp
  ♦ @test problems when programming graphics etc
  ♦ @test qume diagnostic options with sheet feeder
  ♦ @test selecType problems
  ♦ @test tabbing & width problems
  ♦ @test the beep error options
  ♦ @test the computer hangs option                       Sp
  ♦ @TEST THE ERROR CODE                                  Sp
  ♦ @test the error option for the laserjetII menu        Sp
  ♦ @test the fx100 buzzer menu.
  ♦ @test the fx100 maninmenu
  ♦ @test the profeeder options
  ♦ @TEST THE SELECTION FOR THE FX85/105 PRINTER          Sp
  ♦ @test the self-test option
  ♦ @test the type setting menu
  ♦ @TESTING MODULES FOR PRINTER & INTERFACE
  ♦ @type style problems
  ♦ @what if questions and answers
  ♦ @what to do now?
  ♦ @yes/no laserjet1
  ♦ @yes/no laserjet2
  ♦ @yes/no laserjet3
  ♦ @yes/no laserjet4
  ♦ @yes/no laserjet5
  ♦ @yes/no laserjet6
  ♦ @yes/no laserjet7
    e2$
    ER$
  ♦ laserjet quality 3
  ♦ laserjet quality1
  ♦ laserjet quality2
  ♦ print quality problems for the laserjet
  ♦ RETURN TO REST OF CONSULTATION                        Sp
  ♦ RETURN TO REST OF CONSULTATION-GENSUB3             Ex Sp
  ♦ test the laserjet print quality main option menu
  ♦ test where to go back to-gensub3                   Ex Sp
```

```
*
*  RULE DICTIONARY                      Wed Mar 28 14:15:37 1990  Page:
*  ---------------
  ♦ @IBM Typewriter systematic tests                      Sp
  ♦ beep and light signal problems
  ♦ check no blockages
  ♦ erasing problems
  ♦ IBM 6746 TESTS                                     Ex Sp
  ♦ IBM 6747 TESTS
  ♦ IBM 6783 TESTS                                     Ex Sp
  ♦ OLIVETTI 111 TESTS
  ♦ OLIVETTI 121 TESTS
  ♦ orientation of diagnostics-6746                       Sp
  ♦ paper load freely                                     Sp
  ♦ Print Quality Problems
  ♦ replace faulty items and continue
  ♦ RETURN TO THE REST OF THE CONSULTATION-TYPEWRITERS Ex Sp
  ♦ stand-alone procedures-6746                           Sp
  ♦ test where to go back to-TYPEWRITERS               Ex Sp
  ♦ test which way to go                               Ex Sp
  ♦ TW:check rollers etc                                  Sp
  ♦ TW:print drift                                        Sp
  ♦ TW:sticky labels problem?                             Sp
  ♦ Typing Problems
```

```
*
*  RULE DICTIONARY                      Wed Mar 28 14:12:35 1990  Page:
*  ---------------
    3rd party diag tests
  ♦ All Compaq error codes listed                         Sp
  ♦ Amstrad keyboard faults                               Sp
  ♦ BEGIN ONCE AGAIN                                   Ex Sp
  ♦ check DOS CONFIG                                      Sp
  ♦ check memory allocation                               Sp
  ♦ check reference to DOS                                Sp
  ♦ check the dot-sys files                            Ex Sp
  ♦ compaq deskpro 286e test                              Sp
  ♦ compaq deskpro 386 test                               Sp
  ♦ COMPAQ DESKPRO 386/20e
  ♦ compaq deskpro 386/25                                 Sp
  ♦ Compaq device errors                               Ex Sp
    CR$
    CR1$
    CR2$
    CR3$
    CR4$
    CRIBM$
    cribm1$
    CRIBM2$
  ♦ device error messages                                 Sp
    DOP
  ♦ DOS ROUTINE
    EXPANSION BOARD
  ♦ Floppy Drive Errors                                   Sp
    FONT$ VARIABLE SCREEN
  ♦ gen keyb problems-1                                   Sp
  ♦ general keyboard problems                             Sp
  ♦ hard disk error messages                              Sp
  ♦ Hard Disk Errors                                      Sp
    HERCULES GRAPHIC CARD
  ♦ IBM device errors                                     Sp
  ♦ IBM XT 286 ERROR CODES                                Sp
  ♦ IBM XT ERROR CODES                                    Sp
  ♦ in DW3?                                            Ex Sp
  ♦ IRRATIONAL BEHAVIOUR                                  Sp
  ♦ is num lock on?                                    Ex Sp
  ♦ Is there a beep noise?                             Ex Sp
  ♦ keyboard problems for the PC                       Ex Sp
  ♦ LOOK AT SYMPTOM/ACTION CHART-XT                       Sp
  ♦ LOOK AT SYMPTOM/ACTION CHART-XT 286
  ♦ LOOK AT THE AMSTRAD CODES                             Sp
  ♦ LOOK AT THE COMPAQ CODES                              Sp
  ♦ LOOK AT THE IBM CODES                                 Sp
  ♦ LOOK AT V2.1 DOS ERRORS                               Sp
  ♦ LOOK AT V3 DOS ERRORS                                 Sp
  ♦ look for disk fragmentation                           Sp
  ♦ machine slows-up                                      Sp
  ♦ monitor problems for the PC                           Sp
    MOUSE
  ♦ noisy disk                                            Sp
    OPTIONPC$ VARIABLE SCREEN
  ♦ PC HANGS UP OR STOPS                                  Sp
    pc manufacturers' own diag tests covered
  ♦ PCxMAIN$ VARIABLE SCREEN                           Ex
    plk
  ♦ POWER ON SELF TEST error codes                        Sp
  ♦ REPEAT EVERYTHING
  ♦ REPEAT V3                                             Sp
```

```
*
*  RULE DICTIONARY                        Wed Mar 28 14:12:38 1990  Page:
*  ---------------
  ♦ REPEAT VIEWING-1
  ♦ REPEAT-3
  ♦ REPEAT-4
  ♦ REPEAT:V2                                                   Sp
  ♦ REPEAT:V2-1
  ♦ REPEAT:V3-1                                                 Sp
  ♦ REPEAT:V3-2                                                 Sp
  ♦ RESTART PROCESS-1                                           Sp
  ♦ RESTART PROCESS-2                                           Sp
  ♦ RESTART PROCESS-3                                           Sp
  ♦ RESTART PROCESS-4                                           Sp
  ♦ RESTART THE PROCESS                                         Sp
  ♦ return to main menu                                         Sp
  ♦ RETURN TO REST OF CONSULTATION-HARDWARE                     Sp
  ♦ return to the PC local menu                                 Sp
  ♦ RETURN TO THE REST OF THE CONSULTATION-HARDWARE             Sp
  ♦ save file routine                                           Sp
  ♦ SAVE THE HARDWARE CONSULTATION ROUTINE                   Ex Sp
    savepc
  ♦ see the listing?                                            Sp
    SOFT$ VARIABLE SCREEN
  ♦ software keyboard configuration                             Sp
  ♦ System Unit problems for the PC                          Ex Sp
  ♦ TEST THE APPLE QUIT OPTION                                  Sp
  ♦ Test the bits menu options                               Ex Sp
  ♦ test the hard disk error symptom                            Sp
  ♦ test the options to wipe with special                       Sp
  ♦ test the save file option                                   Sp
  ♦ test the save hardware file option                       Ex Sp
  ♦ test where to go back to-HARDWARE                           Sp
  ♦ TEST WHETHER C OR R- FOUR                                   Sp
  ♦ TEST WHETHER C OR R- ONE                                 Ex Sp
  ♦ TEST WHETHER C OR R- THREE                               Ex Sp
  ♦ TEST WHETHER C OR R- TWO                                 Ex Sp
  ♦ TEST WHETHER C OR R-FIVE                                    Sp
  ♦ test which ibm option to follow:v2                          Sp
  ♦ test which ibm option to follow:v2-1                        Sp
  ♦ test which ibm option to follow:v3                          Sp
  ♦ Test which ibm option to follow:v3-1                     Ex Sp
  ♦ test which option to follow:v3-2                            Sp
  ♦ THE SYSTEM DOESN'T BOOT UP                                  Sp
  ♦ try another disk                                            Sp
  ♦ try another machine                                         Sp
  ♦ use manufacturer's self-diagnostic tests                    Sp
  ♦ Various attachment problems for the PC-boards etc.          Sp
  ♦ what to do now?                                          Ex Sp
  ♦ which compaq to choose?                                     Sp
  ♦ WIPE VARIABLES                                           Ex Sp
    ZLIB$ VARIABLE SCREEN
```

```
*
*  RULE DICTIONARY                      Wed Mar 28 14:08:26 1990  Page:
*  ---------------
  ♦ #problem persists?
  ♦ @beeper error warnings
  ♦ @computer hangs up when trying to print               Sp
  ♦ @does problem still exist1
  ♦ @does problem still exist4
  ♦ @does the problem still exist2                     Ex
  ♦ @does the problem still exist3
  ♦ @does the problem still exist5
  ♦ @does the problem still exist6
  ♦ @does the problem still exist7
  ♦ @error messages LJ2                                   Sp
  ♦ @FX-100+ systematic tests
  ♦ @FX-800/1000 systematic tests
  ♦ @FX-85/105 systematic tests
  ♦ @fx100 beeper error warnings
  ♦ @general troubleshooting for laserjetII
  ♦ @graphics problems
  ♦ @HP Deskjet systematic tests
  ♦ @HP laserjet main menu                                Sp
  ♦ @HP Laserjet SII systematic tests
  ♦ @HP Laserjet systematic tests
  ♦ @laserjet error codes                                 Sp
  ♦ @laserjet self-tests
  ♦ @laserjetII yes/no2
  ♦ @LaserjetII yes/no3
  ♦ @LaserjetII yes/no4
  ♦ @LaserjetII yes/no5
  ♦ @LaserjetII yes/no6
  ♦ @LaserjetII yes/no7
  ♦ @PAPER FEEDING PROBLEMS
  ♦ @paper jam problems with LaserjetII
  ♦ @power problems with the laserjet
  ♦ @print changes during printing
  ♦ @print quality problems
  ♦ @PRINTER DOES NOT PRINT                               Sp
  ♦ @printer fails to print with no reason
  ♦ @Printing is garbled
  ♦ @Problem remain1
  ♦ @problem remain2
  ♦ @problem remain3
  ♦ @problem remain4
  ♦ @problem remain5
  ♦ @problem still exist?
  ♦ @Problems with the papar-end defector
  ♦ @Qume diagnostics with sheet feeder
  ♦ @Qume diagnostics without sheet feeder
  ♦ @Qume Sprint 11 Plus(11/40) systematic tests
  ♦ @resort to the self tests
  ♦ @return to main menu
  ♦ @RETURN TO REST OF CONSULTATION                       Sp
  ♦ @return to the PC local menu
  ♦ @self test for the fx100
  ♦ @still losing data
  ♦ @Systematic investigation of printer faults
  ♦ @Tabbing problems with printer
  ♦ @test beep error warnings option
  ♦ @test blinking error problems
  ♦ @test commonly recurring problems
  ♦ @Test flip menu
  ♦ @test for printing quality problems
```

```
*
*  RULE DICTIONARY                      Wed Mar 28 14:08:28 1990  Page:
*  ---------------
  ♦ @test fx800 / fx1000 printer options
  ♦ @test garbled printing option
  ♦ @test graphics optionsd
  ♦ @test hp deskjet main menu options
  ♦ @test if we can refer to error section
  ♦ @Test Laserjet series two main menu options
  ♦ @test menu not printing
  ♦ @test paper feeding options
  ♦ @test paper feeding problems
  ♦ @test paper related problems
  ♦ @test PC indicates a problem
  ♦ @test print not what you expect
  ♦ @test print quality options
  ♦ @test printer does not print options
  ♦ @test problems when programming graphics etc
  ♦ @test qume diagnostic options with sheet feeder
  ♦ @test selecType problems
  ♦ @test systematic printer test option                  Sp
  ♦ @test tabbing & width problems
  ♦ @test the beep error options
  ♦ @test the computer hangs option                       Sp
  ♦ @TEST THE ERROR CODE                                  Sp
  ♦ @test the error option for the laserjetII menu        Sp
  ♦ @test the fx100 buzzer menu.
  ♦ @test the fx100 maninmenu
  ♦ @test the profeeder options
  ♦ @TEST THE SELECTION FOR THE FX85/105 PRINTER          Sp
  ♦ @test the self-test option
  ♦ @test the type setting menu
  ♦ @TESTING MODULES FOR PRINTER & INTERFACE
  ♦ @type style problems
  ♦ @what if questions and answers
  ♦ @what to do now?
  ♦ @yes/no laserjet1
  ♦ @yes/no laserjet2
  ♦ @yes/no laserjet3
  ♦ @yes/no laserjet4
  ♦ @yes/no laserjet5
  ♦ @yes/no laserjet6
  ♦ @yes/no laserjet7
    DBASE3 SECTION
  ♦ DECMATE PROBLEMS                                      Sp
    DISPLAYWRITE 3 MODULE
    DISPLAYWRITE 370 MODULE
    DISPLAYWRITE 4 MODULE
    DO A TEST LOAD ("ETC....TO GIVE:-
    DOS SECTION
    e2$
    ER$
    export the saved file
  ♦ initialise variables-MAIN                             Sp
  ♦ laserjet quality 3
  ♦ laserjet quality1
  ♦ laserjet quality2
    LEX86 SECTION
  ♦ LOTUS 123 MODULE
  ♦ LOTUS 123 OR DW3/4 ?
  ♦ OTHERS MODULE
  ♦ PC HARDWARE PROBLEMS & QUERIES
  ♦ print quality problems for the laserjet
```

```
*
*  RULE DICTIONARY                      Wed Mar 28 14:08:31 1990  Page:
*  ---------------
  ♦ RECALL A PREVIOUS CONSULTATION                     Ex Sp
  ♦ SYSTEM PROFILES                                       Sp
  ♦ Systematic investigation of application software.
  ♦ test application software local menu
  ♦ TEST OPTION CHOSEN WITHIN THE "OTHERS" SECTION
  ♦ TEST THE APPLE QUIT OPTION
  ♦ test the laserjet print quality main option menu
  ♦ top-level troubleshooting
  ♦ TRY SOMETHING ELSE                                 Ex Sp
```

```
*
*  RULE DICTIONARY                      Wed Mar 28 14:02:56 1990  Page:
*  ---------------
  ♦ HP7475 ERROR MESSAGES
  ♦ HP7550A ERROR MESSAGES
  ♦ REFER TO PLOTTER SUBSECTION                           Sp
  ♦ RETURN TO THE REST OF THE MENU-PLOTTERS               Sp
  ♦ symptoms and solutions                                Sp
  ♦ TEST WHICH PLOTTER WE NEED                         Ex Sp
```

```
*
*  RULE DICTIONARY                     Wed Mar 28 14:04:15 1990  Page:
*  ---------------
  ♦ #problem persists?
  ♦ amend the ribbon
  ♦ amend the ribbon-1000                                 Sp
  ♦ amend the ribbon-QUME                                 Sp
  ♦ beep and light signal problems
  ♦ check paper OK
  ♦ CHECK PAPER OK-LJII
  ♦ check paper thickness                                 Sp
  ♦ check paper thickness-1000                            Sp
  ♦ check paper thickness-QUME                            Sp
  ♦ CHECK TONER & SHAKE
    DBASE3 SECTION
  ♦ DESKJET
    DISPLAYWRITE 3 MODULE
    DISPLAYWRITE 370 MODULE
    DISPLAYWRITE 4 MODULE
    DO A TEST LOAD ("ETC....TO GIVE:-
    DOS SECTION
  ♦ erasing problems
    export the saved file
  ♦ fx-1000*
  ♦ fx-85 printq
  ♦ HP Laserjet systematic tests
  ♦ IBM 3812
  ♦ IBM 6783                                              Sp
  ♦ IBM TYPEWRITER
  ♦ LASERJET BITS                                         Sp
  ♦ laserjet quality 3
  ♦ laserjet quality1
  ♦ laserjet quality2
  ♦ LASER_JET II                                          Sp
    LEX86 SECTION
  ♦ LOTUS 123 MODULE
  ♦ LOTUS 123 OR DW3/4                                    Sp
  ♦ manufacturer’s self-tests                             Sp
  ♦ new ink cartridge                                     Sp
  ♦ OTHERS MODULE
  ♦ Print Quality Problems
  ♦ QUME*                                                 Sp
  ♦ resolve paper thickness -1000
  ♦ resolve paper thickness problem                       Sp
  ♦ resolve paper thickness problem-QUME                  Sp
  ♦ return to main menu
  ♦ RETURN TO REST OF CONSULTATION                        Sp
  ♦ RETURN TO REST OF CONSULTATION-PQ                  Ex Sp
  ♦ return to the PC local menu
  ♦ RETURN TO THE REST OF THE CONSULTATION-TYPEWRITER  Ex Sp
  ♦ RETURN TO THE REST OF THE CONSULTATION-TYPEWRITERS Ex Sp
  ♦ Systematic investigation of application software.
  ♦ Systematic investigation of printer faults
  ♦ test application software local menu
  ♦ test flip menu
  ♦ test flip PQ menu                                     Sp
  ♦ test flip w menu                                      Sp
  ♦ test for printing quality problems-laserjet I         Sp
  ♦ test for the print quality problem with LjetII.       Sp
  ♦ test garbled printing option
  ♦ test menu not printing
  ♦ TEST OPTION CHOSEN WITHIN THE "OTHERS" SECTION
  ♦ test print quality options
```

```
*
*  RULE DICTIONARY                      Wed Mar 28 14:04:18 1990  Page:
*  ---------------
  ♦ test the laserjet print quality main option menu
  ♦ test where to go back to- PCOA                     Ex Sp
  ♦ test where to go back to-?                         Ex Sp
  ♦ Top-level test for Deskjet                            Sp
  ♦ top-level test for dot-matrix printers
  ♦ top-level test for dot-matrix:1000
  ♦ top-level test for Laserjet I                         Sp
  ♦ top-level test for laserjet II
  ♦ top-level test for Qume Daisywheel
  ♦ top-level troubleshooting
  ♦ TRY A NEW TONER-LASERJET ii                           Sp
  ♦ try different paper:DM
  ♦ try different paper:DM1000                            Sp
  ♦ try different paper:QUME
  ♦ try new ribbon
  ♦ try new ribbon /daisywheel                            Sp
  ♦ try new ribbon-1000                                   Sp
  ♦ try new toner                                         Sp
  ♦ TW:check rollers etc                                  Sp
  ♦ TW:check the paper thickness lever is set correctl Ex Sp
  ♦ TW:check the typewriter's impression strength is O Ex Sp
  ♦ TW:fault lies in the typewriter itself             Ex Sp
  ♦ TW:platen is dirty                                 Ex Sp
  ♦ TW:print drift                                        Sp
  ♦ TW:rectify fault with paper thickness              Ex Sp
  ♦ TW:replace faulty item and repeat                     Sp
  ♦ TW:sticky labels problem?                             Sp
  ♦ TW:try a change of paper                           Ex Sp
  ♦ TW:try again                                          Sp
  ♦ TW:try other supplies
  ♦ TW:typewriter print unsatisfactory                    Sp
  ♦ TW:typewriter with new supplies                       Sp
  ♦ Typing Problems
```

```
*
*  RULE DICTIONARY                      Wed Mar 28 14:36:47 1990  Page:
*  ---------------
  ♦ DECmate boot-up error codes
  ♦ DECMATE GENERAL TROUBLESHOOTER                        Sp
  ♦ DECmate related printer problems                      Sp
  ♦ RETURN TO THE REST OF THE CONSULTATION-DECMATE     Ex Sp
  ♦ test where to go back to-DECMATE                   Ex Sp
*
```

Appendix XV

A Survey of Users of Expert Systems in Manufacturing Sectors

Appendix XV: Part A.

An Exhibit of the Questionnaire

EXPERT SYSTEMS APPLICATIONS SURVEY QUESTIONNAIRE

Please Note: All information will be kept confidential. Only aggregated data will be used in the analysis of results. The personal details requested below are to enable identification of duplicate returns for an application. Write "AS BEFORE" in Sections 1 and 2 if more than one questionnaire is being submitted in the same batch.

SECTION 1 : PERSONAL DETAILS

Name: . . . .

Job title: . . . .

Company: . . . .

Department: . . . .

Address: . . . .

. . . .

. . . .

Telephone Number: . . . .

My overall responsibility for expert systems is:

IT specialist with interest in es ☐

Non-IT specialist with interest in es ☐

Developer ☐

User of an es application ☐

Consultant ☐

Other . . . .

I am prepared for follow up consultation Y / N

SECTION 2 : COMPANY DETAILS

The overall status of expert systems in my organisation is:

No current interest ☐

Past interest but now discarded ☐

To maintain a watching brief ☐

To develop prototype(s) / demonstrators ☐

Applications are under development ☐

Have operational systems ☐

If 'No interest ' please indicate why:

(if you have any experience of expert systems, please complete the rest of the questionnaire)

How many applications have been considered in your organisation (current or not)? *(please write in number)*

Ideas ____

At specification stage ____

Prototype / demonstrator developments ____

Full development systems ____

Operational systems ____

Total ____

Please photocopy this questionnaire as many times as the number of applications totalled in Section 2 and complete for each. Where the application is not operational, please answer the questions to describe the system as it is intended to operate. Remember: it is as important to identify the constraints on system development as to identify successful applications.

SECTION 3 : DESCRIPTION OF USER

In what type of industry is the application used *(e.g. Banking / Textile manufacture)*:

. . . .

. . . .

Please give a brief job description of the user *(e.g. Industrial nurse / Accountant / Salesman / Personnel Manager)*

. . . .

Please give a brief description of the application:

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

SECTION 4 : APPLICATION DETAILS

Application Name: . . . .

What is its current status? *(Tick box)*

Idea ☐

Specification ☐

Prototype / demonstrator ☐

Development ☐

Operational ☐

Development terminated / system withdrawn ☐

Other *(please state)* . . . .

What was used to develop the system?

Software name:*(please state)* . . . .

Shell ☐

AI Tool ☐

AI Language ☐

Conventional Language ☐

Development Hardware: *(please state)* . . . .

Delivery Hardware: *(please state)* . . . .

Who was or will be responsible for development?

Systems dept ☐

User ☐

Specialist Knowledge Engineer ☐

Consultants ☐

Other *(please state)* . . . .

SECTION 5 : SYSTEM CLASSIFICATION

Which functions are supported by the application? *(Tick all that apply. Please rank those ticked in order of importance.)*

a. Administration ☐ ___
b. Information acquisition ☐ ___
c. Information presentation ☐ ___
d. Information interpretation ☐ ___
e. Prediction ☐ ___
f. Diagnosis ☐ ___
g. Selection ☐ ___
h. Alerting / Warning ☐ ___
i. Scheduling ☐ ___
j. Configuration ☐ ___
k. Design ☐ ___
l. Monitoring ☐ ___
m. Control ☐ ___
n. Testing ☐ ___
o. Reporting ☐ ___
p. Debugging ☐ ___
q. Research ☐ ___
r. Training ☐ ___

s. Other *(specify)* . . . . ___

t. Other *(specify)* . . . . ___

SECTION 6 : SYSTEM USE

Was the system designed to: *(Tick all that apply. Please rank those ticked in order of importance.)*

a. Give advice to a 'lay' person ☐ ___
b. Support a knowledgeable user ☐ ___
c. Replace a human ☐ ___
d. Real-time feedback control ☐ ___

e. Other . . . . ___

Is the system used as originally designed? Y / N

If not, what has been changed? . . . .

. . . .

. . . .

How integrated is your expert system with other software?

a. Embedded: (Forms part of a wider system) Y / N
b. Integrated:
  Interfaced to a database Y / N
  Interfaced to other systems Y / N
  Is loaded from other systems Y / N
  Is downloaded to another system Y / N
c. Front End:
  (Used to provide easy access to another system) Y / N
d. Stand alone: Y / N

If interfaced to a database, which one?:

. . . .

How is the knowledge base to be maintained?:

. . . .

. . . .

. . . .

. . . .

SECTION 7: DEVELOPMENT PROGRAMME

Have you got a methodology or use any published guidelines, please describe key steps *(eg prototyping, documentation)*:

1. . . . .

2. . . . .

3. . . . .

4. . . . .

5. . . . .

Were project planning/control tools used? ☐
Were software engineering tools used? ☐

Any other tools? *(specify)* . . . . ☐

How was the knowledge obtained?:

a. Interviews ☐
b. Literature ☐
c. Manuals ☐
d. User observation techniques (eg video) ☐

e. Other . . . . ☐

SECTION 8 : IMPLEMENTATION SUCCESS

Please give your best qualitive estimate, by ringing the appropriate number, along the scale of:
0 = *'No effect'*, 2 = *'some effect'* 4 = *'significant effect'*

1. **Do you consider or expect the system to** *(scale)*:
2. **What did you hope would be the key benefits of your system?** *(Please rank in order of importance)*

a. Increase the quality of work? 0 1 2 3 4 ___
b. Increase accuracy of work? 0 1 2 3 4 ___
c. Increase accuracy of decisions? 0 1 2 3 4 ___
d. Increase output? 0 1 2 3 4 ___
e. Reduce the skill level? 0 1 2 3 4 ___
f. Reduce skilled personnel required? 0 1 2 3 4 ___
g. Enable staff reductions? 0 1 2 3 4 ___
h. Increase workload? 0 1 2 3 4 ___
i. Be cost effective? 0 1 2 3 4 ___
j. Increase problem solving ability? 0 1 2 3 4 ___

k. Other *(specify)* . . . . 0 1 2 3 4 ___

l. Other *(specify)* . . . . 0 1 2 3 4 ___

3. What were the system implementation constraints? *(scale)*
4. Did any of these prevent implementation? *(tick box)*

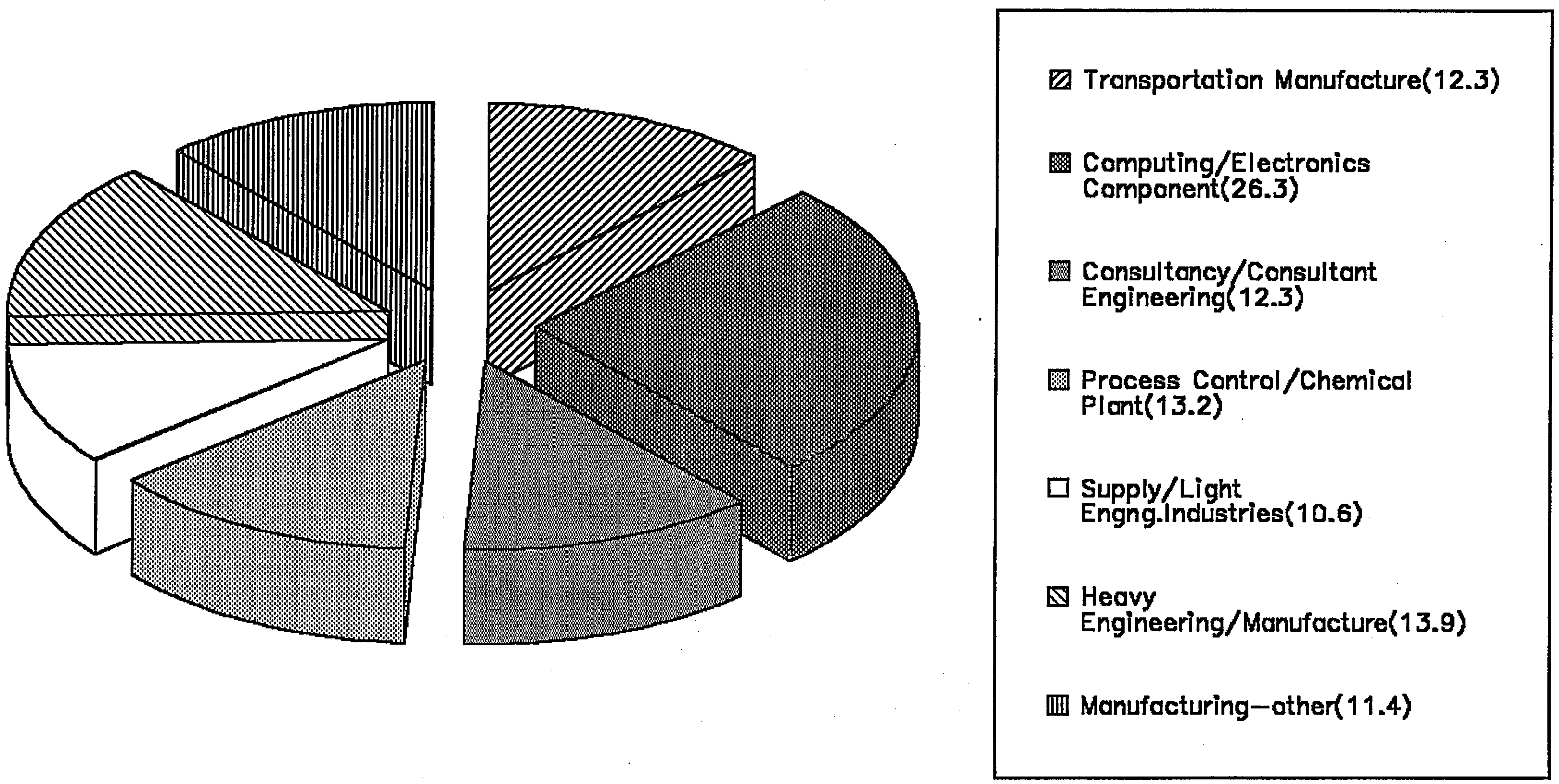
a. The lack of an identified user? 0 1 2 3 4 ☐
b. The lack of suitable experts? 0 1 2 3 4 ☐
c. The lack of technical backup? 0 1 2 3 4 ☐
d. The lack of knowledge engineers? 0 1 2 3 4 ☐
e. A lack of financial gain? 0 1 2 3 4 ☐
f. A lack of awareness of benefits? 0 1 2 3 4 ☐
g. A lack of confidence in the system? 0 1 2 3 4 ☐
h. The lack of management awareness? 0 1 2 3 4 ☐
i. The lack of management support? 0 1 2 3 4 ☐
j. The lack of budget provision? 0 1 2 3 4 ☐
k. Preference for conventional systems? 0 1 2 3 4 ☐
l. Other *(specify)* . . . . 0 1 2 3 4 ☐

Please add any further comments: *(separate sheet if needed)*

Appendix XV: Part B

A Listing of Figures (A- H)

Figure A1: Percentage Distribution of Sample By Manufacturing Sector



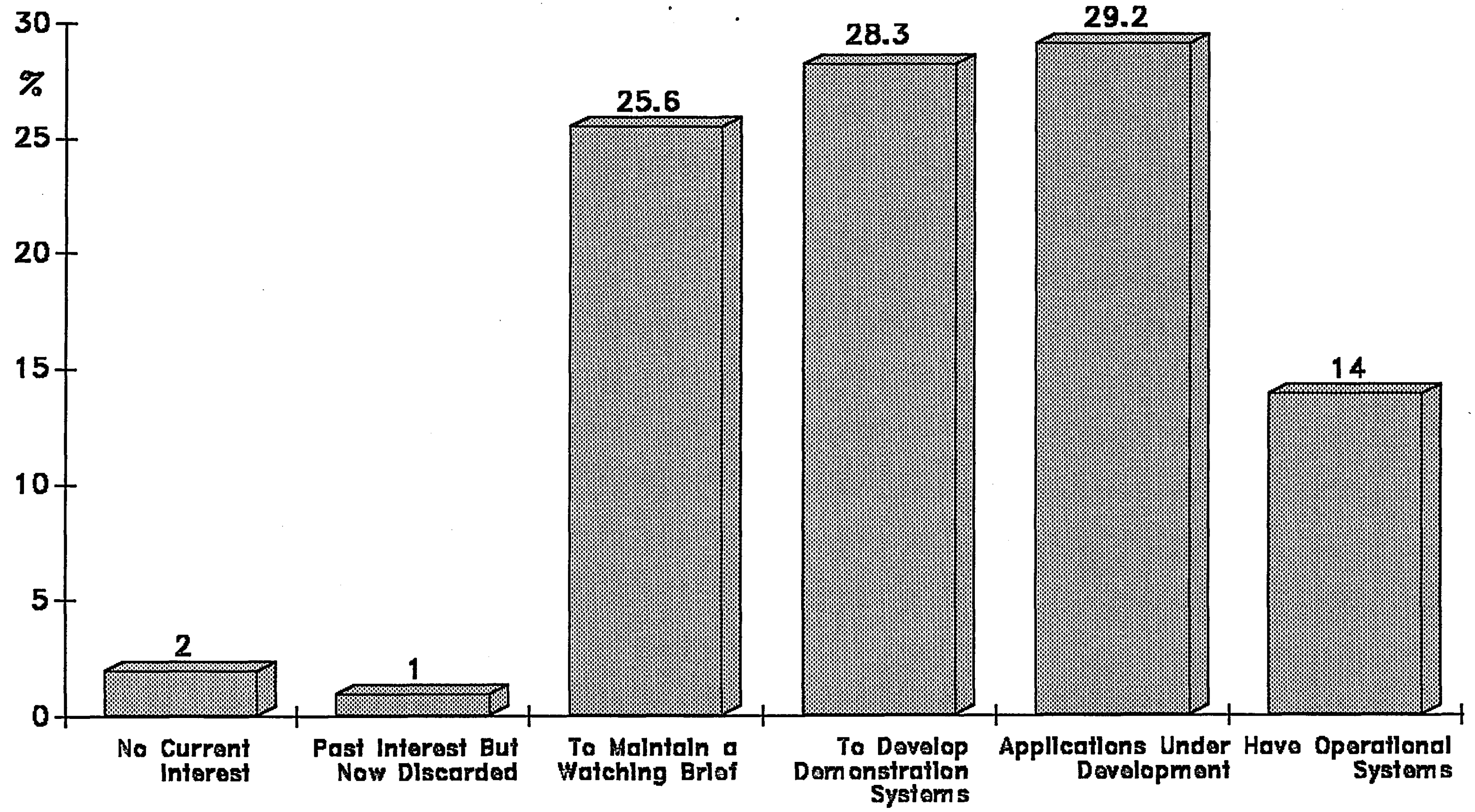


Figure A2: The Overall Status of Expert Systems in the Organisation

Figure A3: The Status Of Expert Systems Across Manufacturing Sectors

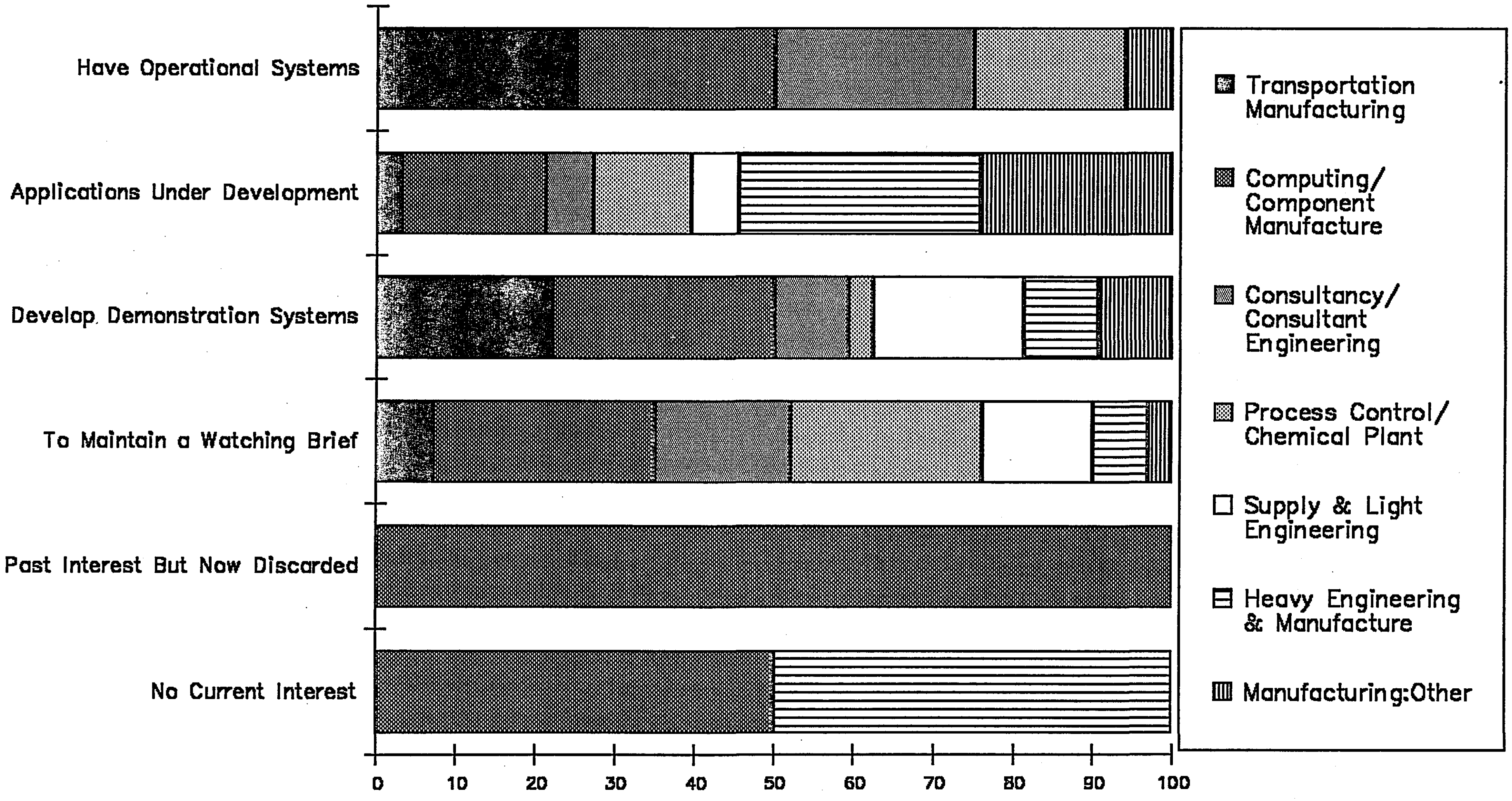


Figure A4: An Account of Why Firms Show No Interest in Expert Systems

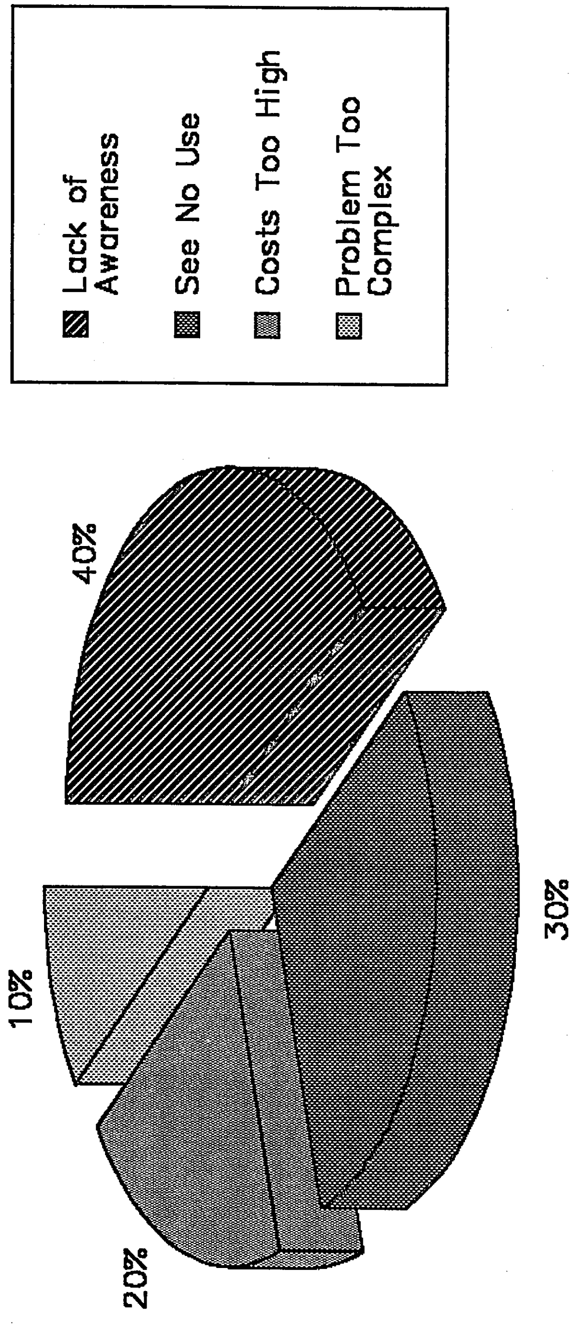
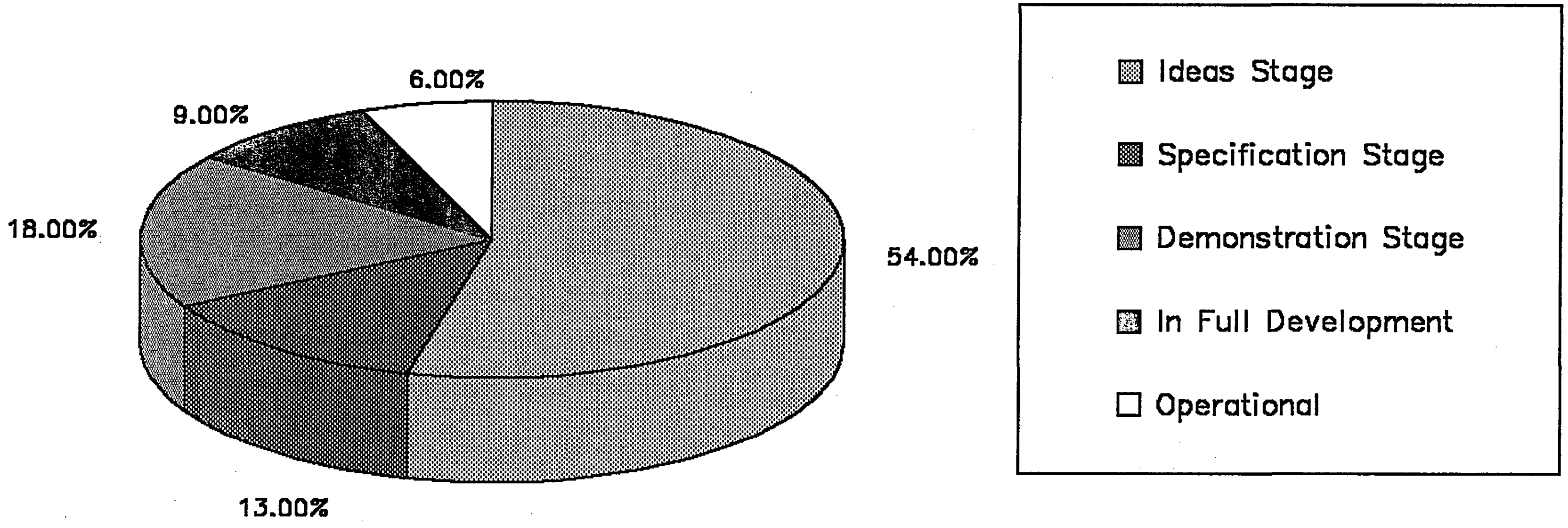


Figure B1: The Number of Applications Considered in Organisations & Their Development Status



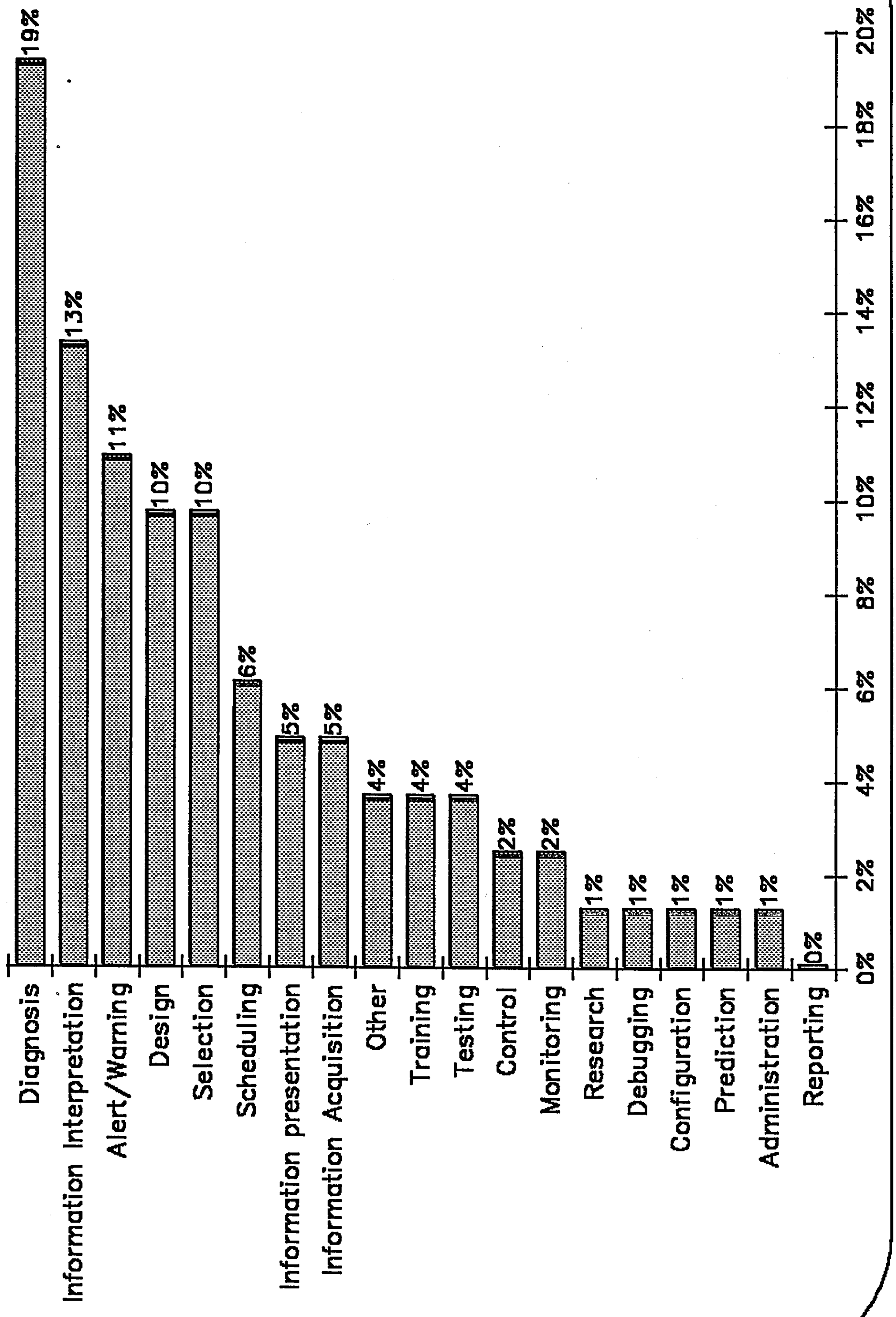


Figure B2: The Most Important (Ranked First) Application Function

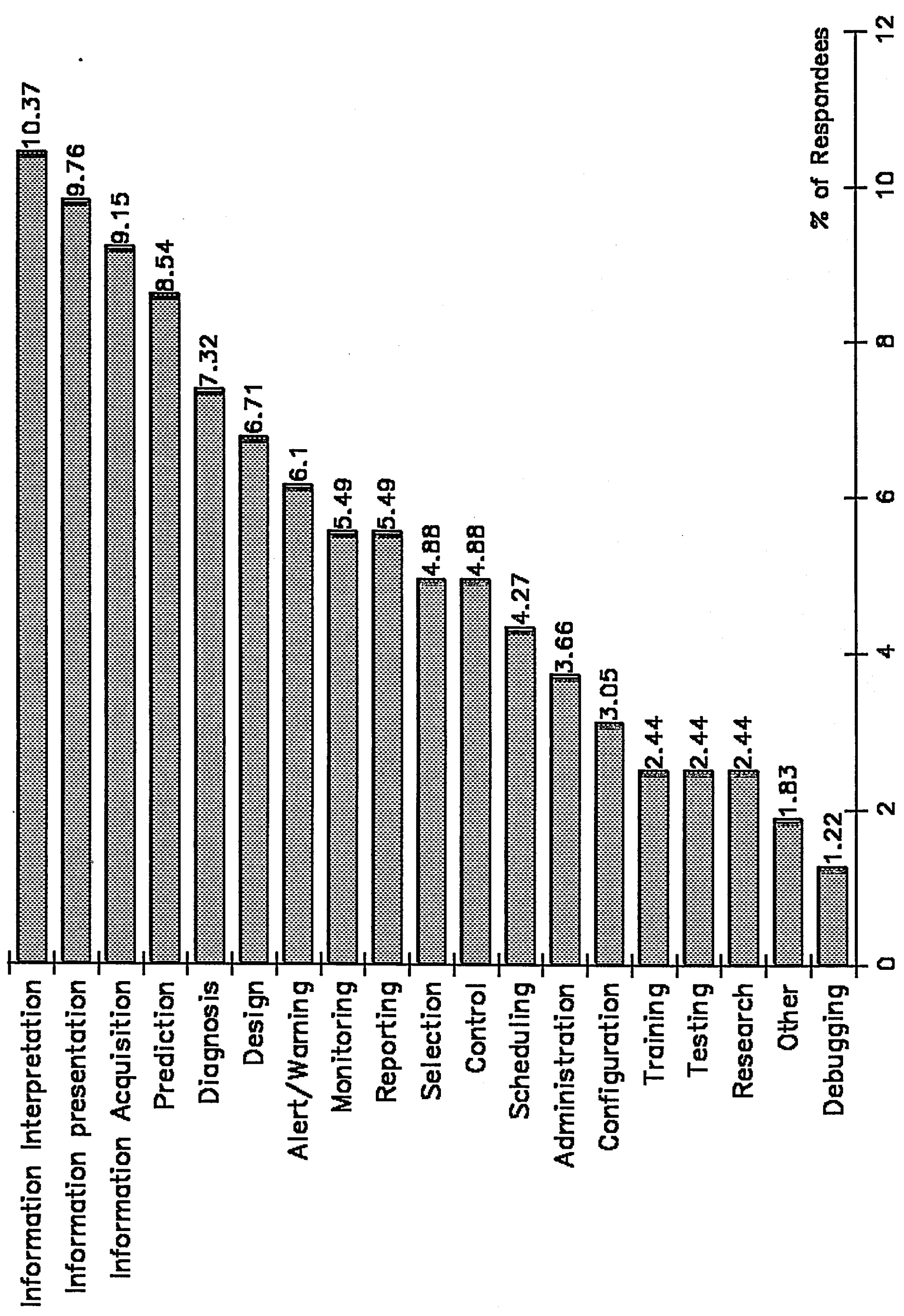


Figure B3: Most Frequently Supported Application Function

Figure C1: Categorisation of Use By User (%)

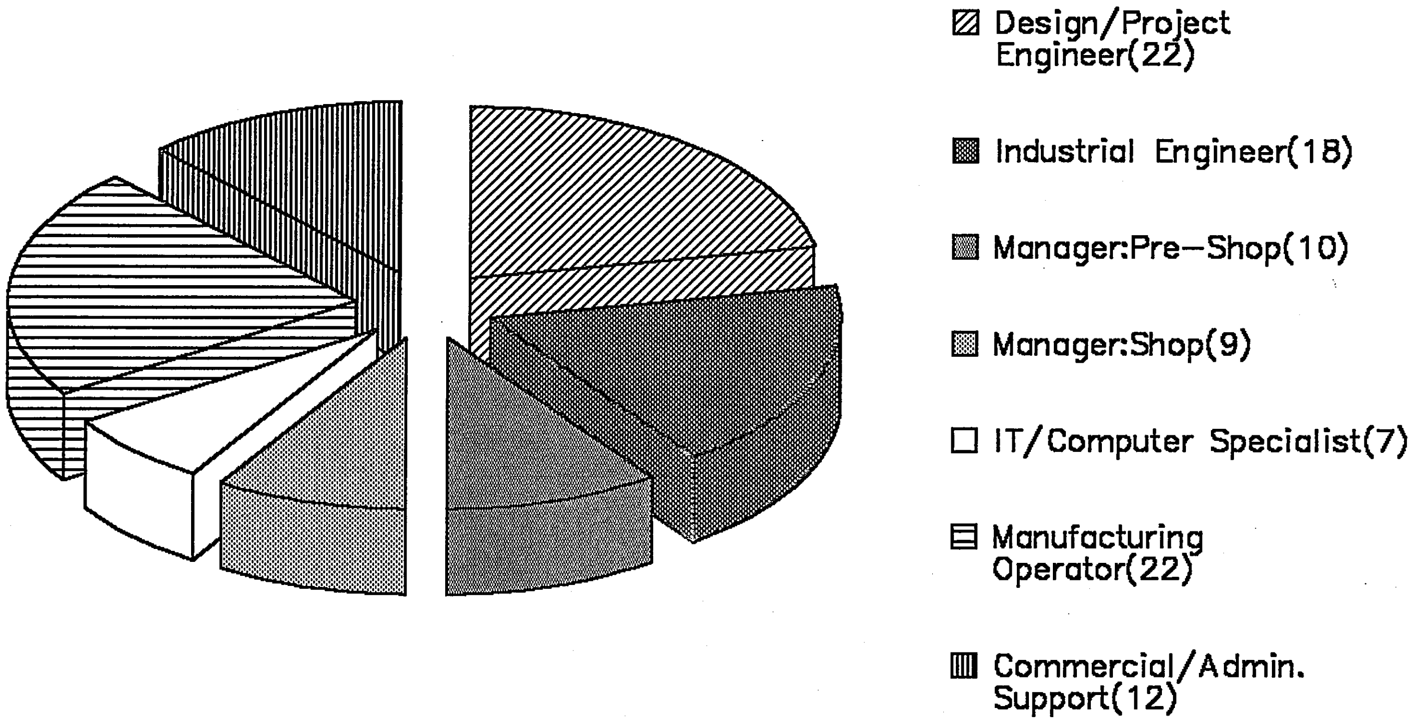
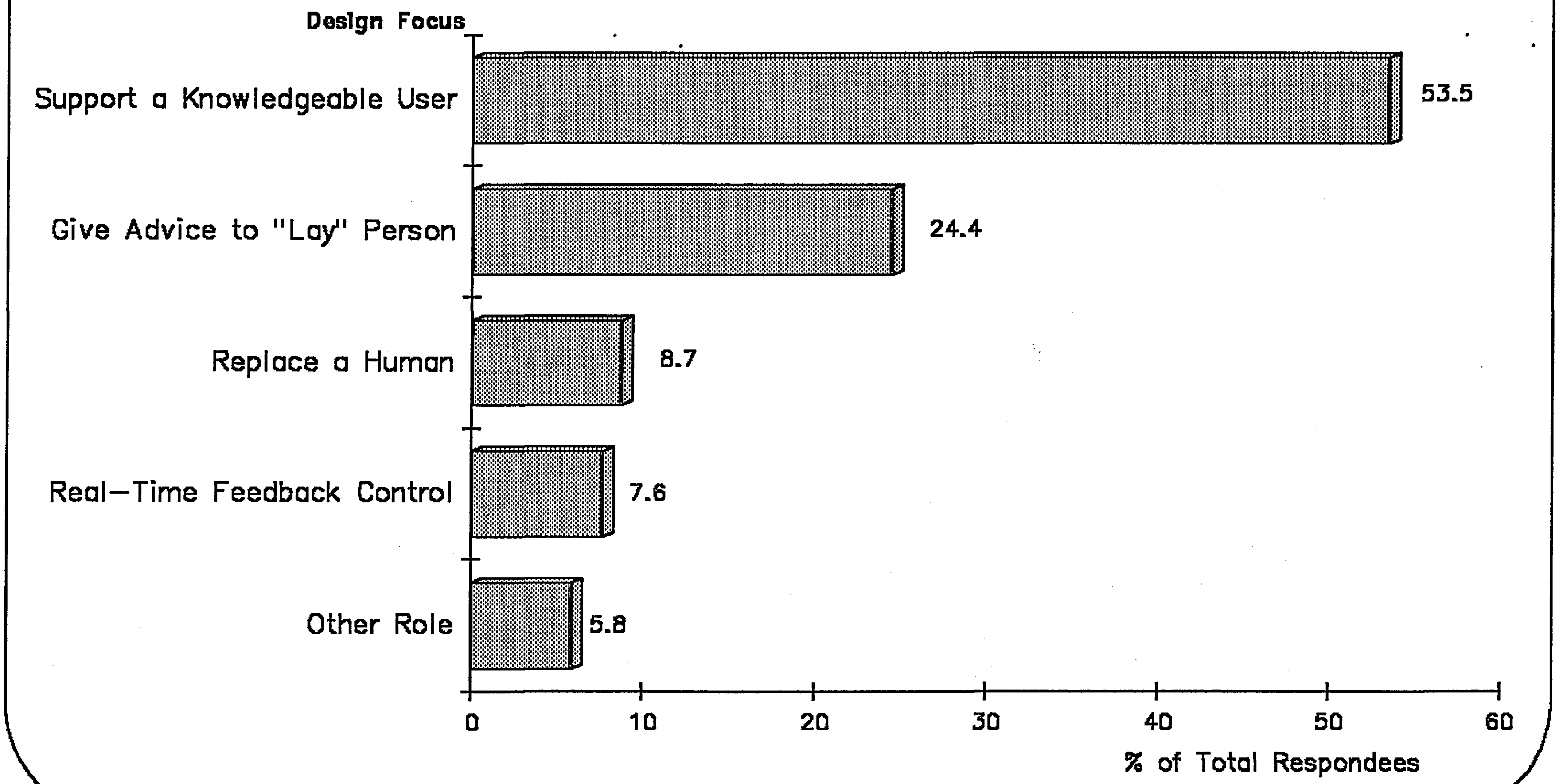


Figure C2: The Intended Use of the System



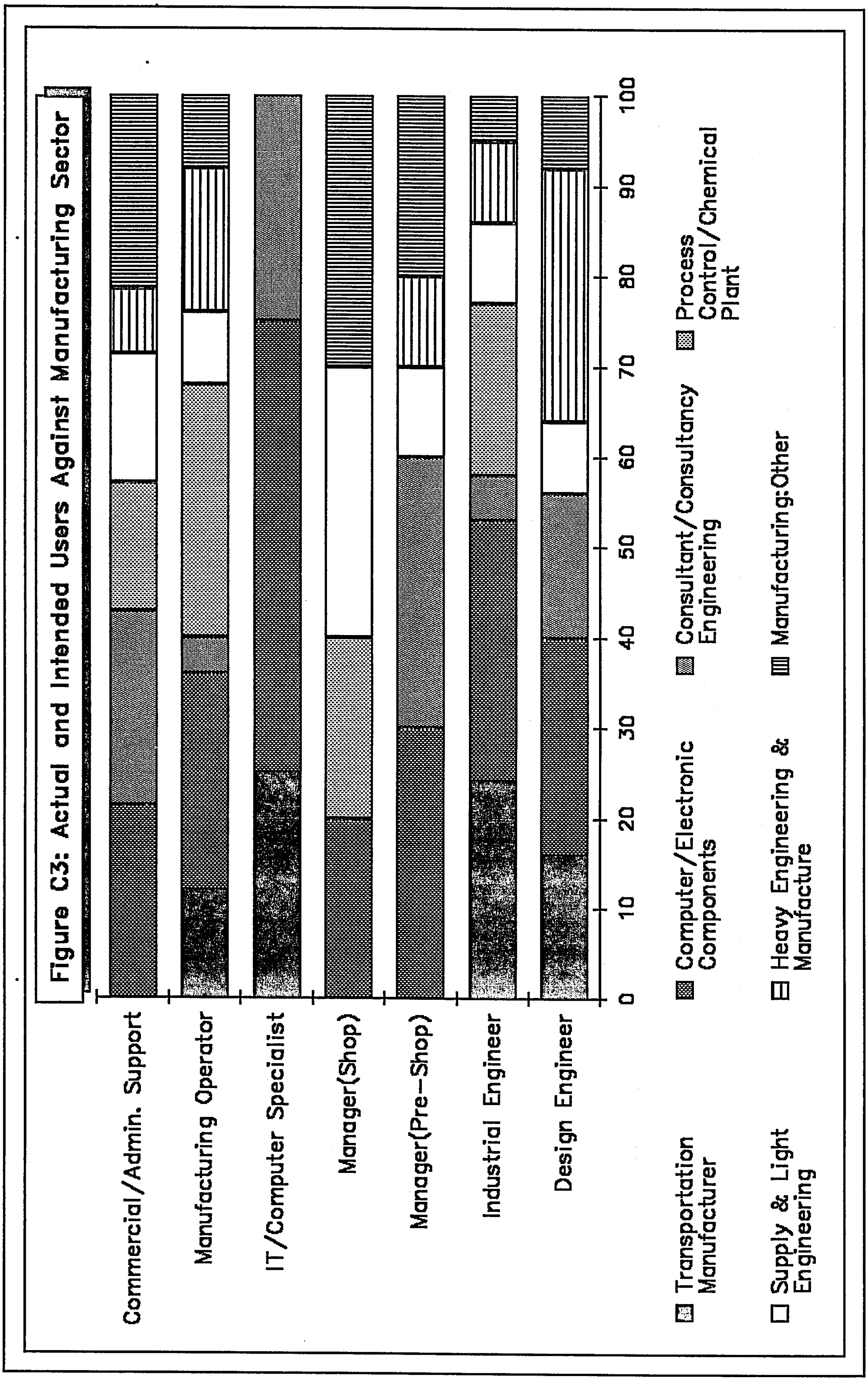


Figure C3: Actual and Intended Users Against Manufacturing Sector

Figure D1: Application Software and Hardware Requirements

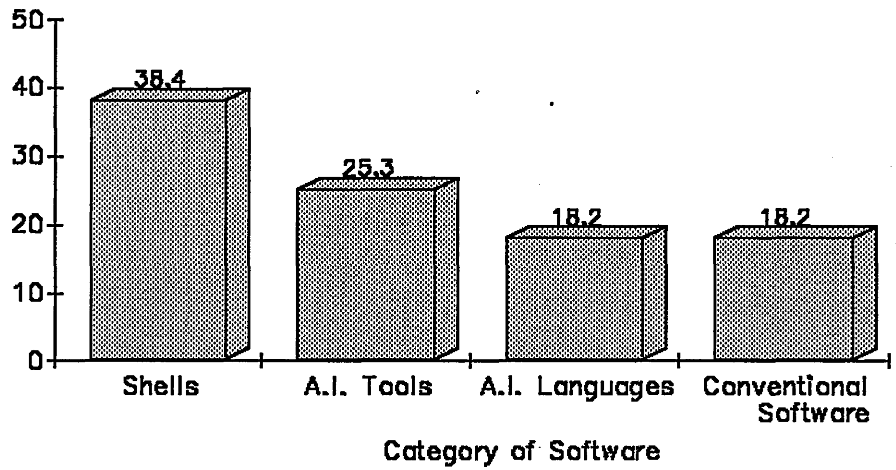


Fig D1.1 The Use of Application Software

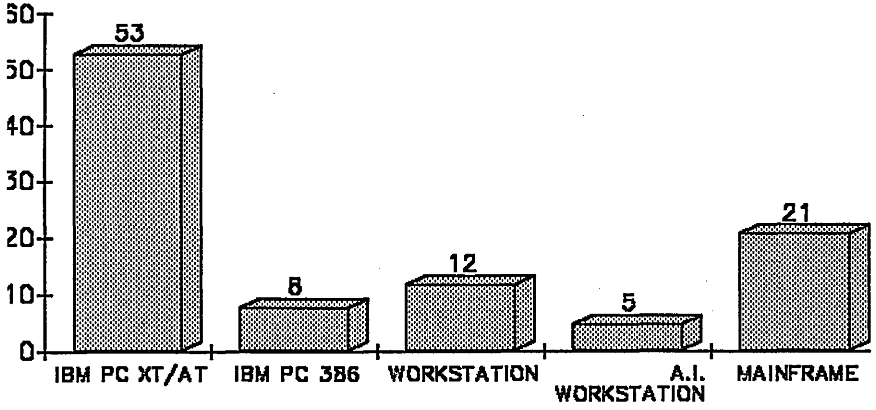


Fig D1.2: The Use of Development hardware

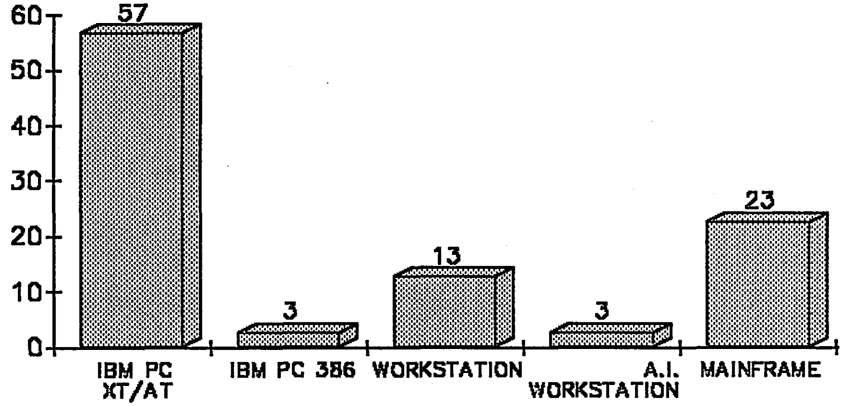


Fig D1.3: The Use of Delivery Hardware

Figure D2: Integration of the Expert System

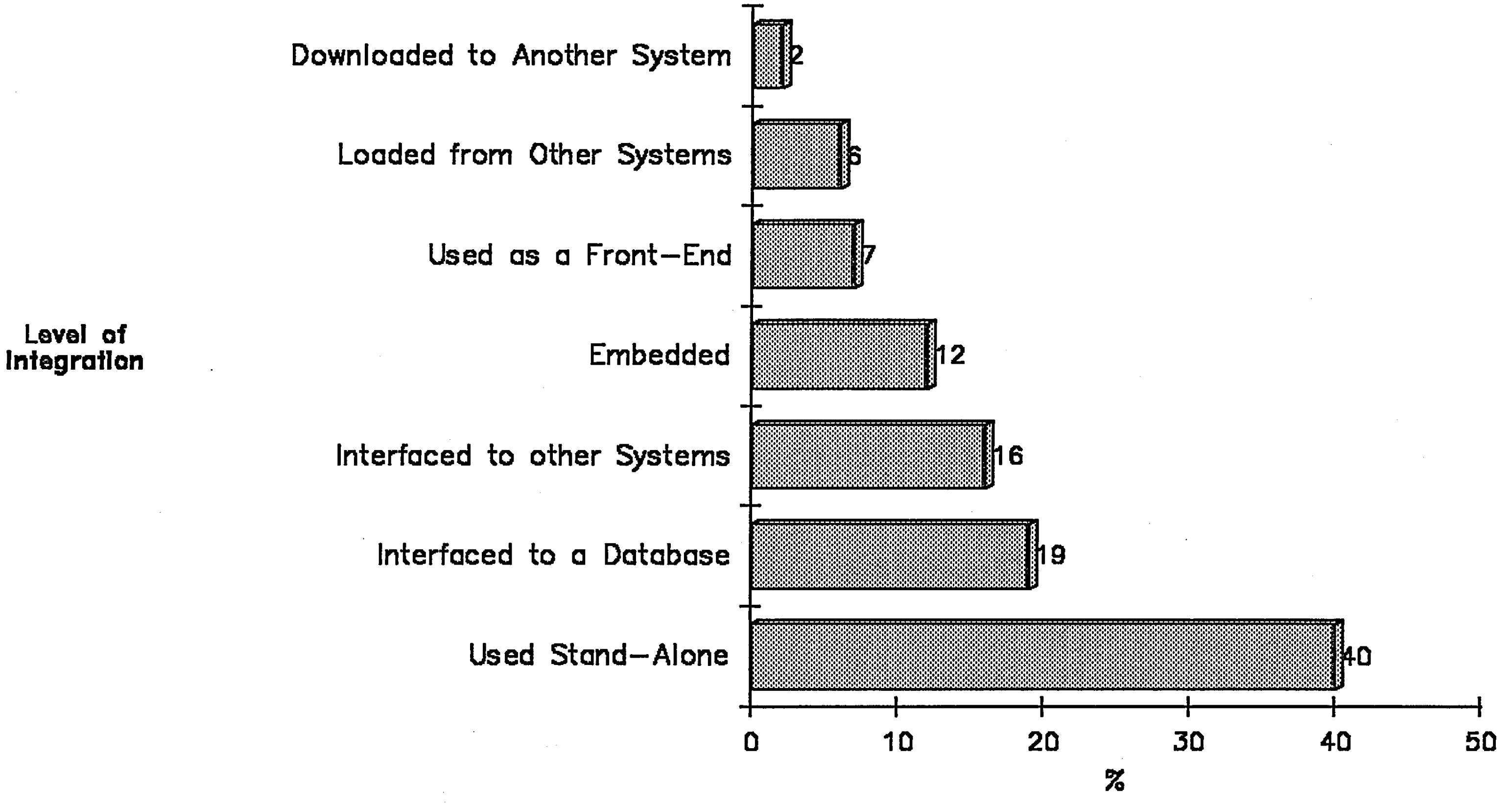


Figure D3: The Impact of Integration Level upon System Use

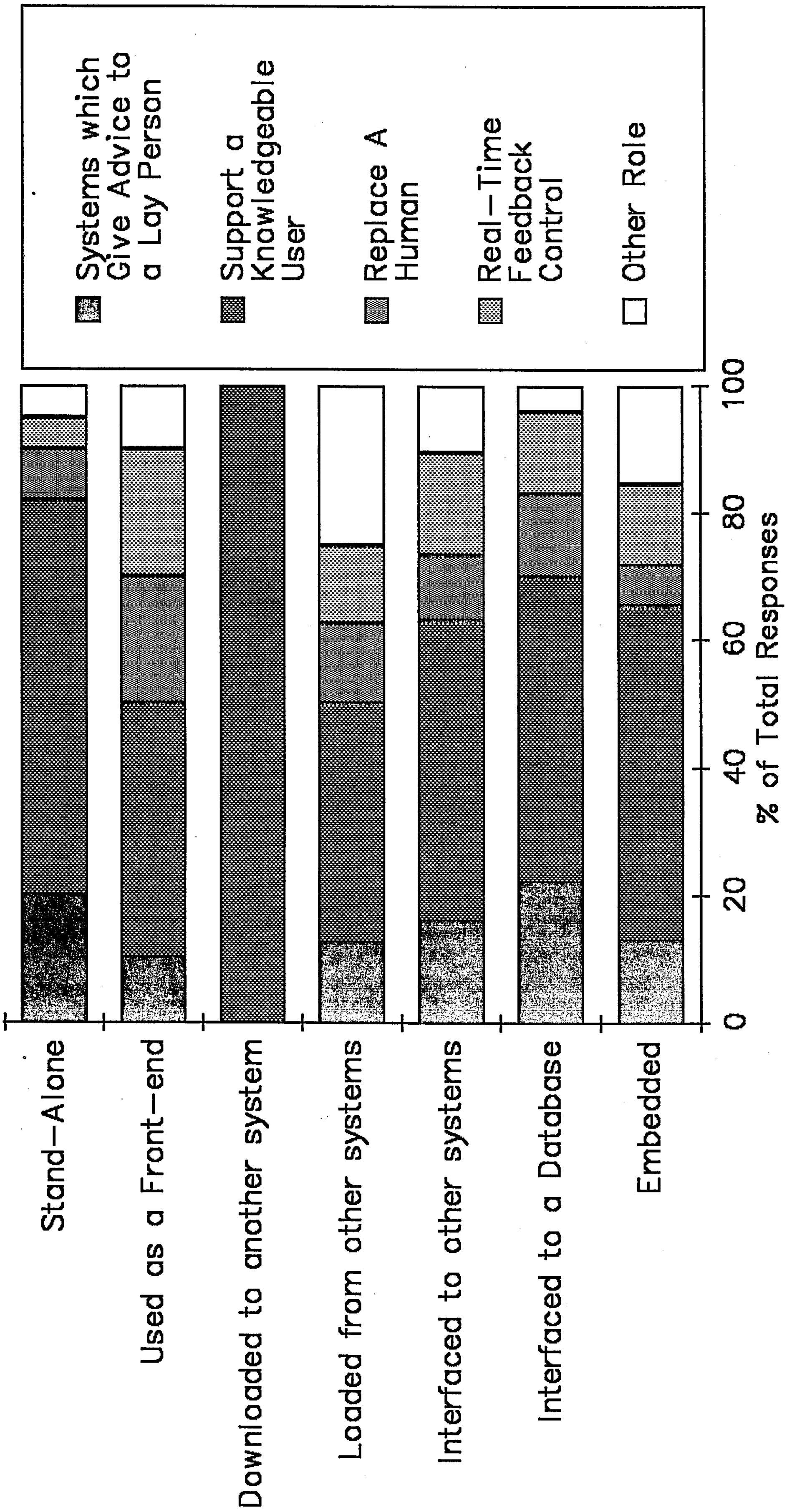


Figure E1: Approaches Towards Development

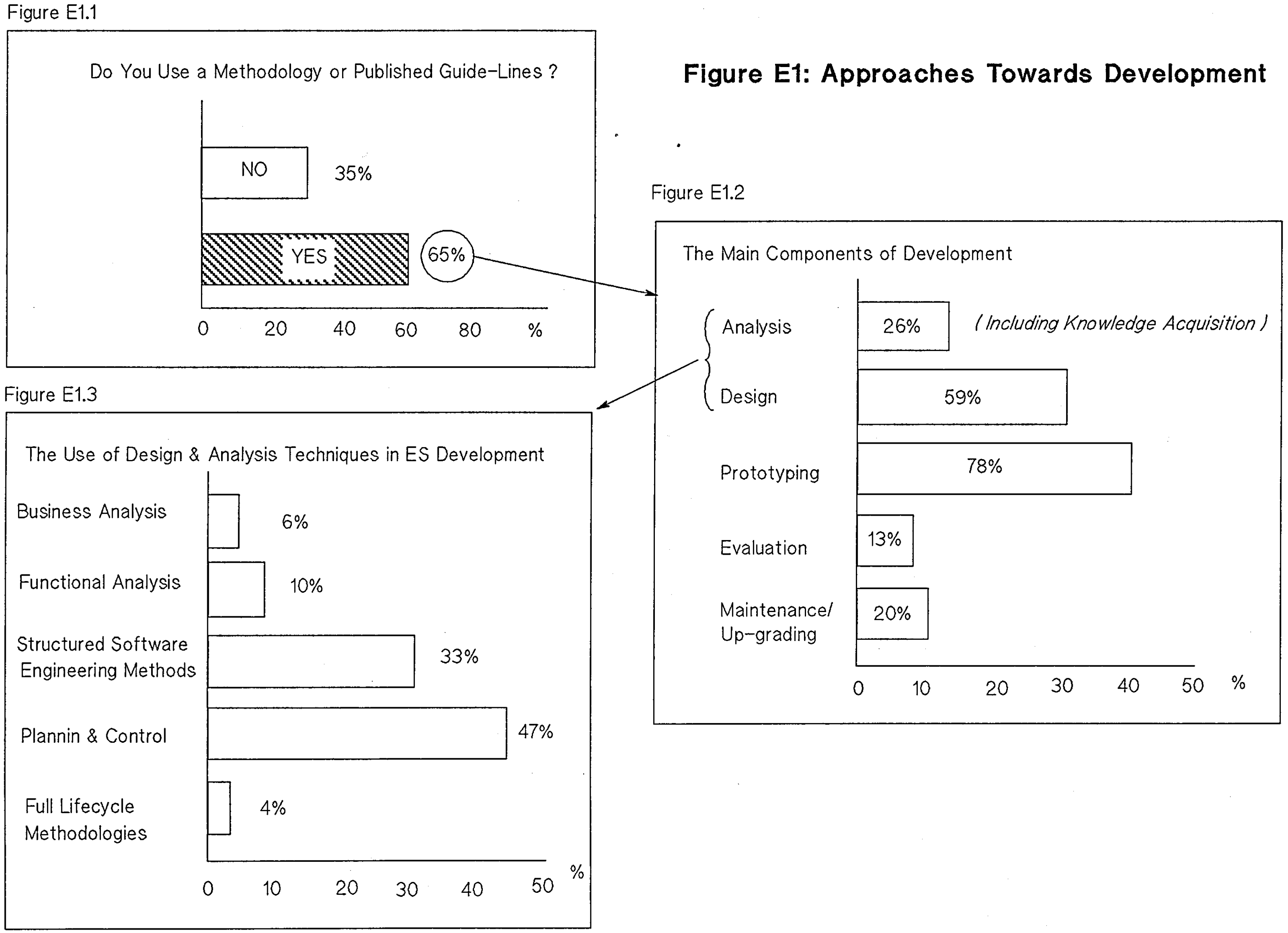


Figure E1.1

Figure E1.2

Figure E1.3

Figure E2: How was the Knowledge Obtained ?

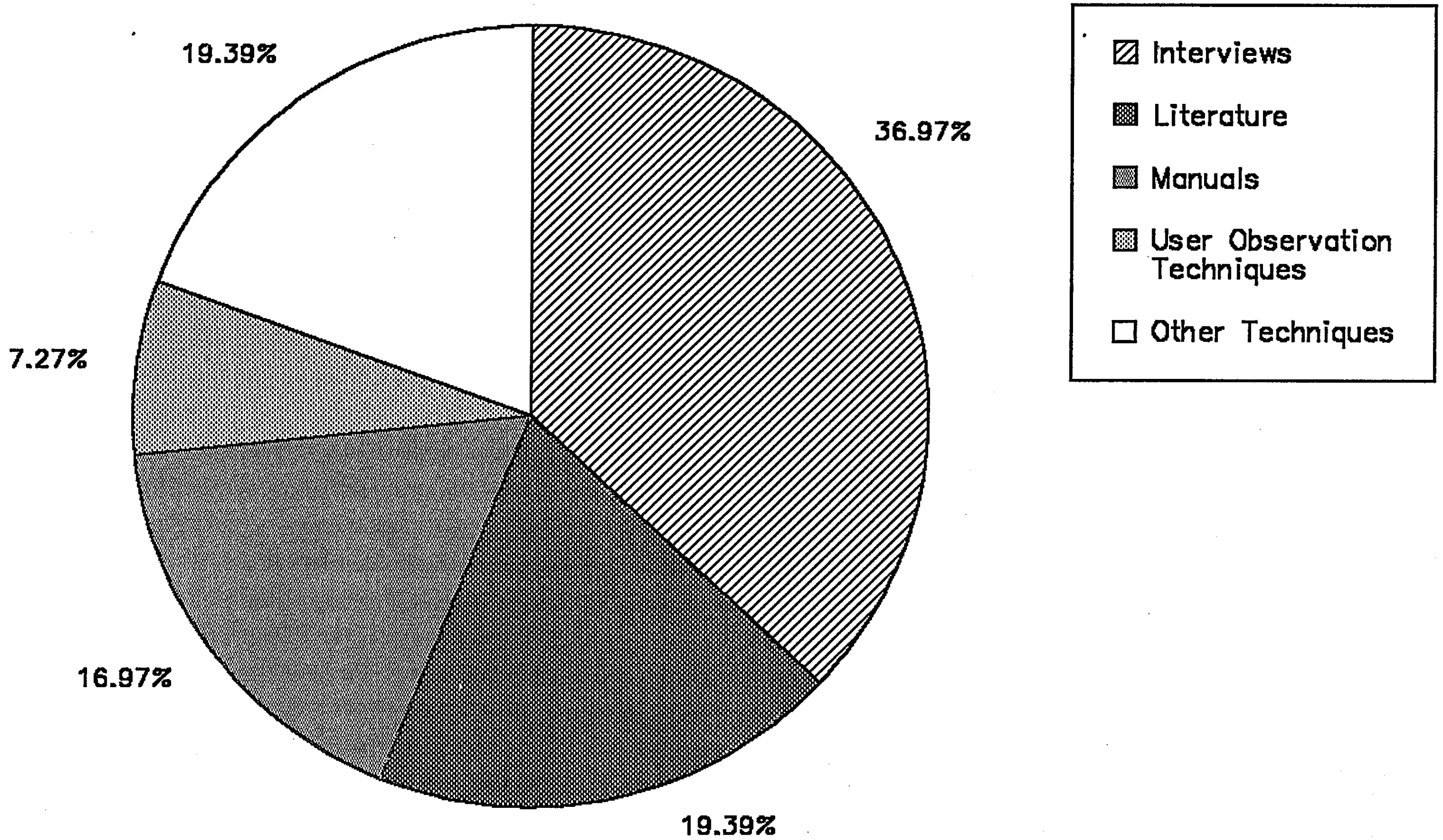


Figure E3: Responsibility for Development

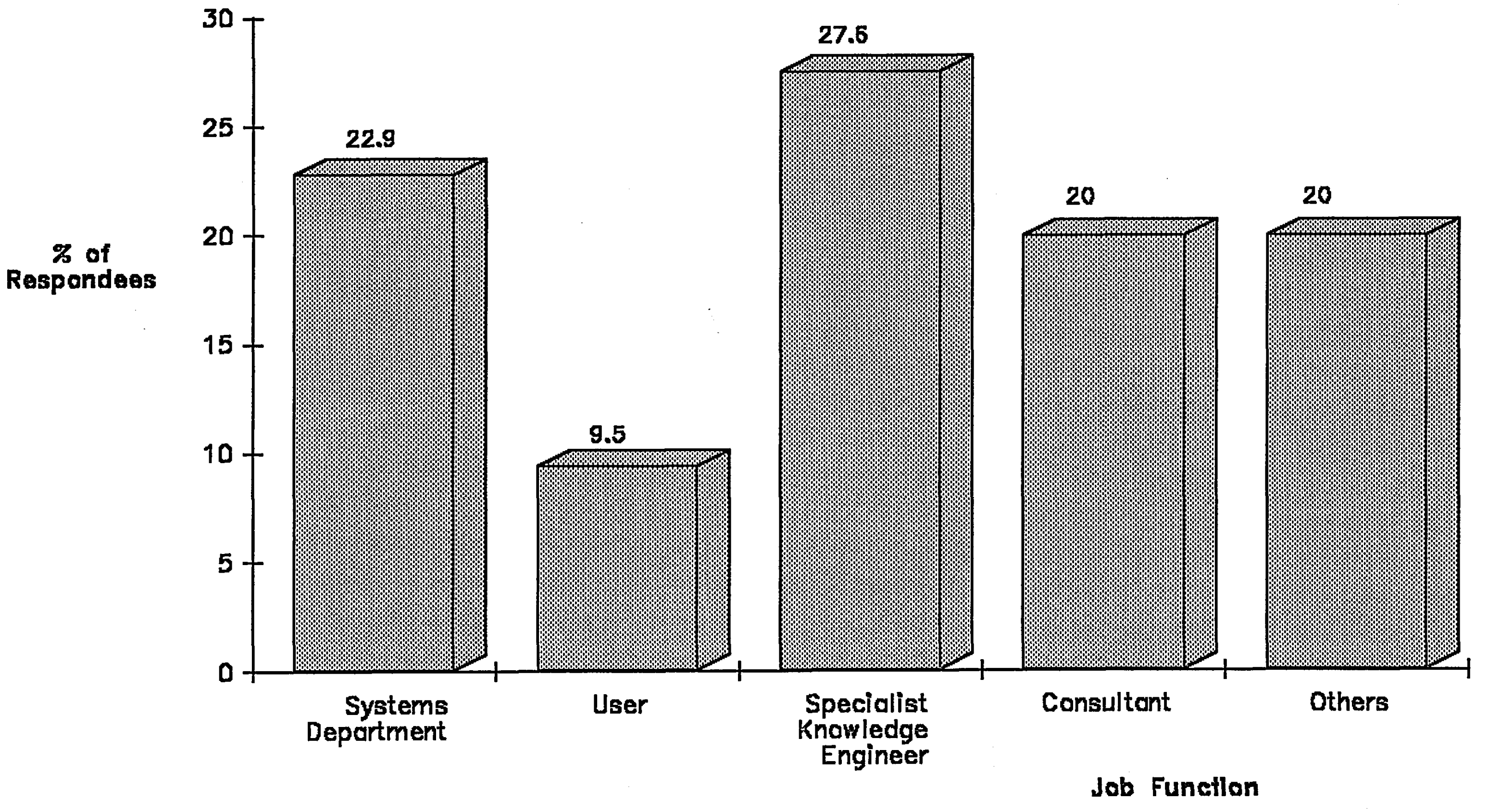
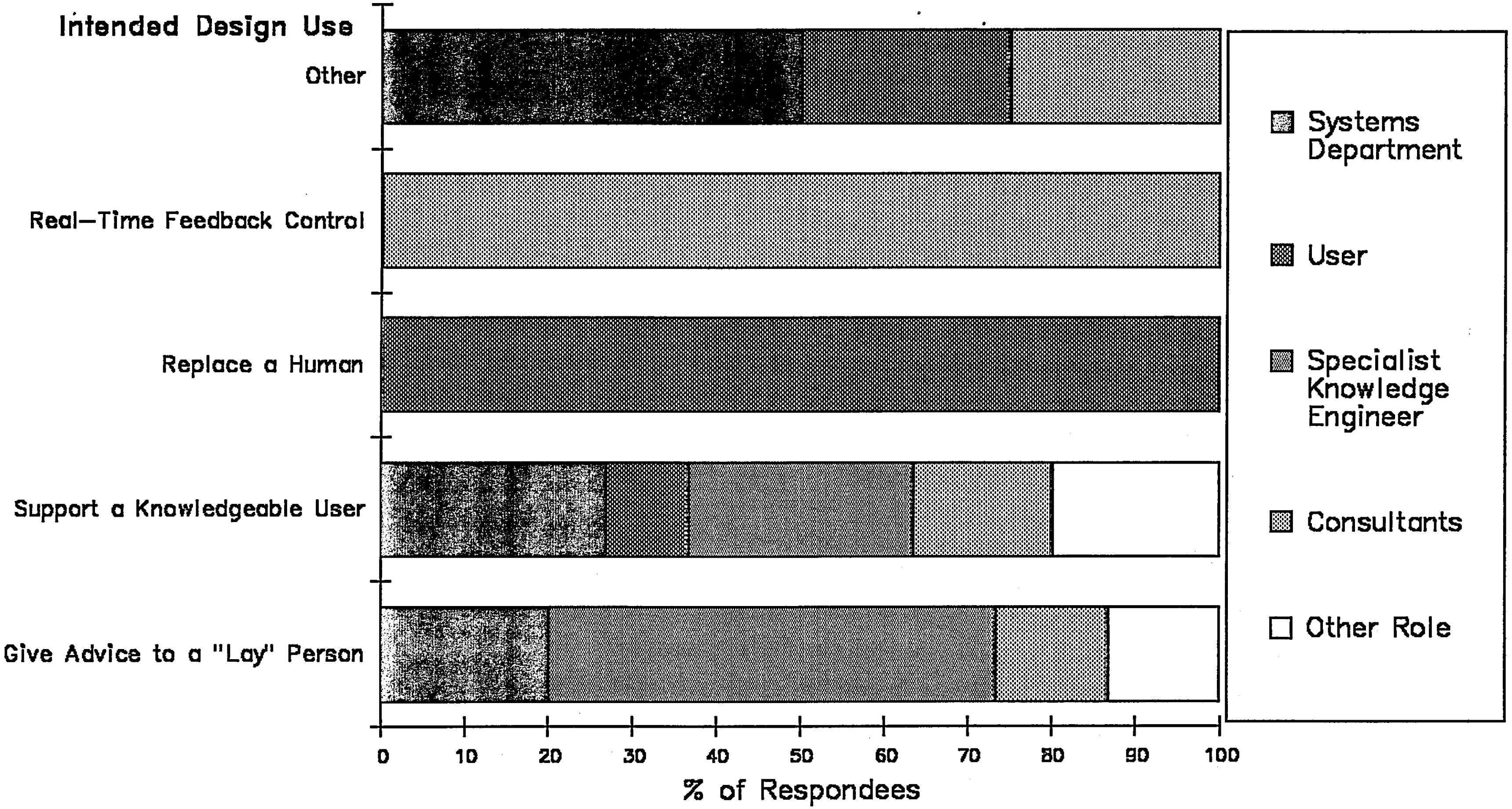


Figure E4: Highest Ranking System Use Against Development Responsibility



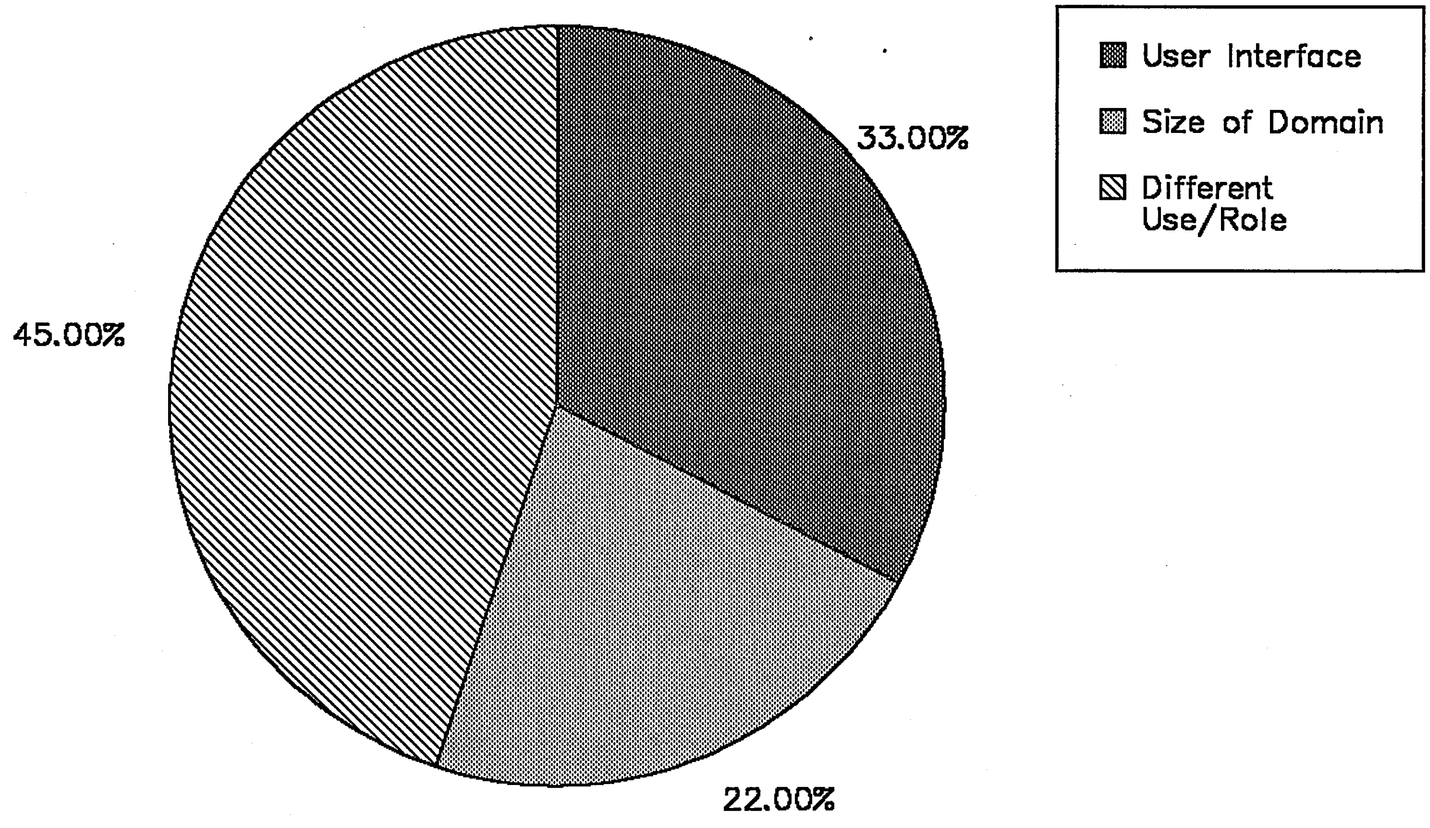


Figure F1: Changes in Design & Use During Operations

Figure F2: Most Important Expressed Key Benefits (Percentage)

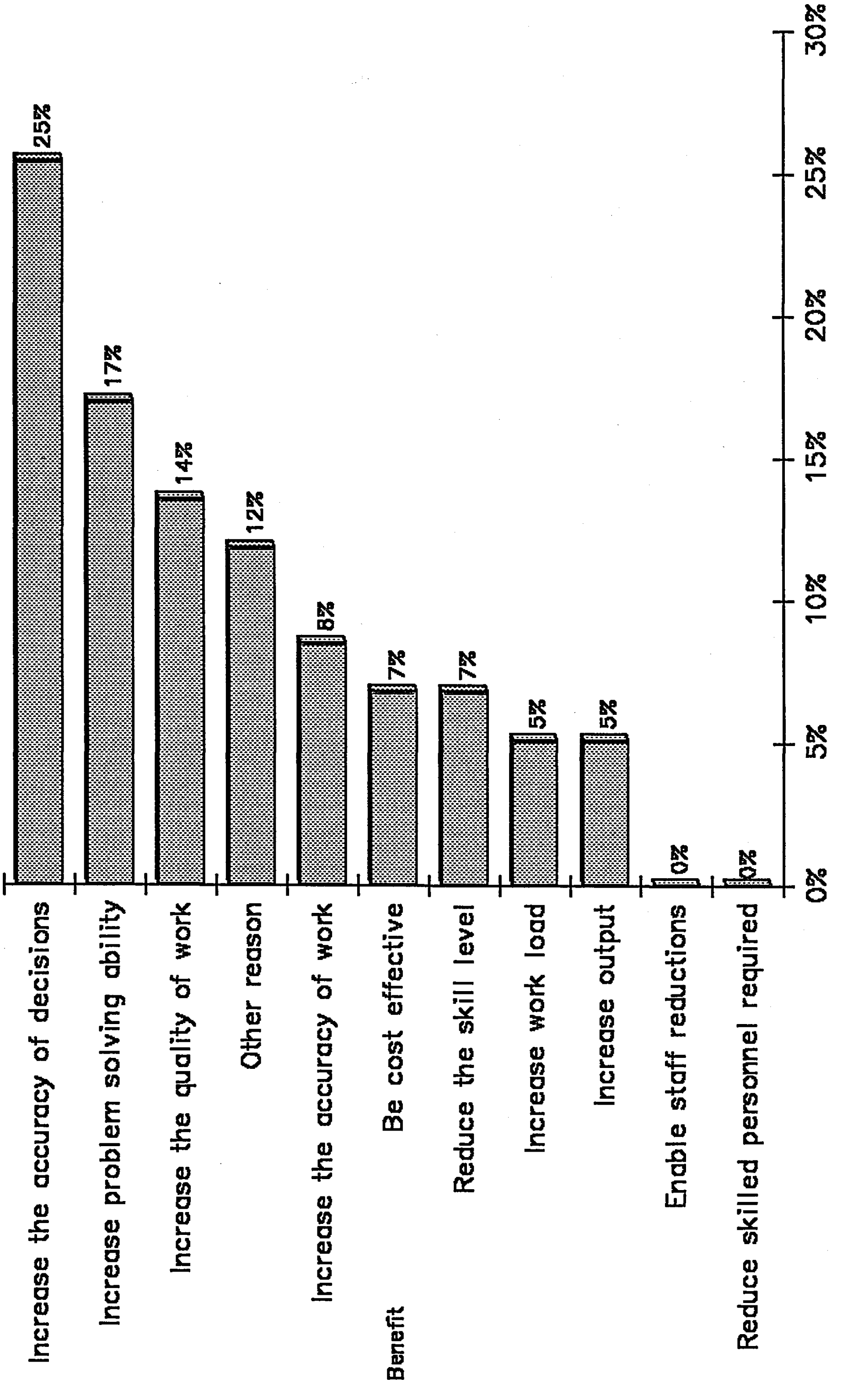
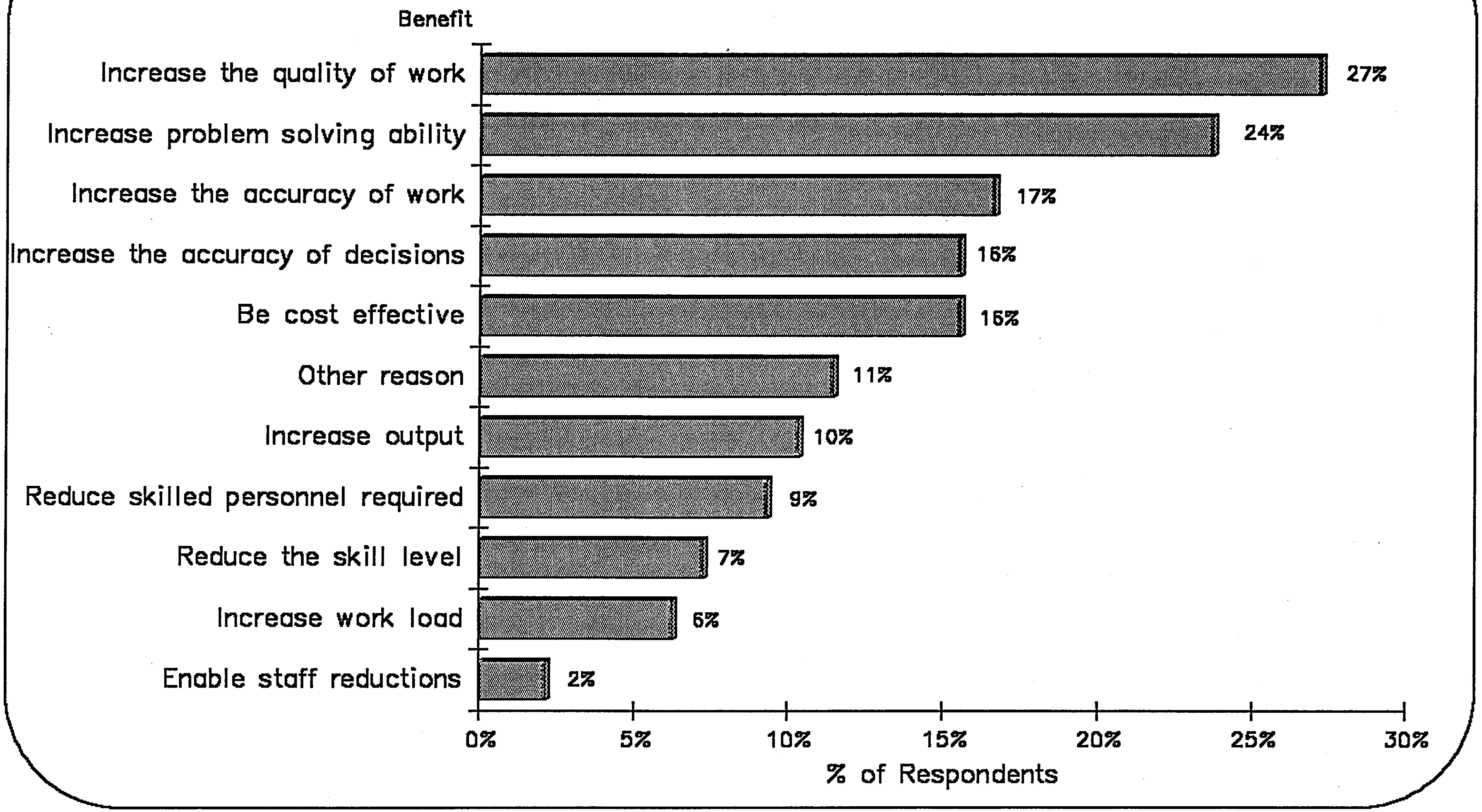


Figure F3: Key Benefits (Those Ranked 1, 2 or 3 in Importance)



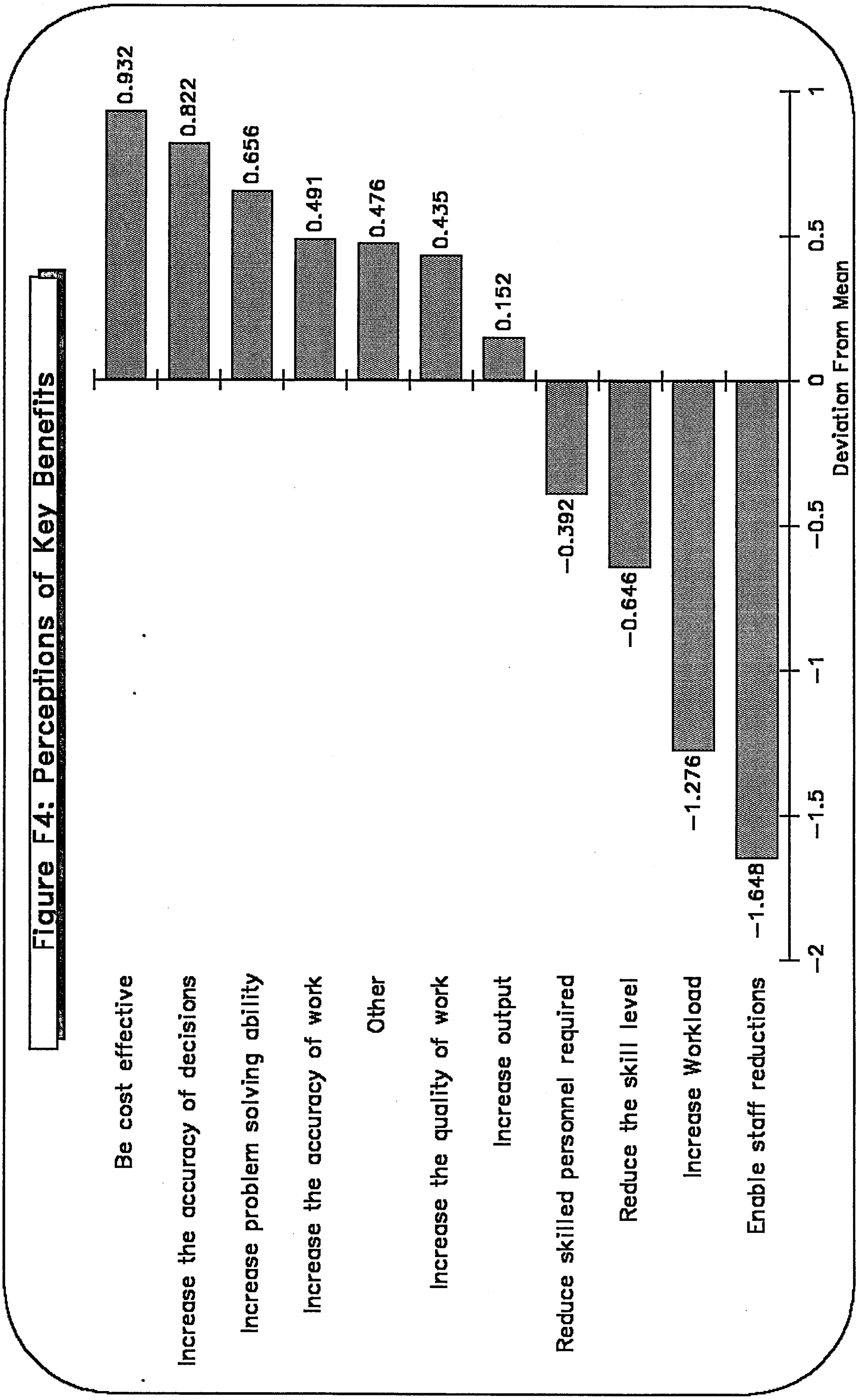


Figure F4: Perceptions of Key Benefits

Figure F5: Perception of Constraints

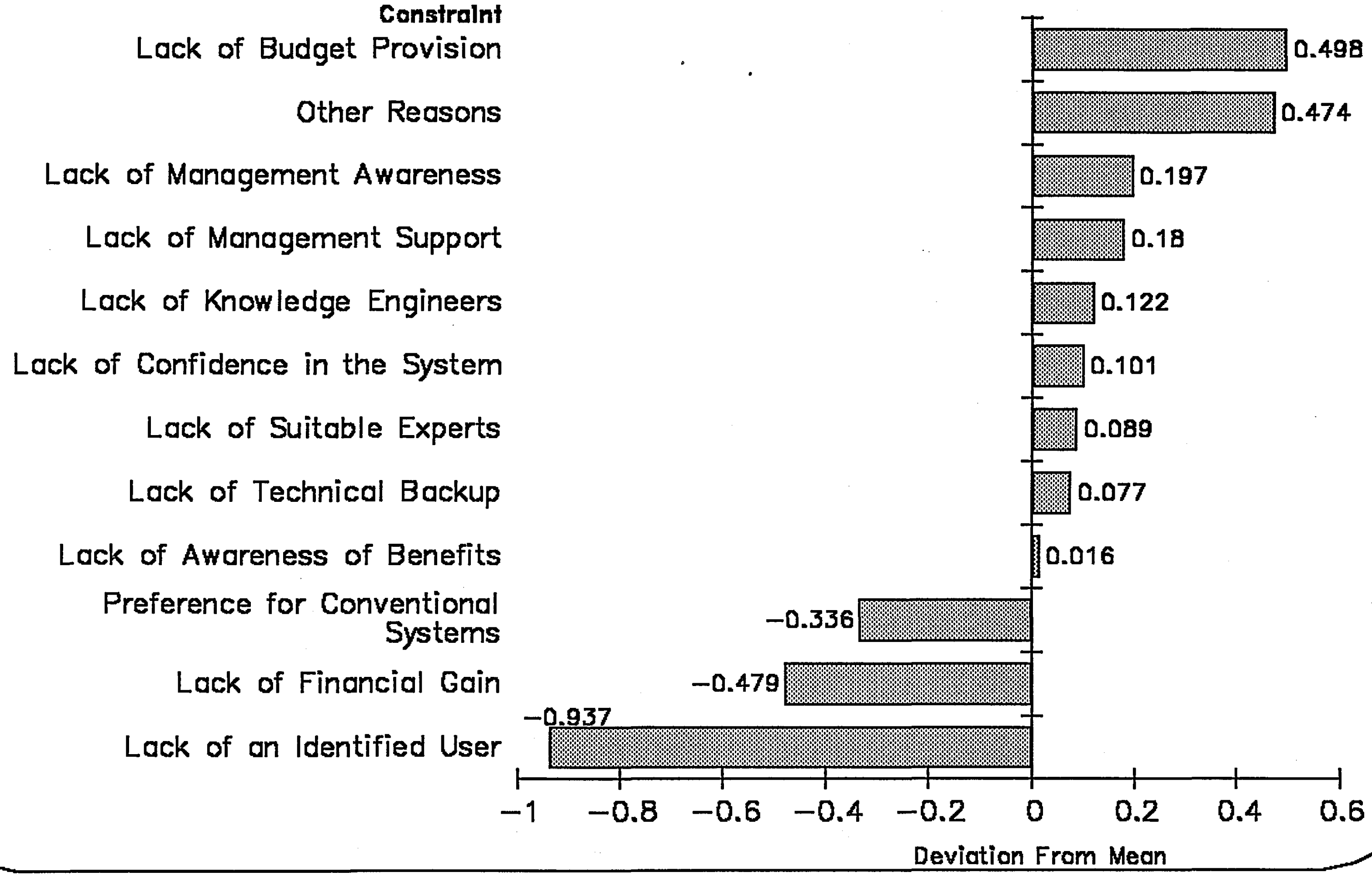
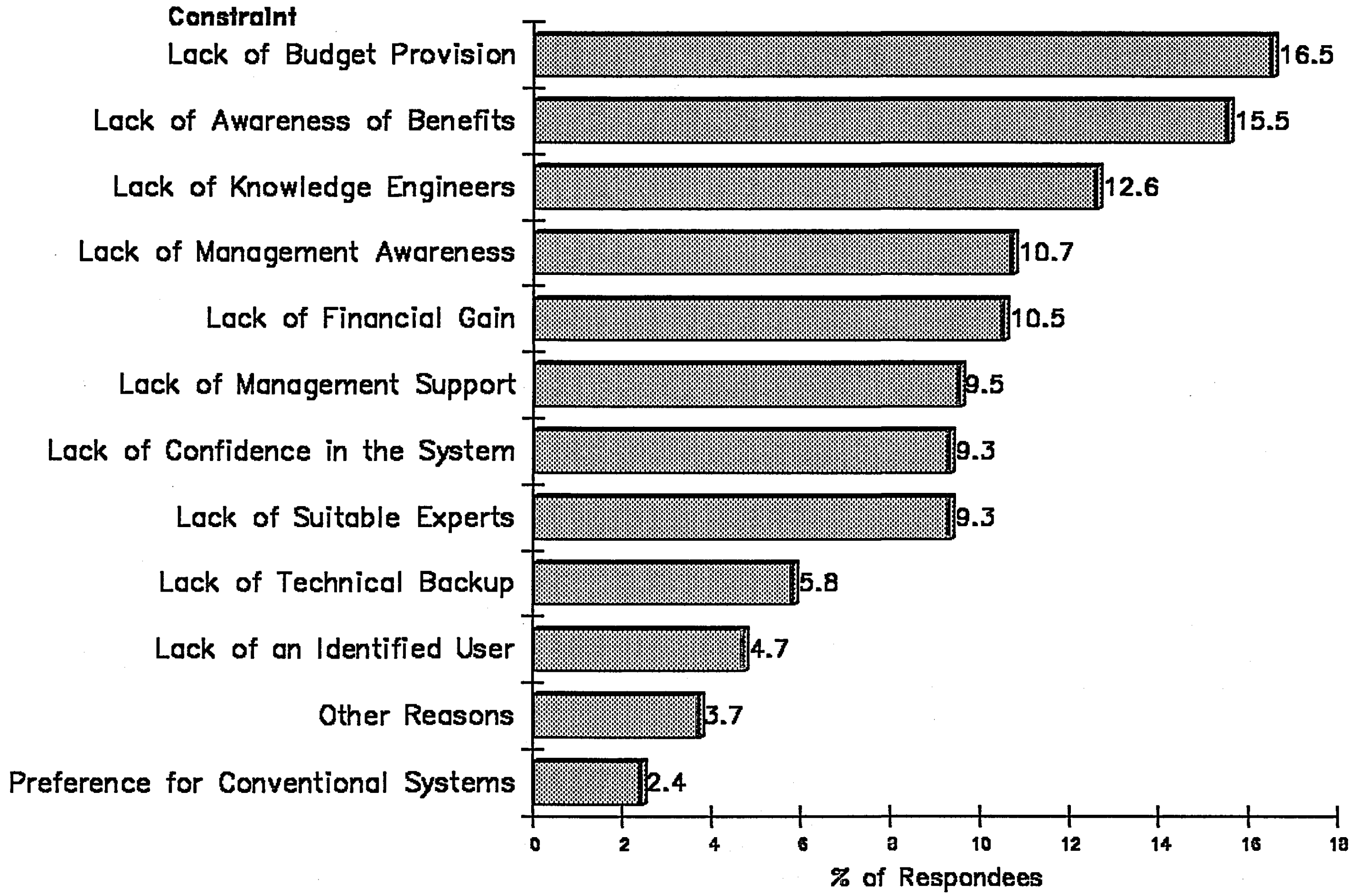


Figure F6: Key Constraints (those that prevented implementation)



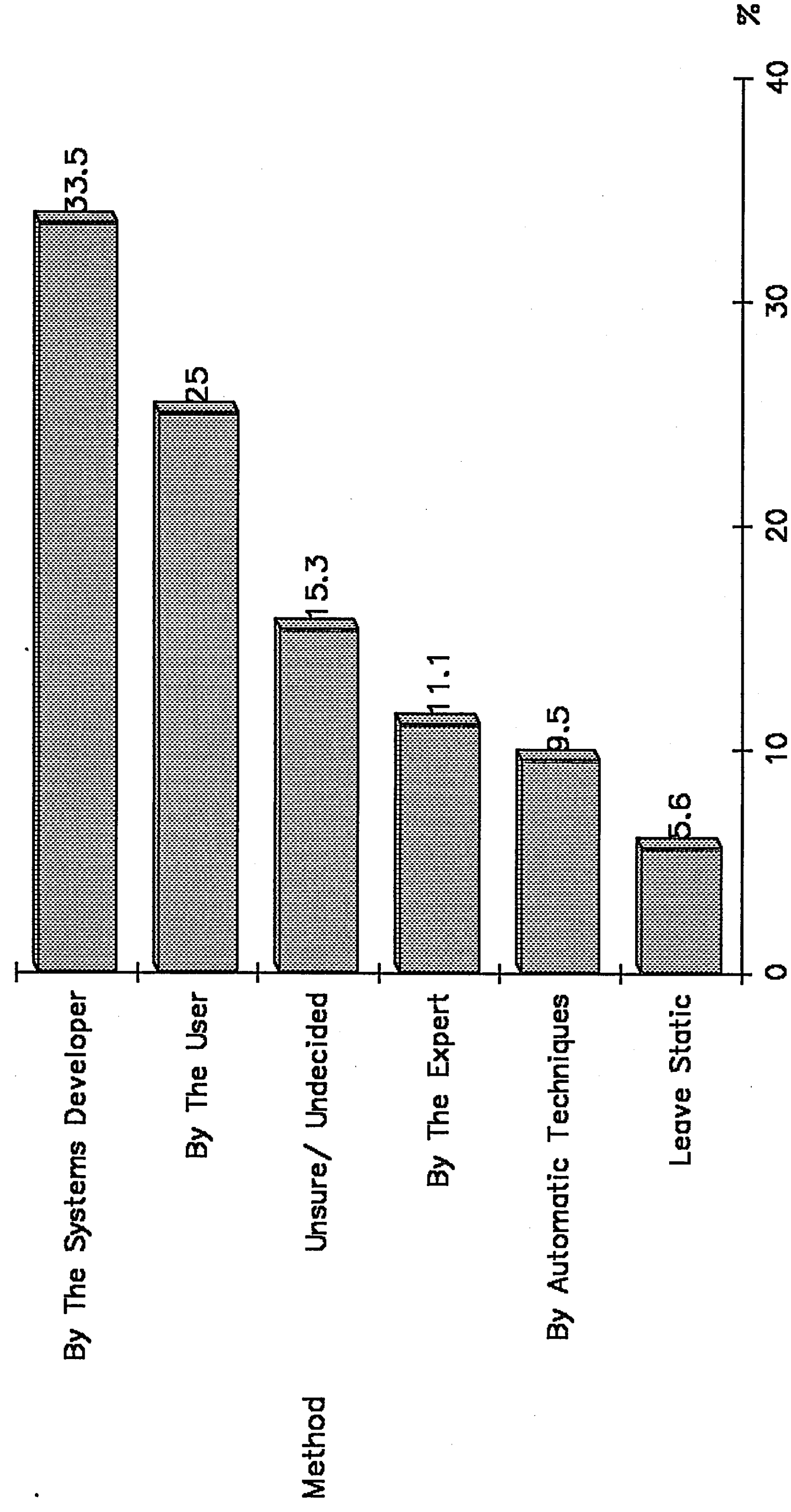


Figure G1: How Is The Knowledge-Base To be Maintained?

Figure G2: Views on Maintenance through the Development Lifecycle

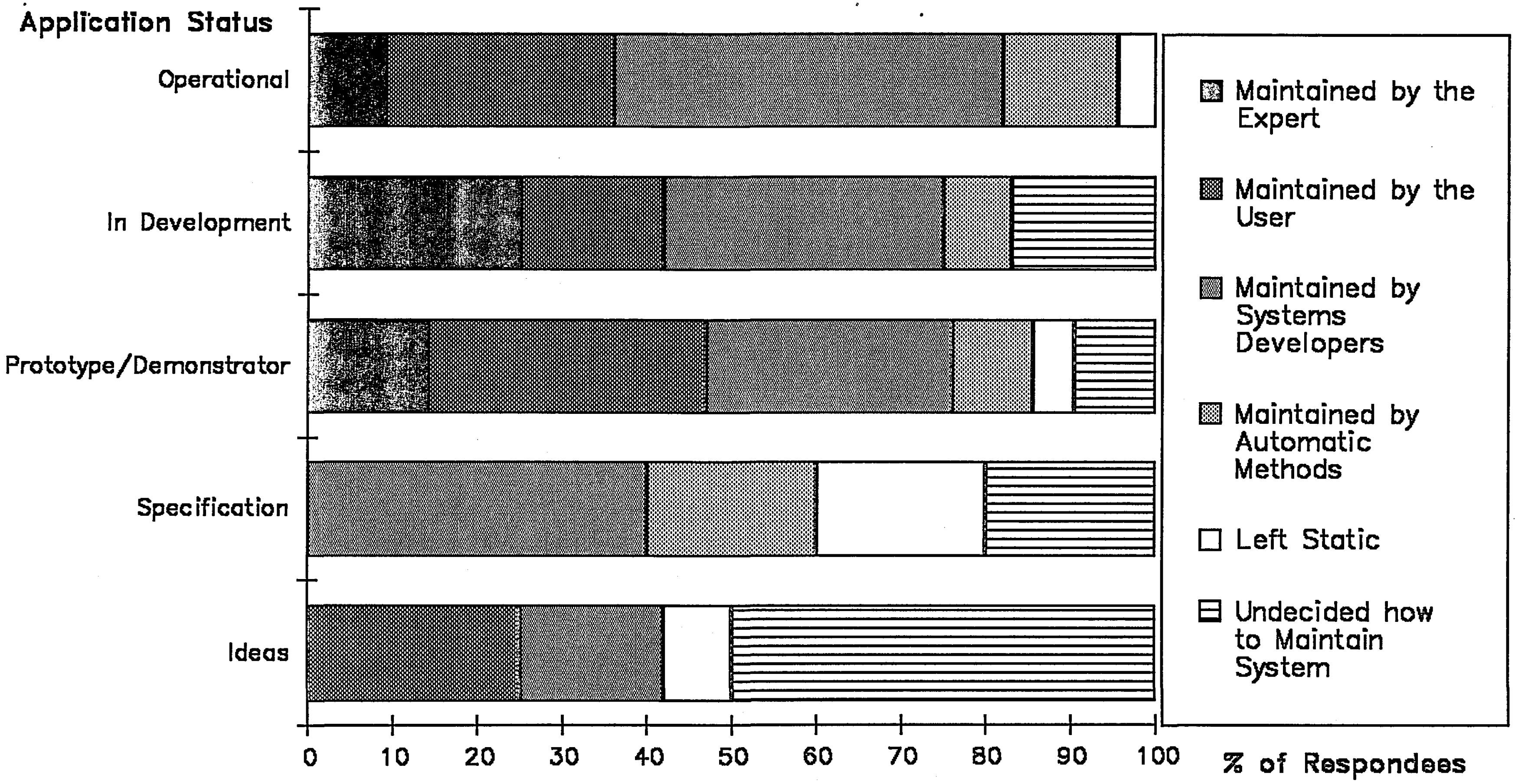


Figure G3: Maintenance Technique Against Responsibility for Development

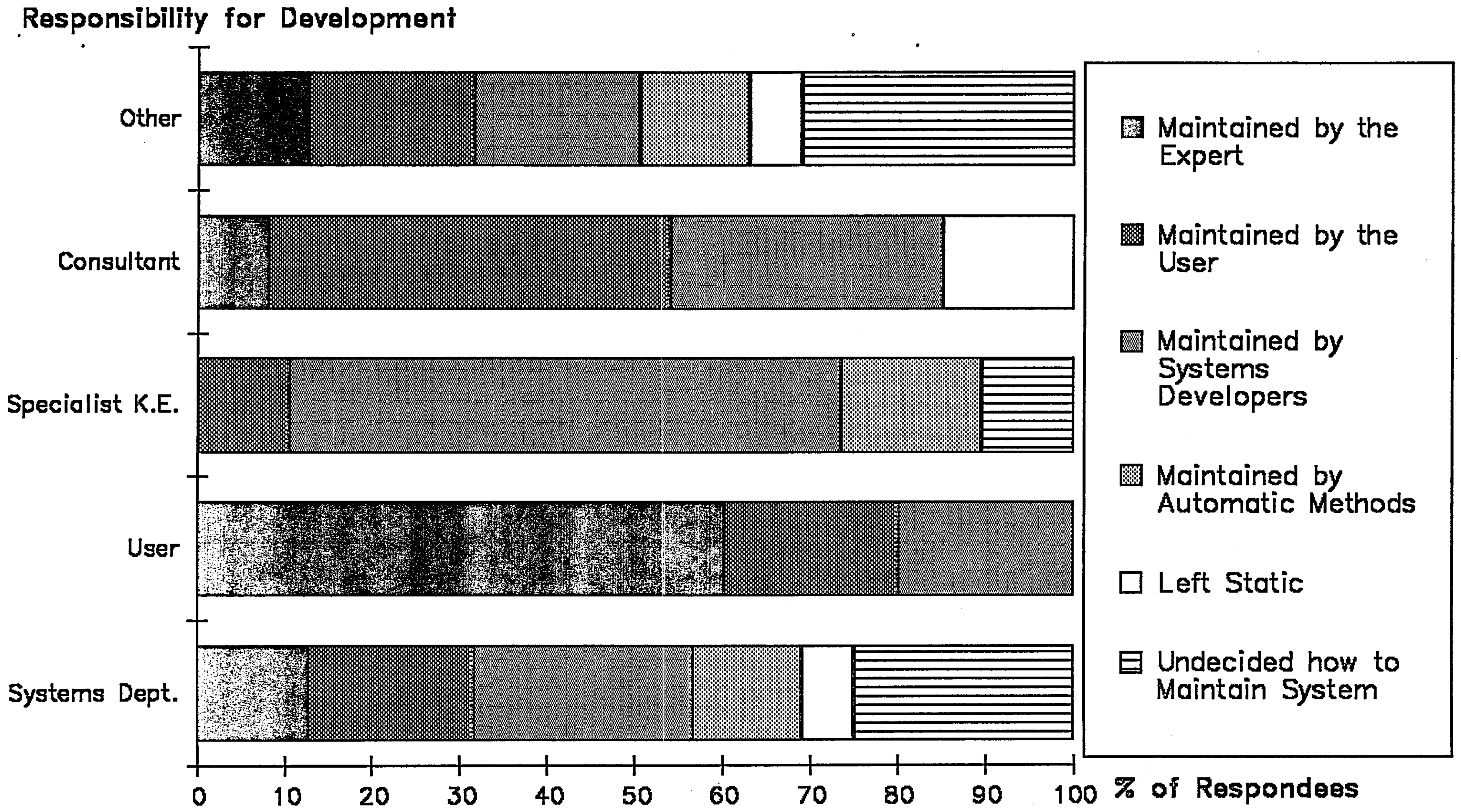


Figure G4: Responsibility for Maintenance Against Software Tool

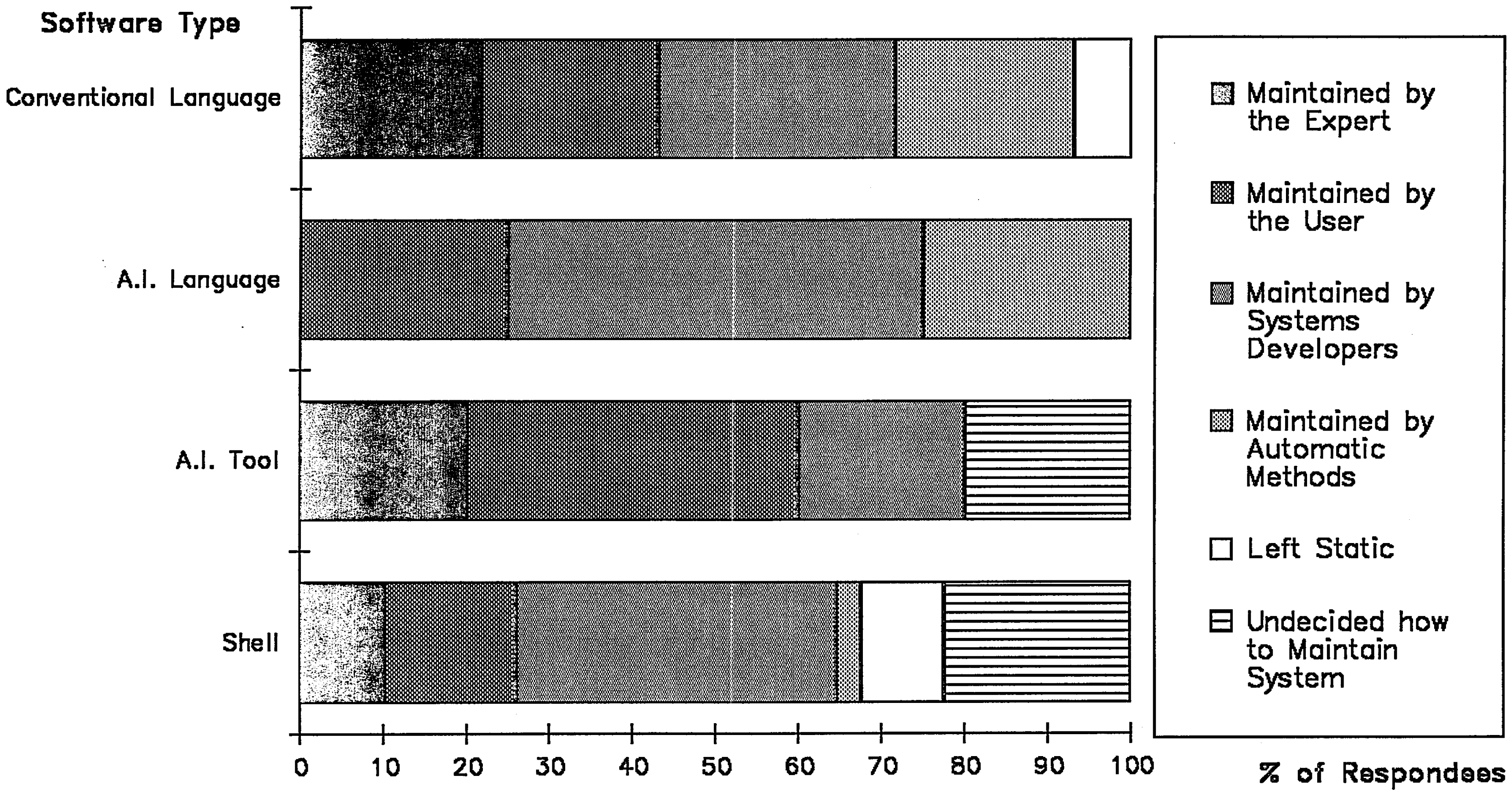
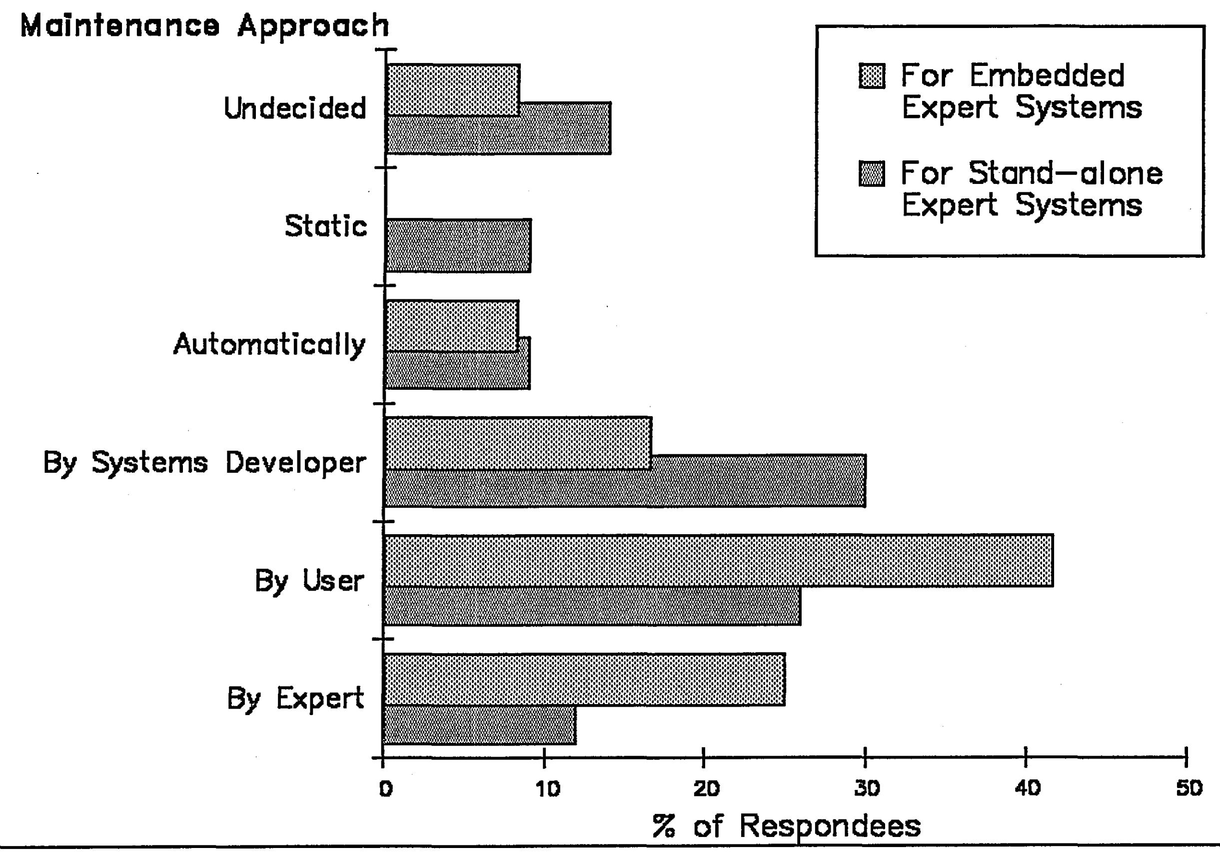
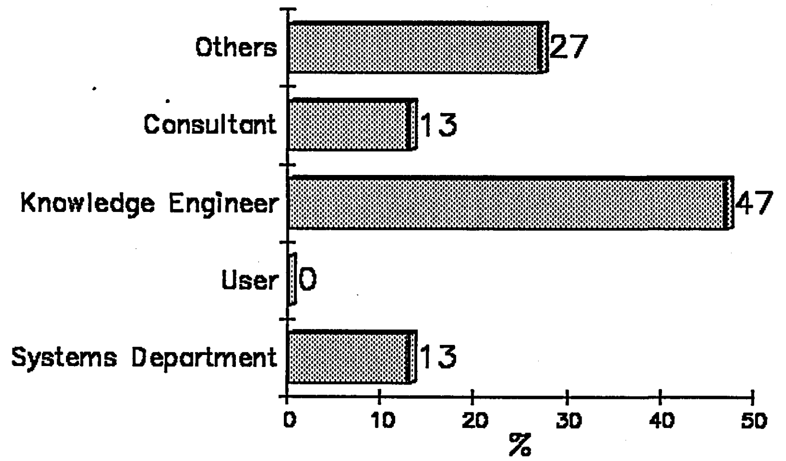


Figure G5: The Impact of Integration upon Maintenance Requirements



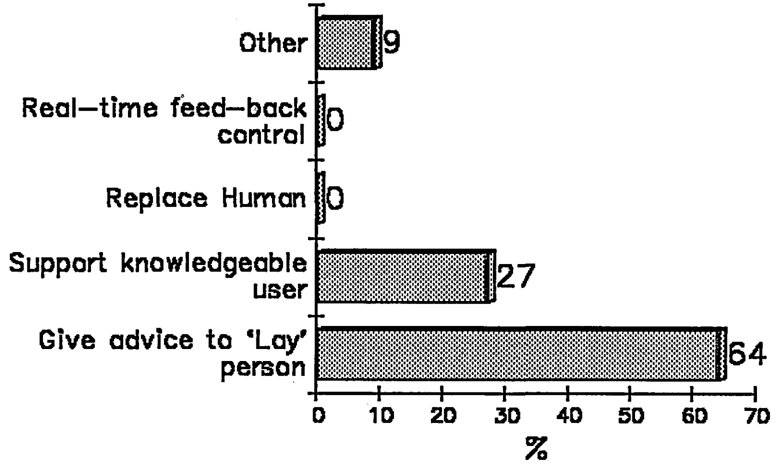
1

Figure H1.1 Responsibility for Development



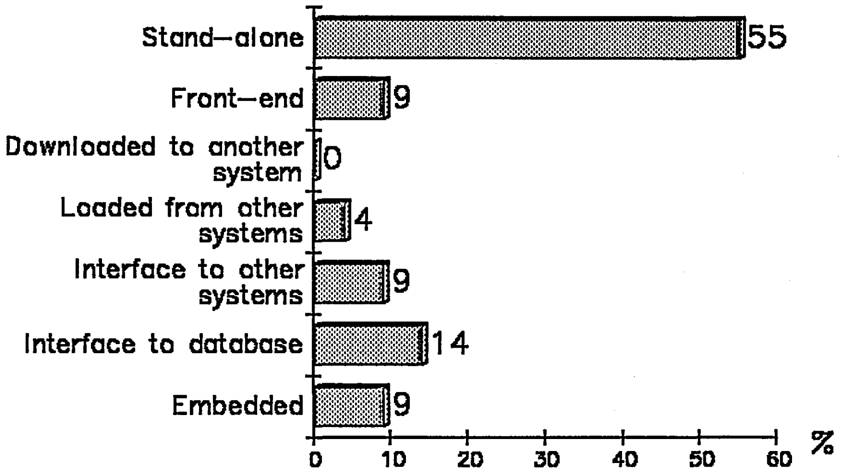
2

Figure H1.2 System Design Role



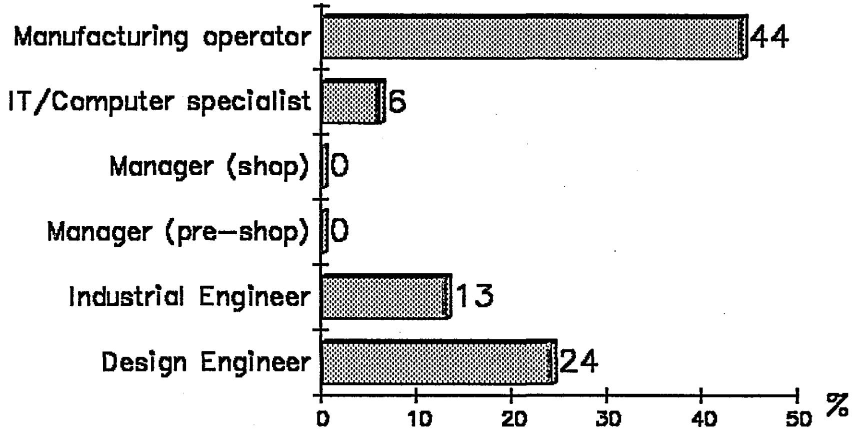
3

Figure H1.3 Level of Integration



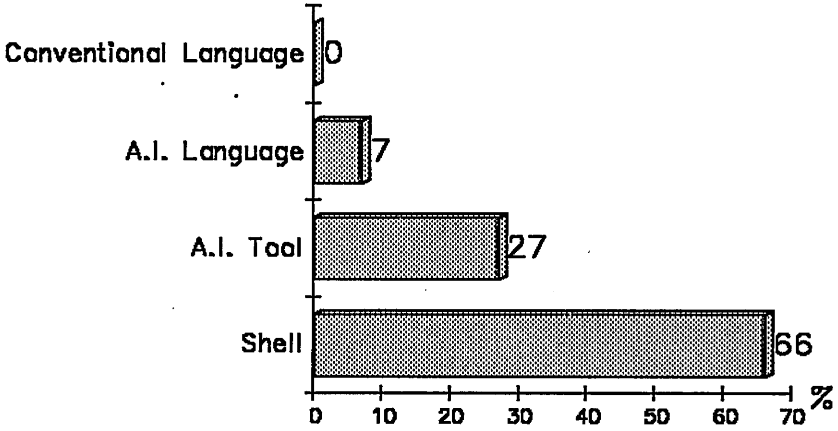
4

Figure H1.4 System user



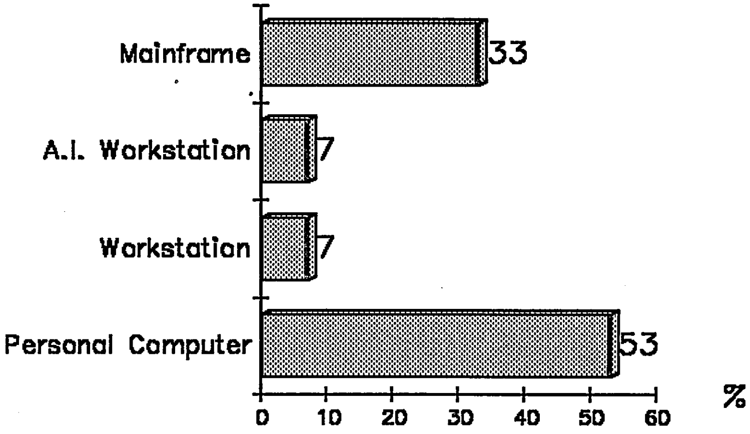
1

Figure H2.1. Choice of Software Tool



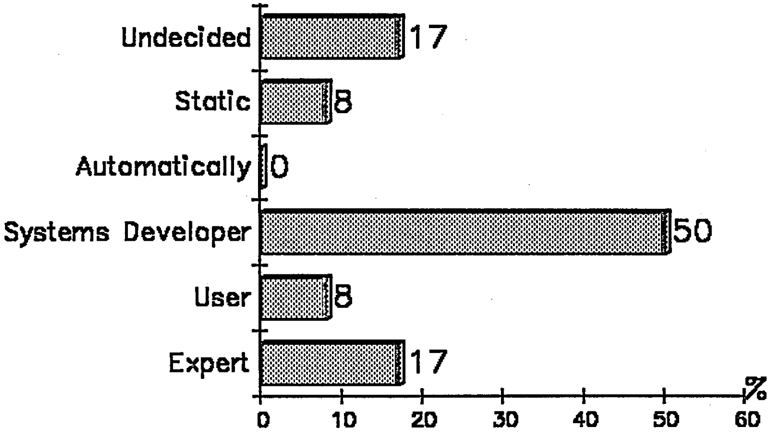
2

Figure H2.2. Choice of Hardware



3

Figure H2.3. Method of Maintenance



4

Figure H2.4. Method of Knowledge Acquisition