Towards a Methodology for Incorporating Human-Computer Interaction Protocols in Knowledge-Based Systems

A dissertation presented
in partial fulfilment of the requirements
for the degree of
Doctor of Philosophy in Computer Science
at Massey University

Elizabeth Angela Kemp

1995

Abstract

The research presented in this thesis describes the development of the FOCUS framework for use during the analysis stage of the knowledge-based system life cycle. The application of FOCUS (FunctiOns and Communication facilities for USers) helps the knowledge engineer to tackle the important human-computer interaction issues that arise when building knowledge-based systems.

The motivation for this research arises from the complexity of the interaction process. Firstly, the functions that users require to help them to achieve their goals have to be identified. Secondly, adequate communication facilities must be provided so that users can run the knowledge-based system, understand its problem solving capabilities and ask questions about the underlying domain. The situation is further complicated if users have little in common; their domain and/or computing backgrounds might be quite different. Analysis of the literature indicates that human-computer interaction is an issue of some importance but that detailed guidelines are often lacking.

FOCUS has been developed to assist the knowledge engineer during the analysis phase of the knowledge-based system life cycle. FOCUS has five stages: problem specification, preliminary analysis, user analysis, functional specification and detailed analysis. It recognises that the intended users of an expert system in an organisation may not all want the same problem-solving capabilities; the major user groups are identified and the functional requirements of each group specified. Communication issues can then be considered for each group. At the same time the analysis of the organisation's needs and elicitation of knowledge are not neglected.

By the end of the analysis stage, the knowledge engineer has completed the conceptual model with its three components: the model of expertise, model(s) of communication and user requirements. A comprehensive picture can be built up of the users' application, explanation and interface needs. The resulting user models together with the model of communication are the basis at the design stage for developing an interface to provide users with the desired functionality.

The FOCUS process has been evaluated using student enrolment at Massey University as the domain. The purpose of the case study is not to build a knowledge-based system but to assess the value of FOCUS It is suggested that a framework of this kind, for the analysis phase, should be structured, focused, open and practicable. Experience with FOCUS indicated that these criteria could all be met.

In summary, FOCUS integrates principles from the area of human computer interaction with a user-centred approach to knowledge-based systems development.

Acknowledgements

I would like to thank Professor Apperley, my chief supervisor, for his guidance and support throughout this research. In particular, I am grateful to him, during his time as head of the Department of Computer Science, for giving me the opportunity to carry out this research in a reasonable time frame.

I would also like to thank Chris Phillips, my second supervisor, for his sterling efforts in reading and correcting this thesis. The assistance of Elisabeth and Arthur Todd, John Hudson and Paul Clark is also appreciated. I am indebted as well to all the staff and students at Massey University who provided the data which enabled me to test out the ideas described in this thesis.

Finally, the love and support of my family has been crucial. I would like to thank Rebecca, Stephen and, especially, my husband Raymond, who always gave generously of his time to read the various drafts of the thesis.

Publications

The following publications all relate to the research carried out for this thesis:

Kemp, E. A. and Kemp, R. H. (1990). Integrating Expert Systems into an Information Systems Strategy. C. Yau (Ed.), *Proceedings of International Conference on Systems Management* (pp. 109-115), Hong Kong.

Kemp, E. A. (1990). Interface Issues in Expert Systems. *Proceedings of NZES 90* (pp. 145-158), Massey University, New Zealand.

Kemp, E. A. and Kemp, R. H. (1991). Explanation in knowledge-based systems: a user perspective. *Proceedings of IJCAI Workshop on explanation generation for knowledge-based systems* (pp. 57-80), The Netherlands: University of Twente.

Kemp, E. A. and Kemp, R. H. (1991). The management of the lifecycle in expert systems development. *International Journal of Information Resource Management*, 2(1), 11-23.

Kemp, E. A. (1992). Cognitive Ergonomics and the External Task. *Proceedings of 4th NZ Ergonomics Conference* (pp. 129-151), Massey University, New Zealand. Also in S. V. Burger and F. W. Darby (Eds.), *Human-Computer interaction in New Zealand* (pp. 50-70). New Zealand Ergonomics Society.

Kemp, E. A. (1992). Communicating with a knowledge-based system. P. Brezillan (Ed.), *Proceedings of ECAI-92 Workshop W15 " Improving the Use of Knowledge-Based Systems with Explanation"*, 92/91 (pp. 23-32), Paris: Institut Blaise Pascal.

Kemp, E. A. (1993). FOCUS: A User-Centered Approach to Expert Systems Development. In N. K. Kasabov (Ed.), *The First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems* (pp. 188-191). Los Alamos, California: IEEE Computer Society Press.

Kemp, E. A., Todd, E. G., da Silva, A. and Gray, D. I. (1994). Knowledge acquisition applied to farmer decision making. *Proceedings of SPICIS 94* (pp. B7-B12), Singapore.

Table of Contents

Figures and Tables

Figures

Tables

Part 1

Introduction and Surveys

Chapter 1

Introduction

1.1 The context of the research

Knowledge-based systems, as the name suggests, are programs that make use of real world knowledge and expertise (Hayes-Roth *et al.*, 1983). Systems of this kind have been developed from the mid 1960s onward. They are one of the first fruits of research into Artificial Intelligence (AI), a branch of Computer Science that aims to develop techniques that allow computers to perform tasks that apparently require intelligence when performed by humans (Tanimoto, 1987). Knowledge-based systems represented a move away from general problem solving approaches towards a reliance on domain knowledge. Some notable early systems include MYCIN (Shortliffe, 1976) for diagnosing and treating blood diseases, DENDRAL(Lindsay *et al.*, 1980) for assisting chemists to determine the structure of unknown compounds, PROSPECTOR (Gaschnig, 1982) which was used to look for mineral deposits (Hayes-Roth *et al.*, 1983) and R1 (McDermott, 1982) which enabled customer requests for VAX computer systems to be configured.

Often, those knowledge-based systems which embody the judgement, insight and experience of a human in some area of specialisation are referred to as expert systems (Hayes-Roth *et al.*, 1983; Bielawski and Lewand, 1991; Nijholt, 1991; Turban, 1992). This term has been viewed as pretentious and misleading (Wielinga *et al.*, 1987). Wielinga *et al.* prefer to talk of knowledge-based systems as this emphasises the central role of domain knowledge (as opposed to academic expertise) when solving problems. Moreover, not only expert systems but also Hypermedia programs and Computer-aided software engineering (CASE) programs can be described as knowledge-based (Tuthill, 1990). Nevertheless, many researchers do not make such fine distinctions between knowledge-based and expert systems and use the terms as synonyms (Kirakowski, 1988; Hickman *et al.*, 1989; Coyne *et al.*, 1990; Gonzalez and Dankel, 1993). This practice is followed here.

1.2 Interaction issues in knowledge-based systems

Those with the responsibility for constructing knowledge-based systems are usually referred to as knowledge engineers (Buchanan *et al.*, 1983). They obtain the requisite domain information, generally from experts, and incorporate it into a system for others to access. System builders, such as knowledge engineers and experts, have quite different interaction requirements from run-time users (Young, 1989). For instance, knowledge engineers need debugging facilities whilst experts may use programs (knowledge acquisition software) to input facts directly into the knowledge base. Run-time users, on the other hand, have to be able to utilise the problem solving capabilities of the knowledge-based system. It is the interaction requirements of this last group, for simplicity referred to as users, that are dealt with in this thesis.

Interaction has been defined as

> *any communication between a user and a computer............The important thing is that the user is interacting with a computer in order to achieve something.*
>
> (Dix *et al.*, 1993, p.3)

Assuming that users of knowledge-based systems wish to accomplish certain goals then two issues arise. Firstly, what functions do the users require to enable them to achieve their objectives? Secondly, how can adequate communication facilities be developed so that users can run the knowledge-based system, understand its problem solving capabilities and ask questions about the underlying domain? The situation may be further complicated by user differences with regard to computer background and/or domain knowledge (Cleal and Heaton, 1988). In these circumstances, the interface is even more important than is usually the case with computer software (Hendler and Lewis, 1988).

It is researchers in the area of human-computer interaction (HCI) who concentrate on the development of principles, guidelines and methods to assist with the design and evaluation of interactive systems (Johnson, 1992). They are aware, though, of the difficulties of developing an interface for an expert system. Kirakowski (1988, p.203), for example, distinguishes between two meanings of the term "interface". Whilst the term is commonly used to describe the styles with which users and computers interact, in knowledge-based systems the interface also refers to the flow of knowledge. Users not only have to be able to interact with the software effectively but may also require access to the problem solving processes and the domain knowledge of the expert system.

Shneiderman comments, in the context of expert systems development, that

> *The user interface is not the paint put on at the end of a project, but the steel frame on which to hang details.*
>
> (Shneiderman, 1988, p.x)

If the interface is, as it were, the framework for the whole system then how should such a structure be built? How can a developer know whether the right details have been put into place? These are important questions since, if user needs are ignored, then knowledge-based systems may be seldom or never used (Kemp and Boorman, 1987; Hendler and Lewis, 1988; Kemp and Kemp, 1991b; Breuker and de Greef, 1993).

1.3 The need for a framework

The failure to meet user needs is not one that is particular to the knowledge-based area. Pressman (1992, p.18) observes that "Software development projects are frequently undertaken with only a vague indication of customer requirements." Approaches to system development that allow user concerns to be taken into account include the classic life cycle (Senn, 1989), prototyping, that is building a model of some aspects of a system (Pressman, 1992) and the spiral model (Boehm, 1988). There is the additional complexity with knowledge-based systems of how to represent a domain in such a way that typical problems in the area can be solved. For this reason many emphasise the importance of prototyping. Pressman (1992) points out that a prototype can be a paper based or working model. In the knowledge-based literature, such a distinction is not usually made and a prototype refers to an executable system, see for example Hayes-Roth *et al.* (1983), Harmon and King (1985), Guida and Tasso (1989), and Gonzalez and Dankel (1993).

According to Turban, a prototype is essential when developing large systems "because the cost of a poorly structured, and then, not used expert system can be very high" (Turban, 1992, p.421). There is a major debate about the value of this approach. The substitution of prototyping for a thorough understanding of the problem can lead to considerable backtracking since this procedure may fail to uncover the basic structure of the expertise (Wielinga *et al.*, 1987).

It was to avoid such difficulties that the KADS methodology (originally an acronym for Knowledge Acquisition and Documentation Structuring) was developed for the commercial sector (Wielinga *et al.*, 1986). Such a methodology, it was believed, would allow businesses to embark confidently on large-scale programmes for knowledge-based systems development (Hickman *et al.*, 1989). KADS has the advantage that it does not get prematurely physical or too dependent on experimentation. Both the spiral model and an adaptation (for knowledge-based system purposes) of the classic life cycle are recommended as suitable paradigms for structuring the development process. Either can be supplemented by the limited use of prototyping.

The KADS methodology essentially encourages the knowledge engineer to cope with the complexity of developing a knowledge-based system by building models of the real-world phenomena under investigation (Hickman, 1989). Models built during the analysis phase take no account of implementation details. Rather, the knowledge engineer seeks to describe, during the analysis phase of the KADS life cycle, organisational and user requirements as well as domain knowledge. KADS is virtually unique in prescribing a detailed analysis stage dedicated to modelling the user, organisation and the domain. Generally, user concerns are dealt with by prototyping (Waterman, 1986; Johnson and Johnson, 1992). There are other approaches that also emphasise the role of the organisation and users (Diaper, 1988; Basden, 1989) but they lack the commitment to modelling which underpins KADS. It is this extensive modelling that allows the knowledge engineer to build an abstract representation of the domain for problem solving purposes which is independent of any computing considerations.

Unfortunately, for all its strengths, KADS does not provide detailed guidelines to assist the knowledge engineer specify user interaction requirements. Nor can it demonstrate, therefore, how to link user modelling with organisation and domain modelling. Consequently, this gap needs to be filled by developing a framework or process for use in the analysis phase of a knowledge-based systems life cycle. The approach proposed would describe the activities for each stage as well as detailing the models that should be produced. It cannot be described as a methodology since, as Harmon and Hall (1993) observe, a methodology requires guidelines for project management such as checklists, and techniques to calculate costs and milestones. The proposed approach is not as far-reaching as this since its objective is to focus on user issues during the analysis phase of a knowledge-based system. Nonetheless, the proposed framework has some features in common with a methodology (the sequencing of activities and the specification of the models required, as with KADS) and can be evaluated in much the same way. Guida and Tasso (1989, p.9) specify the criteria that they believe a methodology for managing the complete expert systems development life cycle should meet. It should be:

- *structured and modular*, supporting the identification of elementary components;
- *complete*, supporting the designer in every way during the expert systems development process;
- *effective*, supporting the planning and control of project development;
- *efficient*, that is easy to employ without incurring a large overhead;
- *practical*, that is easy to teach and apply in a large variety of contexts;
- *flexible*, that is able to accommodate projects of different size and complexity;
- *explorative* [sic], allowing for iterative system specification and design;
- *focused*, emphasising those tasks typical of expert systems development: domain and problem analysis, knowledge acquisition, knowledge modelling and knowledge verification;
- *open*, supporting the use and integration of existing techniques and tools.

These are exacting requirements, some of which would be difficult to check in practice. For example, to substantiate claims that a methodology is efficient, practical and flexible would require large-scale trials on several projects using the specified methodology and at least one other approach. Furthermore, some of these criteria cannot be applied to a process for the analysis phase only. For instance, it cannot be complete and effective as these terms are defined above since overall project development is beyond such a framework's scope. Nevertheless, some of the criteria are relevant. The outputs from the analysis phase should be well structured in order to lay a successful foundation for the design and implementation stages. To be of any value, the framework also has to be both focused with an emphasis on expert systems issues and open so that suitable tools and techniques can be used when necessary.

1.4 Objectives of this research

The principal objective of this research is to develop a framework for the analysis phase of a knowledge-based systems life cycle based on important concerns identified by HCI and knowledge-based systems researchers. Such a process should enable both the functions required by users and their interface requirements to be specified in addition to domain and organisational requirements. This framework should employ the modelling techniques advocated by KADS and make use of but not depend upon the prototyping approach.

This process is designed to be structured, focused, and open, where:

- *Structured* means that domain knowledge should be described in such a way that the outputs of the analysis phase can easily be used during design.

- *Focused* is defined as the requirement to analyse both the problem and the domain.

- *Open* indicates that project management techniques and knowledge acquisition tools may also be employed. It is not the purpose of this thesis, however, to show how these can be fully integrated with the framework described.

Ideally, the approach should also be efficient, practical and flexible. Whilst endeavouring to meet these objectives, it is impossible to test out such claims within the scope of one PhD thesis. It is more realistic to check that the framework is practicable. Following the Illustrated Oxford Dictionary

(Coulson *et al.*, 1976, p.662), *practicable* can be defined as "feasible, that can be done." The secondary objectives of this research, therefore, are to develop a process which is structured, focused, open and practicable.

The framework is evaluated using a case study that starts with the selection of a problem and ends with the specification of the interface. The value of the illustrative case study for evaluation purposes has been recognised by Yin (1989). This approach allows the author to demonstrate to what extent the process meets the objectives specified. It is not the purpose of this research, it should be noted, to build a knowledge-based system but to develop and assess a process.

1.5 Thesis overview

The approach taken in this research is firstly to determine the significance of interaction issues and secondly to ensure that relevant concerns are handled by a framework that a knowledge engineer can use in the analysis phase of a knowledge-based systems life cycle.

Chapter 2 establishes the major characteristics of a knowledge-based system and describes the problems of developing systems for users with differing requirements. The HCI interface literature is consulted for guidelines and the way that ideas and principles from this area have been applied to knowledge-based systems development is discussed.

Students with experience of using an expert system for diagnosing diseases in apple crops were asked for their views on interface presentation styles and explanation facilities. The results of this survey are described in Chapter 3.

Chapter 4 presents several approaches to expert system development with emphasis on the analysis phase. It is argued that it is at this stage that user concerns should be taken into account. The major debate about the role of prototyping is also addressed. Finally, some important principles for knowledge-based systems development are identified.

Chapter 5 introduces the FOCUS process. The five stages of FOCUS: problem specification, preliminary analysis, user analysis, functional specification and detailed analysis, are described in detail with all major inputs and outputs defined. The reasons for developing FOCUS are also discussed.

In Chapters 6 and 7 a case study is worked through in considerable detail. The problem domain concerns student enrolment at Massey University. FOCUS is applied and all relevant models developed. Student requirements are elicited by a major study involving more than 400 participants. The model building process is reviewed to establish whether it meets the objectives described in chapter 5.

Chapter 8 demonstrates how outputs from the analysis phase can be used for interface design purposes. Control is built into the model of communication, the major input and output flows are identified, the sequence of screens produced and storyboards developed.

In Chapter 9 the main conclusions of the research and suggestions for future work are discussed.

The thesis is divided into three parts. Part 1 includes the introductory chapter, Chapter 2 which considers interaction issues in the context of knowledge-based systems, Chapter 3 which presents user views on the interface and Chapter 4 which examines the literature on the knowledge-based systems life cycle. In part 2, FOCUS is described and applied to a case study (Chapters 5, 6, 7 and 8). Part 3 contains a single chapter, Chapter 9, which reviews the contribution made by the research and includes suggestions for future work.

Chapter 2

Communication Issues in Knowledge-based Systems

In this chapter the term knowledge-based system is defined and the main components of such a system are described. The problems that arise when building an interface for users with differing domain and computing backgrounds are outlined. In order to see how these may be overcome, the HCI literature is examined for guidance. Finally, the way that ideas and principles from this area have been applied to knowledge-based systems development is discussed.

2.1 Knowledge-based systems

Many computer systems have been built which attempt to capture and make use of human knowledge about a domain. These knowledge-based or expert systems utilise ideas and techniques derived from AI and are the first type of programs from that area of study to have wide commercial success (Cleal and Heaton, 1988). Much of AI is concerned with the application of symbolic, non-algorithmic methods to problem solving. Since people usually communicate with each other using words (apple, orange, etc) to represent concepts, the symbolic approach has the advantage of handling information in the same way as human beings. People's knowledge of topics is usually represented this way rather than mathematically or quantitatively (Buchanan and Shortliffe, 1984a).

There are many descriptions of expert and knowledge-based systems in the literature (Brachman *et al.*, 1983; Hayes-Roth *et al.*, 1983; Harmon and King, 1985; Bonnet *et al.*, 1988; Coyne *et al.*, 1990; Jackson, 1990; Tuthill, 1990; Lucas and Van Der Gaag, 1991; Turban, 1992; Gonzalez and Dankel, 1993). A useful starting point is to consider the views of the developers of MYCIN, one of the most influential of these kinds of systems. Buchanan and Shortliffe believe that an expert system should provide expert level solutions, be understandable and be flexible enough to accommodate new knowledge easily (Buchanan and Shortliffe, 1984a).

These are quite general requirements. Brachman *et al.* (1983), in the important text "Building Expert Systems", define an expert system at much greater length in the terms of seven semi-independent features:

- *expertise*, an expert system should successfully and efficiently solve the problems to which it is applied;

- *symbol manipulation*, an expert system should use symbolic knowledge when reasoning;

- *general problem-solving ability in a domain*, an expert system should display breadth of coverage and be able to reason within the domain;

- *complexity and difficulty*, an expert system should solve problems of some magnitude;

- *reformulation*, an expert system should be able to model the domain knowledge;

- *abilities requiring reasoning about self*, an expert system should be able to examine how it works and explain its processes;

- *task*, the architecture of the expert system should be related to the task being carried out;

These criteria are discussed in some detail and provide a comprehensive analysis of what is involved in building a system. They can be used to assess whether an expert system should be developed and, if so, its performance.

2.1.1 Knowledge-based systems architecture

The main components of an expert system include a knowledge base, an inference engine, a user interface, explanation and knowledge acquisition subsystems (Harmon and King, 1985). These modules and the relationships between them are shown in Figure 2.1. An expert system can be viewed as a model of some problem solving process with the domain knowledge base and inference engine together representing the capabilities of the expert (Hendler and Lewis, 1988; Lowgren, 1989). The knowledge base contains a suitable model of the problem domain with the facts and rules that embody the domain knowledge normally represented in one or more of the following ways: predicate logic, semantic networks, objects, frames or production rules (Harmon and King, 1985; Gonzalez and Dankel, 1993). Knowledge about objects in the domain and relationships between them provides the core of an expert system. Heuristics (rules of thumb) can supply the insights that distinguish the expert from the novice. Since there are many sources of uncertainty, heuristics have to handle this, often using either Bayesian or fuzzy logic (Hayes-Roth *et al.*, 1983).

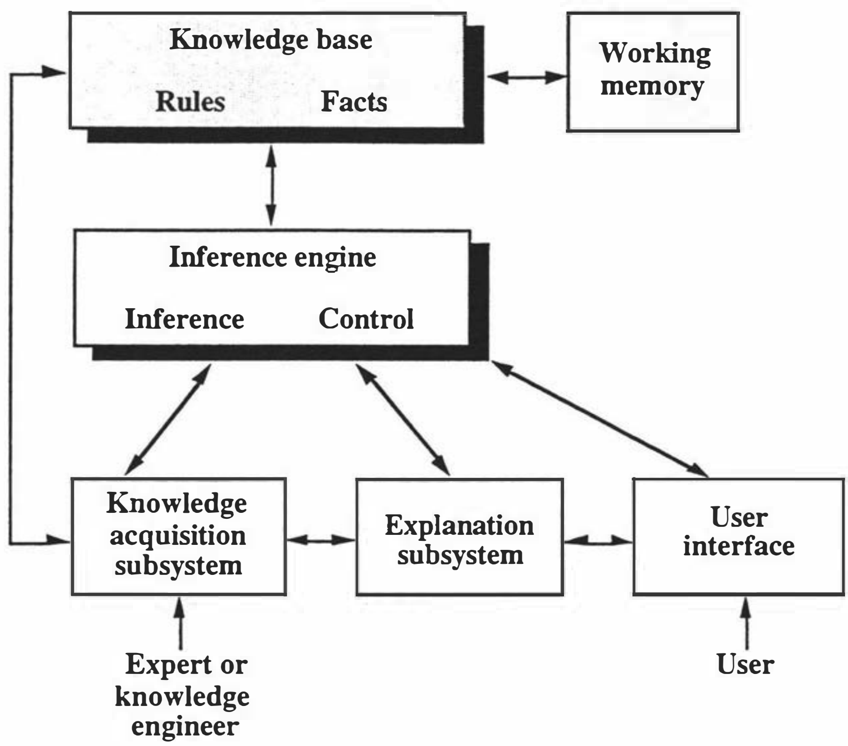


Figure 2.1 Expert systems components (Harmon and King, 1985)

The inference engine drives the expert system, processing the information in the knowledge base to solve the particular problem. Backward chaining, forward chaining and inheritance are all mechanisms that allow a conclusion to be inferred from facts input by the user and the expertise stored in the knowledge base (Harmon and King, 1985; Gonzalez and Dankel, 1993).

It is through the interface that a user can input information and obtain a response, the recommended treatment, for instance, in a medical expert system. Typically, a user may interrogate the system to find out how and why conclusions were reached. Such replies are generated by the explanation subsystem. The knowledge that is needed to answer questions and to solve the problem may be entered via the knowledge acquisition subsystem.

2.1.2 Knowledge-based development tools

Knowledge-based systems can be built using different types of software including programming languages, environments and shells (Hickman *et al.*, 1989; Jackson, 1990; Turban, 1992). Numerous programming languages can be used for the construction of knowledge-based systems including COBOL and 4GLs such as Lotus 1-2-3. TIMM was written in FORTRAN 77, INSIGHT 2 in Turbo Pascal and EXSYS in C (Turban, 1992). Specialised AI languages such as LISP and PROLOG have also been widely employed. Recently, object-oriented languages such as C++ are finding favour (Mehandjiska and Page, 1993). Finally, several languages have been developed specifically for constructing knowledge-based systems. These include HEARSAY-III (Erman *et al.*, 1981), ROSIE (Hayes-Roth *et al.*, 1983) and OPS5 (Brownston *et al.*, 1985).

Knowledge-based systems are often built using a shell, an expert system development tool that already has an inference engine and a knowledge base manager. Only domain knowledge needs to be added. Both general and domain specific shells are on the market. The distinction between these and what are termed *hybrid environments* is that the latter allow knowledge to be represented in a variety of ways (Turban, 1992). ART, KEE, Nexpert Object and Kappa PC are representative of this class. Nexpert Object allows the user to interface with databases, neural networks, spreadsheets and conventional application languages such as COBOL and C. It also has powerful interface prototyping capabilities (Neuron Data, 1993).

2.1.3 Knowledge acquisition

Knowledge has to be gathered and put into a form that is suitable for computerisation. This process is referred to as knowledge acquisition (Buchanan *et al.*, 1983; Gaines, 1987; Kidd, 1987; Hickman *et al.*, 1989; Wielinga *et al.*, 1991). Documents and people can both be used as sources of information (Cordingley, 1989). Many techniques are available to the knowledge engineer for obtaining information from experts and anyone else with relevant domain knowledge. Cordingley (1989) lists twelve of these: interviewing, focused talk, teach back, repertory grid, sorting, laddering, "20 questions", matrix generation, critiquing, protocols, role play and simulation. She discusses at length the principal technique of knowledge elicitation, interviewing experts, and describes the pitfalls involved in recording or taping the proceedings. Other comprehensive reviews of these techniques are provided by Hart (1992) and Gammack (1987).

Criticism has been made of manual methods of knowledge elicitation on the grounds that it is difficult to validate the acquired knowledge since there may be a weak correlation between mental and verbal reports; experts do not always know how they have arrived at their decisions and knowledge engineers do not fully understand the problem domain (Turban, 1992). Two automated approaches are advocated as alternatives to the manual methods. In the first, experts using interactive software can enter information directly into the computer. Examples of such elicitation tools include OPAL (Musen, 1989), SALT (Marcus, 1988), AQUINAS (Kitto and Boose, 1989) and KSSO (Gaines and Linster, 1990). These software tools may be domain dependent (SALT and OPAL) or domain-independent (AQUINAS and KSSO). The second means of automating the knowledge acquisition process is to enable computers to learn, as it were, domain concepts from examples. Various induction algorithms such as ID3 have been developed for this purpose (Quinlan, 1982). Automated methods may speed up the knowledge acquisition process, eliminating or reducing the need for a knowledge engineer (Turban, 1992). Gonzalez and Dankel (1993), for instance, state that induction algorithms can be helpful in the development of small systems where an expert acts as the knowledge engineer. Automated approaches, however, are limited in their scope. They do not deal with the situation that arises when multiple sources of information are needed.

For the knowledge engineer, who has to handle many sources of knowledge, a systems analysis approach has much to offer (Brule and Blount, 1989). From this perspective, there are four ways a knowledge engineer can obtain information: observation, interviewing, reviewing documentation and using questionnaires. An interview in this context can be defined as any meeting with the expert set up for the purpose of knowledge elicitation. It can include activities such as card sorting where domain elements written on small cards are used as a basis for a discussion which allows the knowledge engineer to determine how the expert conceptualises the domain (Cordingley, 1989). Once knowledge has been captured, it has to be put into a suitable form for computerisation. The knowledge engineer may proceed directly to system building and represent domain information using rules, frames, objects, etc.

Alternatively, before advancing to this stage, a model of the domain can be built. The hypotheses that underlie this approach are (Karbach *et al.*, 1990, p.173):

- It is useful to describe problem solving in a more abstract way than knowledge representation languages permit.
- Problem solving behaviour can be described in an application independent way.
- The knowledge acquisition process can be guided by model development.

There are various modelling approaches including Generic Tasks (Bylander and Chandrasekaran, 1987), Method to Task (Musen *et al.*, 1987), Role-Limiting-Method (Eshelman, 1988; Marcus, 1988) and KADS (Breuker *et al.*, 1987). The KADS methodology, for instance, allows the knowledge engineer to build a conceptual model of the problem solving process in a way completely independent of implementation details. The output from this stage can then be translated into rules, frames, objects or whatever form of knowledge representation is desired. A model provides an intermediate representation of a domain and has the advantage that a developer does not have to make a premature decision about design issues.

2.1.4 Explanation facilities in knowledge-based systems

The term *explanation* in knowledge-based systems refers to features that permit users to understand domain related issues. Explanation functions should not be confused with help messages which may be included in a knowledge-based system to assist users run the program. The philosopher Grice (1989) notes that the term *explanation* has two meanings. The first is roughly equivalent to "renders intelligible" whilst the second equates to "accounts for." These correspond to the two definitions that are given in the knowledge-based literature:

- *explanation*, to make something clear, to make the listener understand (Chandrasekaran *et al.*, 1988, p.220);

- *justification,* an explanation which aims not only to clarify but also to support a claim (Wick and Slagle, 1989, p.528).

Whilst the two terms explanation and justification are sometimes distinguished from each other, as in the definitions given above, expert systems researchers use them virtually interchangeably. This practice will be followed here.

Tracing the working of a knowledge-based system

Various researchers have contributed to the understanding of explanation in expert systems. The MYCIN system includes a facility which can answer questions about how and why certain conclusions are reached (Shortliffe, 1976). These explanations are directly linked to rules and goals. "How" relates to how the system reaches a conclusion and can be answered by showing the rules that have been utilised, whereas "Why" questions allow a user to find out what hypothesis is currently being investigated. Code is paraphrased to make explanations comprehensible to users (Buchanan and Shortliffe, 1984a). The problem arises, however, that the use of heuristics (short cuts) in the implementation of the original MYCIN system means that answers cannot be derived from first principles.

To rectify this, NEOMYCIN was subsequently built to represent the diagnostic task more explicitly. Its higher level approach to modelling knowledge means it can give strategic information about its goals (Clancey and Letsinger, 1981). A teaching system emerging from this, GUIDON, permits students to refer back to an earlier topic, change the topic and so on (Clancey, 1983b). To ensure effective learning on the part of the student, GUIDON can tailor what it says to a specific situation, presenting particular information that maximises assistance. During the course of his research, Clancey (1983a). was able to classify the knowledge needed for explanation into strategic, structural and support. Strategic knowledge refers to plans by which goals and hypotheses are ordered in problem solving whilst structural knowledge allows the rule base to be referenced in different ways. At the lowest level, support knowledge augments the information in the knowledge base. In another major development, XPLAIN, a system for generating knowledge-based systems, enables users to find out the reasoning behind the system, the justification for its actions. A history of the design rationale is kept so that knowledge obtained during system construction but not used for problem solving is made available for explanation purposes (Swartout, 1983).

To summarise, explanations were generated by early researchers in one or more of the following ways: paraphrasing the rules (Swartout, 1983; Buchanan and Shortliffe, 1984b), tracing back rules fired, searching forward for a goal (Buchanan and Shortliffe, 1984b), providing canned (pre-prepared) text (Swartout, 1983) and displaying the strategic goals of a system (Clancey, 1983a; Swartout, 1983).

Alternative frameworks for explanation

The main thrust of the earlier research was based on what has been termed *the line of reasoning*, that is tracing the workings of the expert system. Other researchers, however, argue that the line of reasoning is not necessarily the best way to proceed since a sensible line of explanation requires linguistic knowledge and support knowledge in the domain as well as the problem solving knowledge (Paris *et al.*, 1988). Paris's approach offers the following advantages:

- Texts are modelled after explanations given by humans, making the interaction more natural.
- Different lines of explanation may be employed to convince users that the expert systems conclusions are sound.
- Appropriate explanation strategies can be used for different user types.
- The difficulty of having to simultaneously solve a problem and produce an explanation is resolved. The knowledge base can use an efficient problem solving method and another more easily understood method for explanation.
- No change has to be made to the knowledge base to add explanation strategies.

An overall framework for explanation is provided by Chandrasekaran, Tanner and Josephson (1988; 1989). They note that explanation in expert systems is seen as involving issues of system self-understanding (by using a deep model of knowledge), user modelling and presentation. Users are not only interested in the range of problems the knowledge base can handle and the correctness of solutions but also the ability of the system to explain how it reached a decision.

They may wish to satisfy themselves that all relevant information has been taken into account and may want to check that the strategies adopted by the system are satisfactory. This can only be done by questioning the system. To ensure that the explanation meets the user's needs, Chandrasekaran *et al.* (1988) suggest that a model of the user is required. Moreover, the answers should be presented to the user in a meaningful way. If the user is to be able to communicate effectively with the expert system then the fixed dialogue sequences so frequently found in expert systems are insufficient; graphical displays and natural language generation must be explored.

Chandrasekaran *et al.* (1988) define three types of explanations that users might require :

- Type 1 - explains how the data is linked to local goals. This calls for examining appropriate fragments of the system's runtime behaviour. Problem specific data is matched to the knowledge base to show how certain conclusions have been drawn.

- Type 2 - explains knowledge base elements by calling on the deep knowledge from which it has been derived.

- Type 3 - explains the problem solver's control behaviour and problem solving strategy. Users are able to obtain answers to questions such as "Why was this factor not considered?"

Tanner and Keuneke (1991) extend this framework to include the following categories: step explanation, strategic explanation, task explanation and knowledge justification. Step explanation relates portions of the data to the knowledge for making particular choices whereas strategic explanation shows how certain lines of reasoning are linked to the problem solver's goals. Whilst these can be produced by accessing the program's knowledge, the other two kinds of explanations can be concocted, *post facto* justifications. Task explanation, therefore, can relate the system's actions and conclusions to the goals of the tasks that it performs whilst knowledge justification can show the connection between problem solving and domain knowledge. Tanner and Keuneke (1991) also mention that definition of terms may be required. Swartout and Smoliar (1989) discuss this at greater length noting that, often, terms in expert systems acquire their meaning based on how other parts of the system react to them. They advocate the distinct and separate representation of this kind of knowledge. A feature of this kind (the Clarification option) was incorporated within the APPLE system developed at Massey University (Kemp *et al.*, 1989).

Some researchers are concerned that explanations are a "one shot process", that is a single answer to a question is supposed to satisfy a user (Moore and Swartout, 1989; Cawsey, 1993). Follow up questions, in their view, should be allowed and misunderstood explanations clarified. Moore and Swartout (1990) have extended the Explainable Expert System framework to allow users to indicate what portions of an explanation should be elaborated. Cawsey, on the other hand, uses the EDGE dialogue planner to generate explanations in a teaching setting. She is interested in the development of extended dialogue between a computer program and a user. These researchers are all interested in, what can best be described as, the provision of interactive dialogue facilities.

Theory versus practice

It has been observed that many marketed expert system shells do not incorporate features proposed by researchers such as deep explanations, follow up explanations and user models (Dieng and Giboin, 1991). Since this observation was made, little seems to have changed. In the latest release of Nexpert Object (version 3), the only explanation facilities supported are "How", "Why", definitions and support knowledge (Neuron Data, 1993). Gregor (1991), too, comments that theoretical work in the area of explanation is quite distinct from commercial developments. Wognum and Mars (1991) indicate that, despite shortcomings in the provision of explanation facilities (such as lack of explicit user models), users still seemed satisfied with the limited facilities offered by the systems with which they were familiar.

It appears from a small scale survey carried out by Wognum and Mars that users want explanations of the following kind:

- *Overview*, how exactly a conclusion was reached;
- *Which*, a list of possible answers to a question;
- *Why Not*, why a conclusion failed;
- *Why*, background knowledge about why a question was asked.

Surprisingly, they find that facilities which trace the line of reasoning were never used. This is not the case with Gregor (1992) whose students did use

"How" and "Why" features although the most popular explanation function was a display worksheet that showed a worked example. Her results also suggest that participants with a prior knowledge of the subject use the explanation features more than the others.

Many problems arise when surveying the literature on explanation. Researchers have quite different perspectives on explanation. There is no standardisation on terminology and there appears to be a gap between what features users want and what researchers believe they should have. The onus seems to be on the system developer to determine what explanation functions users require. Explanations can be provided for the following purposes:

- to explain why a particular question was asked (Buchanan's Why, Chandrasekeran's Type 1 and Tanner and Keuneke's Step);
- to show how a conclusion was reached (Clancey's How, and Wognum and Mar's Overview);
- to explain why a conclusion was not reached (Chandrasekeran's Type 3 explanation and Wognum and Mar's Why Not);
- to illustrate the overall problem solving approach used by the program (Clancey's Strategic knowledge and Chandrasekeran's Type 3 explanation);
- to provide background domain knowledge (Clancey's Support and Wognum and Mar's Why);
- to justify an answer using a deep model of the domain (Swartout's Justification and Chandrasekeran's Type 2 explanation);
- to show how the system's actions and conclusions relate to the goals of the tasks (Tanner and Keuneke's Task);
- to define terms (Kemp, Stewart and Boorman's Clarification);
- to provide examples (Wognum's Which, Gregor's Display Sheet);
- to permit interactive dialogue (following Moore, Swartout and Cawsey).

The choice of possible features for the knowledge engineer is lengthy but research indicates that a few crucial functions should suffice.

2.2 Interaction issues in knowledge-based systems

Initially, the technical problems of developing expert systems were the centre of attention but in the late eighties attention was focused on meeting the needs of users. Cleal and Heaton (1988) address the issue of user-system interaction in some detail. They note (p.12) that this interaction should not be dominated by the system, "When a computer system is similar in complexity to a human expert it becomes necessary to allow for a level of communication similar to that between human expert and client." Users of expert systems need to request and receive explanations, estimate the certainty of facts that are being input or express complex ideas. Moreover, Cleal and Heaton (p.45) argue that since expert systems perform tasks normally associated with humans, the user is much more likely to be misled about "the obvious way to do this." Users have, in their opinion, to be protected against the ensuing cognitive dissonance. Given these circumstances, these authors conclude that the interface should allow users to interpret information easily.

Many other researchers note that the interface needs of the user are generally ignored (Kemp and Boorman, 1987; de Greef *et al.*, 1988; Breuker and de Greef, 1993). Hendler and Lewis (1988) refer to knowledge-based systems where the interface has been developed with no user consultation whatsoever, which sit on a shelf and are never used. Others have rejected expert systems because they only permit rigid system-oriented dialogues (Berry and Hart, 1990). The interface is clearly of crucial importance; it forms the bridge between the user and the knowledge base. In Payne and McArthur's words (1990, p.129), the interface "reflects both the state of the world that is being used in the reasoning and the process and rationale for that reasoning." In addition, a system may have to cater for a wide variety of user backgrounds.

2.2.1 Diverse backgrounds of users

Young (1989) makes the useful distinction between the build-time and run-time aspects of the expert system. This allows us to divide users into two groups: those involved with constructing the knowledge base and those for whom it is developed (see Figure 2.2). It is not the interaction requirements for build-time users (the knowledge engineer and the expert involved with knowledge acquisition), however necessary they are, that is of concern in this thesis. Instead, the emphasis is on identifying those factors that will enable those other users to interact successfully, in the widest sense, with the expert system. These users fall into various categories: consumers of expertise, experts and students (Hendler and Lewis, 1988).

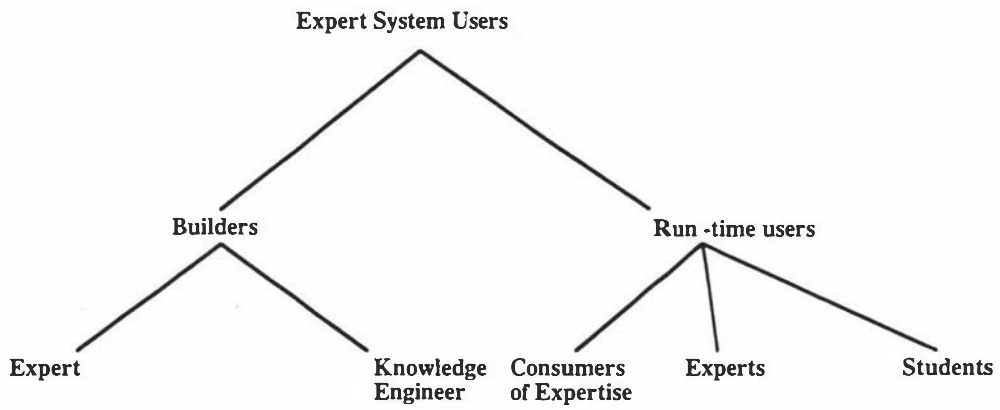


Figure 2.2 Knowledge-based system users (Kemp, 1990)

Consumers of expertise

Those described by Hendler and Lewis (1988) as consumers of expertise have a rather limited knowledge of a domain and will usually trust the results of an expert system. If, though, they are to take a more active role and not just obtain answers from the oracle, they must be able to ask questions, find out what the expert system is doing and experiment without fear of the consequences. Information about the task has to be provided in an easily assimilated way if users are to have confidence in the conclusions drawn or the actions recommended (Ellis, 1989).

Experts

Users who are experts in their own domain (medicine, law, physics, etc) may need convincing that the technology is sound and the knowledge stored in the knowledge base is correct. Furthermore, such individuals may wish to change the knowledge themselves and expect suitable explanations. Consequently, the user interface not only has to present the conclusions but also explanations of how these were reached. It may be crucial to the acceptance of the system that an explanation of the decision making process and a display of this reasoning is available (Hendler and Lewis, 1988).

Students

Intelligent teaching or tutoring systems can be built on expert systems (Wenger, 1987); the advantages include access to both information (from the knowledge base) and explanations ("How" and "Why" type questions can usually be answered). A well-known example is GUIDON which was built on top of MYCIN. Medical students could be exposed to a greater variety of cases via GUIDON than they were able to meet in their clinical studies. In practice, students had difficulty in absorbing the expertise since it was difficult to understand the reasoning strategy. Hence interaction proved difficult (Clancey, 1984).

Such a taxonomy has some value in assisting a knowledge engineer. It makes it clear that there are many ways that an expert system may be used. It also highlights the problems that can arise when developing a system for more than one group. In practice, though, the situation is more complicated than this. Within each group, levels of expertise may differ and users may have their own quite distinct domain needs. The problem is compounded when the computing background of users is also considered. Cleal and Heaton (1988) classify users of knowledge-based systems as follows: computer literate, computer illiterate, naive and average. This classification only considers users at a particular instance in time. Once experience is gained with a knowledge base, users may become more skilled at using a computer or learn more about the domain. A knowledge engineer may have to allow for all these possibilities. Given the complexity of the interaction that takes place in knowledge-based systems and the diverse background of the users, the interface may have to support many different ways of exchanging information. Researchers in the area of the human-computer interface have grappled for years with the problem of delivering user friendly systems with adequate functionality (Bright *et al.*, 1989). This literature is examined to see how these issues have been tackled.

2.2.2 Human-Computer communication

Dix *et al.* (1993, p.3) define interaction as "any communication between a user and a computer, be it direct or indirect." There are many difficulties involved, though, in defining the term *communication* as Dance (1970) discovered. He identifies 15 concepts that relate to communication. Two of these concepts are relevant to a discussion of interaction between people and machines: the concept of interchange and the associated idea of a process. Communication as an interchange refers to something being transferred from one thing or person to another. The definition is not precise and can describe what is transferred, the means by which it is transferred or the whole process. Dance conceives a process to be the transmission of information and skills by the use of symbols such as words, pictures, figures and graphs. These definitions anticipate those of the HCI community which can refer simultaneously to the channel of communication, what is transferred along that channel by symbols or the whole process (Kemp, 1992b).

A major problem for Booth (1989), is that whilst HCI researchers are concerned with the communication process, there is no theory of human-computer dialogue. Human beings speak to each other using natural language and, although they rarely use perfect grammar, are able to repair dialogue failures. In contrast to this, the communication that takes place via the computer is much less flexible. Nonetheless, Booth does put forward definitions of such important concepts as *dialogue*, *structure*, *content* and *style* which at least provide a framework within which the subject of communication can be explored. *Dialogue* is regarded as not only the exchange of symbols between two or more parties but the meanings that participants in the communicative process assign to these symbols. The *structure* of a dialogue refers to a description of its elements both within and between dialogue exchanges whilst the *content* is the semantics of the data exchanged in terms of the user's general knowledge of the meanings of words and specific knowledge of the nature and consequences of computer representation and actions. Semantics incorporates *pragmatics*, the study of meaning in context. Finally, the interaction *style* describes how the exchange takes place, whether command language, menu, form-fill, direct manipulation or natural language is used. What is ultimately communicated to the user may depend, therefore, not only on the literal meaning of the words but also on structure, content and style. There is no doubt that a heavy burden is placed on the developer whose system might be rejected by its intended users. As Booth (1989, p.58) observes, "A user's knowledge of the task, knowledge of natural language and knowledge of the machine can all potentially interfere with the interaction in subtle and complex ways."

There are those who have tried to model human-computer exchange on human-human relationships. The work of the philosopher Grice (1975) has been very influential; he puts forward the idea that human conversation takes place in a number of conversational moves each resulting in a new state. This approach has been developed by Reichman-Adar (1984) who uses an augmented state transition network to show how conversational moves may be mechanised. In another refinement of Grice's ideas, Allen and Perrault (1980) see speech acts as operators in a planning system. Agents attempt to recognise the plans of others and, in their answers, provide more information than explicitly requested. Their approach also allows for the generation of clarification sub-dialogues when the intent of the previous utterance is not clear. Whilst they are clearly concerned with the structure and context of the dialogue, this approach will not always be successful. The individual's plan might not be recognised. Without a generally accepted theory of human-computer dialogue, it is still not easy to determine how users communicate their intentions to the system and understand the system's responses.

Interaction styles

At its simplest, a dialogue takes place between people and computers which results in the exchange of information. In the context of human-computer dialogue, this information usually takes the form of data input, computer output, help and error messages. Typically, users interact with the computer using one or more of the following interaction styles: command language, menus, form-fill, WIMP (Windows, Icons, Menus, Pointers) or natural language (Shneiderman, 1987; Booth, 1989; Dix *et al.*, 1993). Each of these will be briefly described.

Command language

Users can communicate with the computer by typing in what are usually brief instructions chosen from a predefined command language. Feedback is immediate and this gives users of languages which offer a large number of options a great deal of power and flexibility. Such users, however, have to memorise many instructions and be able to select the most relevant to their needs. The burden on memory can be considerable and this interface style does not suit a novice user. Worse still, as Dix *et al.* (1993) observe, commands may not only be obscure and inconsistent within systems but may vary across systems. These problems can be alleviated by choosing meaningful commands and abbreviations.

Menus

The options available to the user are displayed on the screen in terms they will understand and are selected using key presses or a pointing device. This has the advantage of structuring the task for inexperienced users of computers but may constrain the actions of expert frequent users. Shneiderman (1987) comments that this can be overcome with careful design. Dix *et al.* (1993) also note a problem that arises when menus are hierarchically ordered. They believe that menus have to be logically grouped for users to be able to traverse the structure to find the option they desire.

Form-fill

Form-fill allows users primarily to enter data in fields on a display that resembles a form; it may also be used occasionally for retrieval purposes (Dix *et al.*, 1993). The display may be, but is not necessarily, based on a form with which the user is familiar. This has the advantage of ease of use but is quite limited in the options available. Shneiderman (1987) provides a long list of design guidelines for this type of interface. Amongst the most important are: comprehensible instructions, visually appealing layout, error correction facilities for characters and fields, error messages for unacceptable values and explanatory messages for fields.

The WIMP interface

Currently the WIMP interface is extensively used in the personal computer area. Windows are areas of screen that contain a representation of a software application or a document file. A window may be replaced by its icon, a picture that represents a closed window. Double clicking on the icon with a pointer is usually sufficient to open it. The pointer is often a mouse but may be a joystick or trackball. The selection process is generally referred to as "point and click." Pull down or pop up menus can be made available to the user. Associated with the WIMP paradigm are the inclusion of buttons and dialogue boxes (Dix *et al.*, 1993). WIMP interfaces are seen as a type of direct manipulation interface as described by Shneiderman (1987). The overwhelming advantage of this interface style is its ease of use.

Natural language

Many believe that people should be able to communicate with a computer using their own language. Unfortunately, since language is subtle and context very important, it is difficult for a machine to understand (Shneiderman, 1992). Such are the problems involved that one author criticises designers of expert systems for attempting to embed natural language, "Expert systems already tax the user with complexity, lack of visibility of the underlying processes and confusion about what functions the system can and cannot handle" (Shneiderman, 1992, p.169). Restricted subsets of a language, though, can be used for applications such as database querying and searching. The user may type or speak the instructions.

It can be seen that these categories are not discrete. How does a subset of natural language, for instance, differ from a command language? There is also scope for disagreement. It is interesting to note that Shneiderman, for instance, sees spreadsheets as an example of a direct manipulation interface whilst Dix *et al.* see it as a form-fill approach. Classifying interaction styles is currently a somewhat subjective exercise.

To help people come to grips with an interface style, metaphors have often been suggested as a fruitful way to teach users about a system (Carroll and Mack, 1985). The desk top metaphor underpins the WIMP interface style whilst accounting packages like Excel and LOTUS 1-2-3 are based on the idea of the spreadsheet. Although metaphors may be of use initially they often break down later on. It is confusing to some users of the Macintosh WIMP interface to drag a disk icon to the trash bin when they wish to eject it. This seems an unusual use of a receptacle normally associated with the disposal of rubbish.

Even though there is no generally accepted theory of dialogue, some authors have produced their own guidelines for dialogue design. The following eight golden rules are suggested by Shneiderman (1987):

- strive for consistency;
- enable frequent users to use shortcuts;
- offer informative feedback;
- design dialogues to yield closure;
- offer simple error handling;
- permit easy reversal of actions;
- support internal locus of control;
- reduce short term memory load.

Shneiderman expects that these underlying principles will be refined, interpreted and extended for each environment. Other guidelines are available such as those proposed by Gaines and Shaw (1983) for dialogue engineering. All of these are criticised by Booth (1989) as pertaining more to interface design than dialogue design, that is they do little to enhance our understanding of communication. They do highlight, however, a major problem with the interface, that is the *locus of control*. Who is actually in charge of the communication process, the user or the computer? Often the computer seems to be dominant with the user playing a subservient role. This happens with form-fill interfaces. At the other end of the spectrum, command line interfaces enable the user to be in complete control. Only mixed initiative systems allow each party take control as required (Booth, 1989).

Decisions about presentation styles and the locus of control depend on the nature of the application and the background of the users. The tasks and subtasks to be carried out by the users and the program have to be determined. Shneiderman (1987) suggests that the designer should develop a profile of the intended users and then carry out a detailed task analysis.

Task analysis

In their guide to task analysis, Kirwan and Ainsworth (1992, p.vii) define it as "the name given to any process that identifies and examines the tasks that must be performed by users when they interact with systems." They describe 25 different techniques that help to build up a picture of the system from the human perspective. The guide was written to help designers, ergonomists, safety assessors, project managers, etc select the appropriate technique from the large array at their disposal. Historically, as Johnson (1992) observes, task analysis was strongly influenced by training needs. With its emphasis on analysing tasks in terms of human behaviour, it was subsequently able to make a considerable contribution to HCI. Within HCI, it has been suggested that task analysis be used for requirements specification, to assist with the production of training materials and documentation, and to aid the design and evaluation of interfaces (Shneiderman, 1987; Carey *et al.*, 1989; Johnson, 1989; Kirwan and Ainsworth, 1992).

The two principal approaches to task analysis used in HCI are task decomposition and knowledge-based techniques (Dix *et al.*, 1993). Task decomposition methods can describe system goals and the subtasks needed to achieve them. These can be recorded in matrices, tables or tree structured form. The sequencing, selection and iteration of subtasks can be accommodated. Shneiderman (1987) suggests the use of task analysis to determine the functions of a new system. The outcome, ideally, will be a table of user communities and their associated tasks together with the expected frequency of use. During the design stage, care should be taken to ensure that frequently performed functions are made easy to carry out. He mentions problems, though, with identifying who the users will be and what tasks they may wish to accomplish. At a lower level, Shneiderman is also concerned that, using this technique, it may be difficult to produce a set of atomic actions. For Kirwan and Ainsworth (1992), however, task analysis should be a collaborative process between the task analyst and those involved in operating a system. This overcomes many of the problems identified by Shneiderman but is not always realistic in projects that completely transform the way a job is carried out. The input of the current operators is then of little value. Whatever approach is taken it has to be recognised that the output of task decomposition procedures depends on the skill of the analyst and there is no single correct answer (Dix *et al.*, 1993).

Knowledge-based task analysis originated in a British project to establish a national syllabus for training in information technology. Task Analysis for Knowledge Description (TAKD) was developed to describe knowledge requirements independently of the constraints of the original tasks analysed (Johnson *et al.*, 1984). The resulting taxonomies of generic objects and actions can be used as the basis for describing existing or novel tasks. It was recognised that such an approach could make a contribution to designing and evaluating interfaces (Diaper, 1989; Johnson, 1989). Johnson and other researchers at Queen Mary College, London, went on to develop Knowledge Analysis of Tasks (KAT). This is based on the premise that the knowledge that people have of tasks should be taken into account in the development of interactive software systems. KAT entails collecting data from those involved with a task; identifying people's knowledge in terms of actions, objects, procedures and goals and building this into a model after an analysis which determines the typical and central procedures. After experience with various applications, Johnson believes that KAT can contribute to the design of functionality, presentation styles and dialogue structures.

Benyon (1992) criticises the knowledge-based approach to task analysis on the grounds, *inter alia*, that it is expensive in time and effort and does not allow the analyst to fundamentally change user responsibilities. On the other hand, Johnson (1992, p.175) specifically claims that KAT can be used to model new or changed tasks. This may be the case but, with inexperienced analysts, it is possible that only the existing situation will be computerised.

Bearing in mind the criticisms noted, the advantages of task analysis can be summarised as follows:

- It allows you to identify your community of users (Shneiderman, 1987).
- It recognises the importance of the user (Dix *et al.*, 1993).
- It permits the knowledge that users have about tasks to inform the design process (Johnson, 1989).
- A hierarchical representation of tasks is effective if they can be strictly decomposed (Benyon, 1992).

Nonetheless, there are other ways of specifying the functionality and interface requirements. Systems analysts typically use interviews and questionnaires to determine user requirements. Models are often built of users to feed into the design process. Indeed Dix *et al.* (1993) mention that some aspects of task analysis look like goal-oriented cognitive models.

User models

A wide range of user models is described in the HCI literature (Whitefield, 1987; Young *et al.*, 1989; Nielsen, 1990; Johnson, 1992). A full account is beyond the scope of this thesis. What follows is a discussion of the term *user model* and an account of those ones that may be developed for interface purposes. Kelly and Colgan (1992) believe that there have been essentially three different definitions of the term. There is the designer's model of the user (Norman, 1986), the computer program's model of the user, that is an embedded user model (Clowes *et al.*, 1985), and the user's mental model of the system. Sutcliffe (1988) also points out that there is often ambiguity about who constructs a user model and what is being modelled. He attempts to clarify the situation by identifying the main types of user models that have been found in the human-computer interface literature: user views, cognitive models constructed by psychologists, models of user knowledge, models of user characteristics and user task models. Together these models should allow a designer to construct a suitable interface.

User views

Before discussing user views in any detail, a useful starting point is to consider Norman's (1988) influential model of user interaction. This is a seven stage process:

- 1. Forming the goal
- 2. Forming the intention
- 3. Specifying the action
- 4. Executing the action
- 5. Perceiving the system state
- 6. Interpreting the system state
- 7. Evaluating the outcome

This is a very high level description of interaction with a computer but highlights what Norman refers to as the gulf of execution and the gulf of evaluation. The former describes the gap between what a user wants to do and what are permitted operations. The latter on the other hand is the difference between what appears on the screen and what the user expected. The ensuing cognitive dissonance is often the result of an inadequate mental model or user view.

Norman (1986) suggests that human beings form internal mental models of themselves and of the things and people with whom they interact. Such models can provide predictive and explanatory powers for understanding the interaction. Whilst they evolve naturally through interaction with the particular system under consideration they are also highly affected by the person's prior knowledge and understanding. Although the models are not complete nor accurate they still guide much human behaviour. Consequently, the primary task of the designer is to build an appropriate system image. Everything that the user interacts with to form that image (such as the mouse, keyboard and the documentation, including instruction manuals, help facilities and error messages) must assist with the generation of the image. A great deal must be known about the users' views and characteristics if a designer is to successfully exploit Norman's ideas and enable a user to build an adequate mental model or user view of the system.

Moran (1981) sees Command Language Grammar (CLG) as a way of representing the mental model the user has of a system. CLG is a framework for describing interface aspects of computer systems; the conceptual model is specified and a command language created for user-system communication. A system is partitioned into its conceptual component (task and semantic levels), communication component (syntactic and interaction levels) and physical component (spatial and physical levels). The first component relates to what the user wants to achieve, the second to the dialogue structure and the third to the device used to achieve goals. The levels are developed in the order listed although Moran himself did not go beyond the first four levels.

Users come to the system with a set of tasks that they wish to accomplish. Presumably, some specification of user requirements has taken place. The top level of this model describes the system as seen from the perspective of the users and identifies the tasks in a way amenable to interactive systems. The subtasks needed to accomplish a goal are depicted in a hierarchical fashion.

Moran notes that there are, in practice, two distinct conceptual models of a system: the model the user has and the model the designer thinks the user has. These models are assumed to be identical. Overall, it is difficult to accept Moran's claim that he introduces a psychological dimension to interface design since little information seems to be available about users (knowledge, characteristics, etc). For this reason, Johnson (1992) believes that CLG should be considered as a designer's model of the interaction. Nonetheless, like task analysis, it relates user goals to subtasks. It also emphasises that communication issues should be central when developing an application.

Cognitive models

The GOMS (Goals, Operations Methods and Selection) model attempts to predict such factors as performance time and short-term memory requirements for error-free expert users (Card *et al.*, 1983). User goals are decomposed into the methods that can accomplish them. GOMS tries to predict what method the user will select. Methods are decomposed into the basic operations, the actions that the user must perform to affect the environment. This model is very specifically targeted and, since the users are expert, it sidesteps the problem of obtaining information about their background (Dix *et al.*, 1993).

An extension of GOMS, known as CCT (Cognitive Complexity Theory), was developed by Kieras and Poulson (1985) to predict the difficulties a person is expected to encounter when learning and using an interactive system. Both the task and the device (software) are depicted. Production rules describe the knowledge needed to achieve the users' goals, whilst the software is represented by a state transition network. Both of these, as Dix *et al.* (1993), note can be represented as hierarchies. Johnson (1992) is critical of CCT because it does not define what a task is and fails to differentiate between the many alternative ways that tasks can be performed. Dix *et al.* (1993) also mention the problems with CCT but regard it as a tool giving a rough measure of learnability and difficulty as well as a detailed description of user behaviour.

Models of user knowledge

Models of user knowledge are often embedded in intelligent teaching systems. Explanations and feedback can be given to students if the system can assess the current state of user knowledge. The problem of building an expert system for users with quite different backgrounds in the domain has already been discussed.

Models of user characteristics

Users can be defined according to their experience in dealing with computers. The four main categories in the HCI literature are naive, novice, skilled and expert users. To classify these, some metrics are needed. Sutcliffe (1988) proposes that frequency of use, computer familiarity and willingness to learn should all be taken into account.

User task models

These describe the users' view of a task, the operations to be carried out and the ordering of these (Sutcliffe, 1988). Obviously, these models can be developed during the task analysis process.

Overall, there is much that is valuable in the modelling literature, particularly the emphasis on helping users build up an appropriate mental model of a system. A good starting point is to identify the goals of users in some way (GOMS, CCT, task analysis or analysis of user requirements) and map them to the procedures that allow users to achieve them. Then, any other models that are required for the application, such as the model of user characteristics, can be developed.

Usability

The need for developers of interactive systems to select an appropriate interaction style and use relevant models is a reflection of a concern for usability. Usability is widely discussed in the HCI literature (Shackel, 1984; Gould and Lewis, 1985; Shackel, 1986; Thimbleby, 1990; Carey, 1991; Guillemette, 1991; Allison *et al.*, 1992; Johnson, 1992; Dix *et al.*, 1993; Preece, 1993). Dix *et al.* provide an up-to-date and comprehensive review of usability. They distinguish two important issues:

- how to build systems that are usable;
- how to demonstrate or measure system usability.

They also identify the three principles of *learnability*, *flexibility* and *robustness* that support the concept of usability. Learnability refers to the ease with which new users can interact, flexibility pertains to the many ways the user and system exchange information whilst robustness relates to the support given users to help them achieve their goals. These can all be further subdivided into more specific guidelines. Some of these principles are to do with software performance (responsiveness and multi-threading) whilst others relate more to helping a user conceptualise what is happening (consistency and predictability). Yet others are concerned with allowing users to control their environment (dialogue initiative, customizability and recoverability). Finally, task conformance, that is the ability of a system to support the tasks the user wishes to carry out, is seen as very important. Whilst guidelines are very useful, they are, to some extent, application dependent. There is not always a need for a multi-threaded system, for instance. Nor can following the guidelines be a guarantee of success. It may prove impossible, in practice, to meet the needs of all users.

To ensure that interactive systems are usable, some researchers believe that interface evaluation should take place throughout the design process (Howard and Murray, 1987; Johnson, 1992; Hix and Hartson, 1993). This is referred to as *formative* evaluation, in contrast to *summative* evaluation which involves assessing the overall performance of an application. Formative evaluation of both the program specification and prototype can take place during the development process and provide relevant feedback. A specification can be tested using methods such as cognitive walk-through (Lewis *et al.*, 1990) and heuristic evaluation (Nielsen and Molich, 1990). The former method attempts to simulate the human-computer interaction whilst the latter is an assessment based on usability principles such as the following (Desurvire *et al.*, 1992, p.93):

- provide simple and natural language;
- speak user's language;
- minimise memory load;
- be consistent;
- provide feedback;
- provide clearly marked exits;
- provide good error messages;
- prevent errors;
- provide shortcuts;
- do not overload the user with documentation.

In a comparison of these two approaches, Desurvire *et al.* (1992) conclude that heuristic evaluation considers more dimensions of the interface than cognitive walkthrough and is better at predicting specific problems that arise when testing in the laboratory. Nielsen (1990) notes that experts in both the domain and human factors area are best at finding an interface's usability problems during a heuristic evaluation. This is an interesting finding given that there will not be too many people who are expert in both human factors and a particular domain problem.

Carey (1991) suggests that measurable objectives for usability should be established. Criteria for ease of use can be established (such as the time it will take 80% of the users to carry out particular tasks). These measurements relate to user performance but if specified during the analysis phase of a project can be used to guide the formative evaluation of an interface.

Whatever efforts are made during system development to meet user needs, it is still likely that users running a system will require assistance from time to time. Aid can be provided by including appropriate help or error messages in the program. Help systems provide users who do not know what to do next with instructions or prompts. Rubin (1988) briefly discusses the several types of help systems available ranging from on-line manuals to context sensitive assistance. *Dix et al.* (1993) believe that a help system should incorporate a number of parts so that users can obtain different types of help for different purposes. Even so, users continue to make mistakes. Ideally, recovery can be achieved using an undo command (Rubin, 1988). This is not always possible and Shneiderman (1987) believes that error messages which tell the user how to rectify a situation are more useful than those that just report a problem.

Finally, usability is very important but usefulness should not be underestimated. Davis (1993) investigates how ease of use and usefulness influence the acceptance and actual usage of software (Figure 2.3), and finds that perceived usefulness is vital. He observes (p.484) that

> *Users may be willing to tolerate a difficult interface in order to access functionality that helps them on their job, while no amount of ease of use can compensate for a system that doesn't do a useful task.*

This, he concludes, underscores the importance of incorporating the appropriate functions in new systems. What Davis highlights is that usefulness is a critical issue. Without this, any other features that make the software easy to use are essentially redundant.

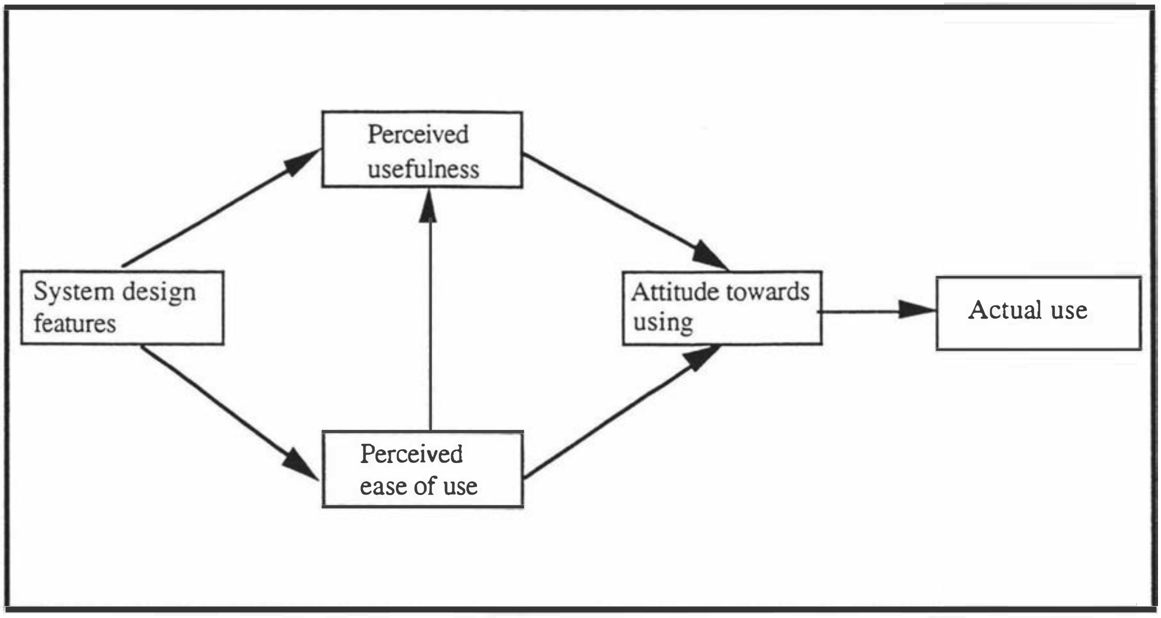


Figure 2.3 Information technology acceptance model (Davis, 1993)

2.2.3 Interaction issues in knowledge-based systems

The foregoing analysis of the HCI literature clearly demonstrates the central importance of the user. Technological concerns should evidently not be permitted to drive the construction of interactive knowledge-based systems. In line with typical HCI concerns, Stelzner and Williams (1988) propound the following basic principles of interface design for knowledge-based systems:

- represent the domain in the user's natural idiom;
- provide immediate feedback on changes in the system state;
- allow the user to try different alternatives;
- support the user at different levels of abstraction;
- provide multiple interfaces if needed.

They believe that the user's natural idiom is often graphical. Such interfaces easily permit users to act directly upon them and to see immediately the outcome of any action. However the domain is represented, users should be able to experiment and even guess at answers. Stelzner and Williams (1988, p.293) claim that "the success of the system is directly related to the users' ability to try out various decisions." Users may need assistance with understanding what is happening in the knowledge base. They can be helped to come to grips with the complexity of the knowledge base by the provision of browsing facilities and multiple interfaces.

Metaphors in knowledge-based systems

Many other issues that have engaged the HCI community have also been tackled within the knowledge-based literature. The role of metaphor has received a great deal of attention. There needs to be both a suitable model of user/system interaction and the selection of an appropriate interface metaphor if co-operation between the user and the system is to be achieved. Initially, the dominant mode of interaction was that of expert and novice (Hayes-Roth *et al.*, 1983). Many authors have called attention to the unnatural and tedious question and answer sessions associated with this (Kemp and Boorman, 1987; Lowgren, 1989). In effect, the user is relegated to merely feeding in data and is not likely to learn a great deal about the domain. Another, more fruitful, interaction metaphor is suggested by Rector *et al.* (1985), with the novice replaced by a helpful assistant. Meanwhile, intelligent teaching systems had already focused on the teacher-student relationship. In GUIDON, the locus of control is not with the system alone since students can refer back to an earlier topic and change the topic if necessary (Clancey, 1987).

Different metaphors have been used to guide interface development. First in time was the implementation of conversational systems with their question and answer sequences. Recently, the *model world* metaphor has been emphasised with Direct Manipulation Interfaces which allow the users to concentrate on the substance of the problem since they offer symbolic representation of the task domain (Potter, 1988). Such interfaces increase the users' understanding since they offer them an exploratory environment where understanding of relationships is increased and exploration of alternatives facilitated (Baroff *et al.*, 1988). Advances in the area of explanation depend upon the availability of both natural language and direct manipulation facilities for the user, so giving the advantages of both metaphors.

User modelling in knowledge-based systems

User modelling in knowledge-based systems is undertaken not only to help users form an appropriate mental model of the system but to assist with the generation of suitable explanations (McKeown *et al.*, 1985; Busche, 1989; Ellis, 1989; Kass and Finin, 1991). It might also be required for problem solving purposes (Busche, 1989). Models, as Rich (1983) points out, can be of individuals or groups (usually referred to as stereotypes), be long term or short term and be defined by the user or by the system designer.

The framework just described, with the exception of a model for problem solving purposes, is little different from that of Sutcliffe (1988). First, let us consider user views. The importance of assisting users to form an appropriate view of a knowledge-based system cannot be overestimated. Lehner and Kralj suggest that the quality of user/system interaction in knowledge-based systems depends upon the user's cognitive model of the reasoning process within the system (Lehner and Kralj, 1988). The user's cognitive model (view of how the expert system processes information) has to be derived from the interface. It is assumed by these authors that users initially create an anchor model based on their own framework of reference, which is adjusted as a result of their experience with a program. Cues should be provided by the output displays which can help users to determine how the system works. Inadequate output with little or no explanation leaves the anchor model (with all its misconceptions) unaltered whilst spurious cues lead only to further errors.

The need for the interface to help the user model the processes of the expert system is also recognised by Hendler and Lewis (1988). These authors describe how early medical systems that provided correct answers still proved unacceptable to both expert and non-expert users since they could not predict what questions the system would ask nor follow the line of reasoning involved. Hendler and Lewis conclude that the ability to explain must be incorporated within the application. They imply that unless the interface delivers explanation functions, users will not be able to form an adequate mental model of the system.

Busche has written at great length about the role of user modelling in knowledge-based systems (Busche, 1989). She believes that user modelling helps the developer match the presentation style and level of support (such as help facilities) to the user's background. An interaction model can contain the interaction characteristics and requirements of a user whilst a personality model describes the user's level of knowledge and experience with computers as well as other personal details. The two together are similar to Sutcliffe's (1988) model of user characteristics. It seems odd to separate the information about computing experience from interaction requirements and it seems preferable to combine this information in a model of user characteristics. Busche also suggests that an embedded model can be held in the computer so that, as users became more experienced using an interface, they can be offered more advanced features. There are two problems with this: firstly, the possible difficulties for users of having to adapt to another interface and, secondly, the separation of the interface from the application. In knowledge-based systems, it may be important to match up the presentation style with the domain needs, graphically representing a device for diagnostic purposes, for instance.

User modelling is also important for controlling the semantics of interaction. This relates, according to Busche, to the contents of system output. If users are to understand what is happening, terminology has to be comprehensible. Users' knowledge may have to be modelled for solutions and explanations to be comprehensible. The information about a user's level of domain knowledge is kept by Busche in the personality model whilst this is pulled out by Sutcliffe (1988) into a model of user's knowledge. Given the importance of tailoring explanations to the user's level of expertise (McKeown *et al.*, 1985; op de Hipt, 1988; Paris, 1991), it seems preferable to draw up a separate model of users' knowledge.

Busche also discusses developing a task profile. This is a description of the user's tasks relating to the environment in which a system is used. It includes a description of the user's task knowledge. This is very similar to Sutcliffe's user task model and is needed to establish user requirements. Busche also introduces a model of the user that is needed for problem solving purposes. If a knowledge-based system is providing advice about superannuation, for example, then it needs all the relevant details for each individual. Whilst such information may be elicited during the knowledge acquisition phase, it does seem useful to bring it all together in one model if only to ensure that it is complete.

A knowledge engineer will have to decide during the analysis phase of a system what user models (knowledge, characteristics, task and problem-solving) are necessary for the particular application. The facets of the user that need to be included within each model also have to be determined. Busche provides a comprehensive list of factors to consider. This can be consulted as necessary by the knowledge engineer. Despite this literature on the topic of user modelling, Regoczei (1992) comments that practically no knowledge engineers think of modelling the user.

User modelling is seen as a solution to the problem that arises when potential users of a knowledge base may have quite different domain and computing experiences. de Greef *et al.* (1988) recognise that an expert system may have to work in more than one mode, that is support different types of users. They provide a framework, referred to as *modality*, which allows the knowledge engineer to specify the co-operation and communication required between the user and the computer. All possible distributions of labour can be identified so that the specific distribution(s) to be supported (for instance student and expert) can be determined. Mixed initiative dialogue, where either party can take control of the exchange, can be modelled.

Task analysis in knowledge-based systems

The framework for task analysis advocated by de Greef *et al.* (1988) is based on Moran's (1983) methodology for designing command languages. Moran's six level structure is condensed into three stages: task level, semantic level and the communication level. Within this framework, both problem solving and the task distribution are elaborated (see Figure 2.4). At the top level, *the basis task*, a global description of the problem solving process, is described. All possible task distributions are also identified. The diagram of the basis task can be annotated to show which activities are assigned to the system and which to the user. During the second stage, the semantic level, the functional capabilities of the future system are defined. Control of the execution of the task distribution is dealt with at this point. Finally, the communication level deals with when and how to interact. The analysis of the domain and co-operation proceeds in a top-down, hierarchical fashion with the occasional zigzag between boxes at the same level. This occurs, for instance, when a task distribution has implications for modelling the basis task.

level \ domain	problem domain	cooperation
user/expert task	1. task model decomposition	2. task distribution
semantic level (functions)	3. problem solver	4. cooperation model
communication level	5. transparency	6. interaction

Figure 2.4 The modality framework (de Greef *et al.*, 1988)

The distribution of tasks has to be modelled with various alternatives possible. At its simplest, tasks can be allocated either to the user or the system. It is also possible for both parties to be involved, with the user performing the task and the system checking or *vice versa.* In addition, the user may have to carry out a job under instruction from the system. The notation used in this methodology shows the ownership and flow of both *ingredients* and *initiative* where the term *ingredients* refers to elements such as data, skill and knowledge whilst *initiative* is concerned with who transfers ingredients. In a sense, this is timing information since the owner of the initiative can signal the moment of transfer or show a need for it. Communication can then take place in many different ways. It is suggested that natural language should be the major mode of entering information but feedback may call not only on text but also pictures, graphics and even animation.

Usability in knowledge-based systems

Mitta (1989) points out that there are problems in establishing the usability of a knowledge base. It appears that user opinions and performance are not well correlated. Users may rate an interface highly but do not find the system itself useful. From professionals in the knowledge-based area, Mitta obtained six important measures of usability:

- user confidence in the recommendation made by the expert system;
- correctness of recommendation;
- inability of expert system to provide recommendation;
- user's perception of difficulty;
- rate of help requests;
- number of responses required of users.

The first three factors relate to the perceived value of the expert system, specifically to its problem solving capabilities. Mitta, like Davis (1993), realises the importance of usefulness. The other three measures are concerned with ease of use (user's perception of difficulty, rate of help requests and number of responses required of users). Of these, the number of responses required, defined as the mean number of responses users must provide to obtain a conclusion, seems a little unusual. Presumably, in some applications the decision path will be quite lengthy. This could be a function of the problem, not the system. This kind of interface evaluation is clearly summative not formative. There is little in the knowledge-based system literature about evaluating the specification of an interface. A prototype, though, may be built and user opinions solicited (Buchanan *et al.*, 1983; Waterman, 1986; Eason and Harker, 1987; Preece, 1990; Turban, 1992). There is no reason, however, why methods employed in the HCI area such as cognitive walkthrough and heuristic evaluation should not be used.

2.3 Conclusion

There seems little doubt that the construction of an interface to assist users of any interactive system to achieve their goals is a complex process. Decisions have to be made about what tasks the user wants to carry out, how to divide responsibilities between the user and the system, and the selection of appropriate interaction styles. When the application in question is a knowledge-based system, this process seems even more fraught with pitfalls.

In addition to the usual difficulties of using a computer to perform a job, users also have to comprehend the reasoning processes of the knowledge-based system. Extensive explanations may have to be made available to assist them. The HCI and knowledge-based literature offers much sound advice to developers. Carrying out some form of task analysis, modelling the user, and designing an interface in the light of pre-specified principles of usability are means of alleviating these difficulties. One perspective that is noticeable by its omission, however, in the literature is that of users themselves. A completely rounded picture of the interface in knowledge-based systems cannot be provided until this deficiency is remedied and users themselves consulted.

Chapter 3

A Study of Knowledge-Based Systems Usability

This chapter presents the results of a survey on interface issues in knowledge-based systems. The survey investigates students' usage of a horticultural expert system called APPLE. A brief overview of APPLE is followed by an analysis of the user population. The principal findings regarding interaction styles, presentation of information and explanation facilities are discussed. The question of whether there is a relationship between users' domain knowledge and their interface preferences is also explored.

3.1 Introduction

For the knowledge-based or expert system to come to sensible conclusions, users have to interact with the system (using, for instance, direct manipulation, natural language, menus, etc) and understand, to some extent, the underlying domain. The user will generally be expected to issue and follow instructions, answer questions, request and digest explanations. Difficulties may arise at many stages. If the terminology in the questions is incomprehensible to the user then even the initial data required for the knowledge base cannot be collected. Users may also have to qualify their answers since they are not necessarily sure of their facts. Similarly, the result of a consultation is not always clear cut and the degree of certainty, for example, of a diagnosis, must be given to the user.

The literature indicates that the interface is an important issue when developing a knowledge-based system (Kemp and Boorman, 1987; de Greef *et al.*, 1988; Hendler and Lewis, 1988) but there is little empirical evidence from users themselves. An exploratory survey was undertaken to find out how a group of users wanted to communicate with a knowledge-based system: their preferred mode(s) of interaction, their explanation requirements and the significance, if any, of the user's level of domain expertise (Kemp and Kemp, 1991a; Kemp, 1992a; Kemp, 1992b). A group of students with differing levels of domain knowledge was given the opportunity to use knowledge-based software, the APPLE expert system (Kemp and Boorman, 1987; Kemp *et al.*, 1989) which diagnoses diseases in apple crops. A questionnaire was developed to measure their preferences. This is a cost effective and straightforward way of gathering information about user perceptions of the interface (Karat, 1988).

3.2 The APPLE system

APPLE is an interactive system, gathering information from the users about a particular plant health problem during a consultation. From the information obtained, a list of possible causes is set up and the likelihood of each being present is estimated. Since one of the main aims of the system is to provide a user-friendly interface, a WIMP interface has been provided. With regard to information input, this interface relies heavily on the selection of one alternative from a list of possible answers by a pointer (Figure 3.1). For output, both graphical and textual presentation are employed.

The user comes to the computer with a plant health problem that may be real or proposed for teaching purposes. At the start of a session users can, if they wish, obtain information about how the system works. Experienced operators can by-pass this stage. The system then asks questions in order to eliminate as many causes as possible and to determine the most likely ones. The initial questions request specific details about location, types of apples affected, soil and leaf test results whilst the more general questions that follow are concerned with leaf and fruit systems. The possible alternative answers to the general questions are listed on the screen and can be selected by pointing and clicking on the appropriate one.

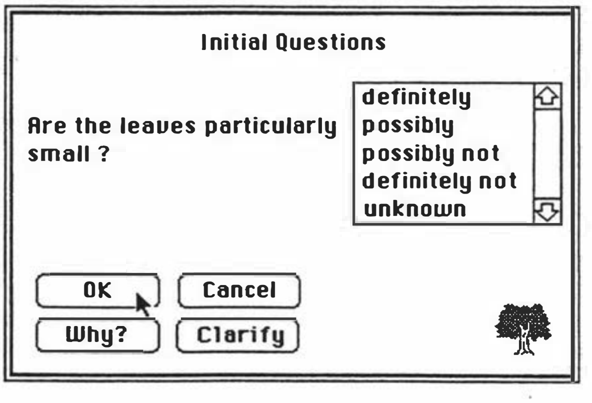


Figure 3.1 Standard question and answer format (Kemp *et al.*, 1989)

Sometimes, however, the user may wish to shade the answer. For example, someone may not be sure whether the leaves have chlorosis. In this case, 'possibly' or 'possibly not' can be selected from the alternatives and the system presents a follow up screen asking users to state how sure of their reply they are (Figure 3.2).

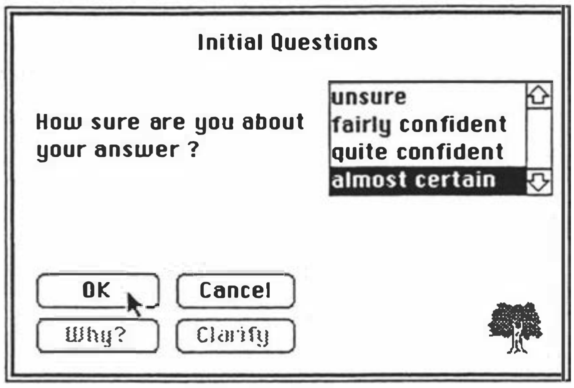


Figure 3.2 Graded scale for uncertain answers (Kemp *et al.*, 1989)

At the end of the first part of the consultation, a set of hypotheses is presented to the user. Each of these hypotheses is then investigated in more detail.

During the consultation the user may request further information about what is being asked. Currently, the system allows for two kinds of elaboration, "Clarify" and "Why". Since users differ in their understanding of plant science vocabulary, terminology is kept to a minimum and where used can be explained by way of the "Clarify" option (Figure 3.3).

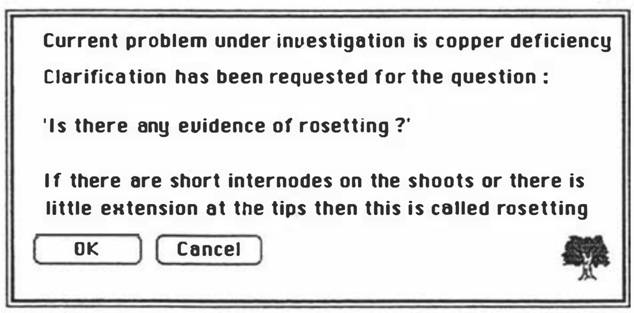


Figure 3.3 The Clarify option (Kemp *et al.*, 1989)

It is often useful to know the reason behind a question to understand the point of it. The "Why" facility provides for this. Explanations of the "Why" form are usually provided in expert systems. Users can be given some explanation of why a question is being asked, that is, what the questioner is leading up to. The question may pertain to a characteristic symptom, or, in other cases to a piece of evidence that would tend to reduce the likelihood of a condition (Figure 3.4).

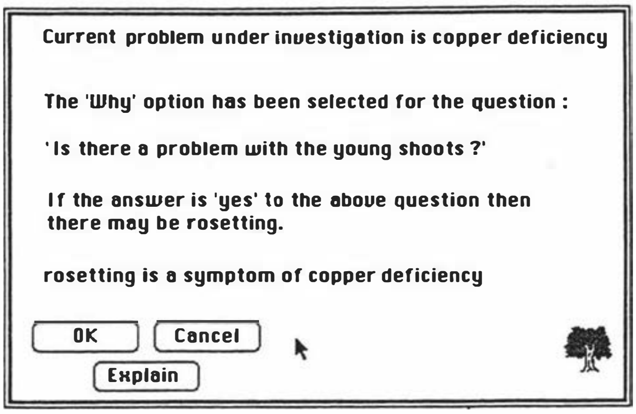


Figure 3.4 The Why option (Kemp *et al.*, 1989)

At the end of the session the system uses the accumulated evidence to produce likelihoods ranging through from 'almost certain' to 'extremely unlikely' for each of the causes considered. This information may be accessed in graphical form (Figure 3.5).

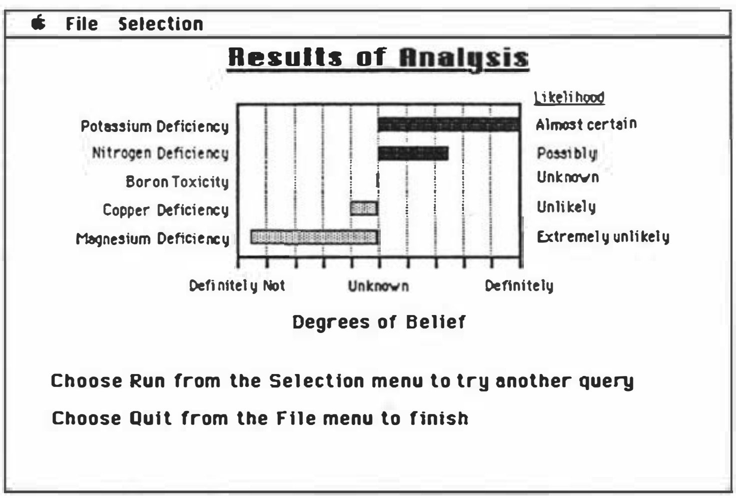


Figure 3.5 Presentation of results in graphical form (Kemp *et al.*, 1989)

3.3 The survey

The population for the study comprised students enrolled on a course at Massey University. They all completed a questionnaire consisting of 19 items. The questions were principally closed in nature and responses to these were measured on a 5 point Likert-type scale. This was an exploratory survey whose results could be quantified and analysed. A survey which would have involved the collection of qualitative data was not undertaken. Whilst data obtained in this way may provide in depth information on the areas of interest, it may not be possible to code it (Sekaran, 1984). Furthermore, drawing conclusions may be fraught with pitfalls. The risks according to Babbie (1989) include: provincialism, hasty conclusion. questionable cause, suppressed evidence and false dilemma.

The group of students selected to carry out the evaluation was chosen for three reasons. Firstly, the 39 students could be divided into two distinct categories, 20 students who were enrolled for degrees in the Agriculture and Horticulture Faculty and 19 who were studying completely unrelated subjects. This provided a wide range of backgrounds so that different user views could be ascertained. Secondly, all these students were enrolled on a course for non-computing specialists. With few exceptions, their experience with computers was restricted to the limited use of packages on either the IBM or Apple Macintosh. Thirdly, 6 weeks of the 24 week course were dedicated to teaching about knowledge-based systems. By the end of this period the students were expected, in groups, to have a simple expert system running. The author was not involved with the course in any way and so could not influence the outcome of the survey results.

The students themselves rated their knowledge of crop disorders on a scale of 1 to 5 (where 1 = Poor and 5 = Good). The results are shown in Figure 3.6.

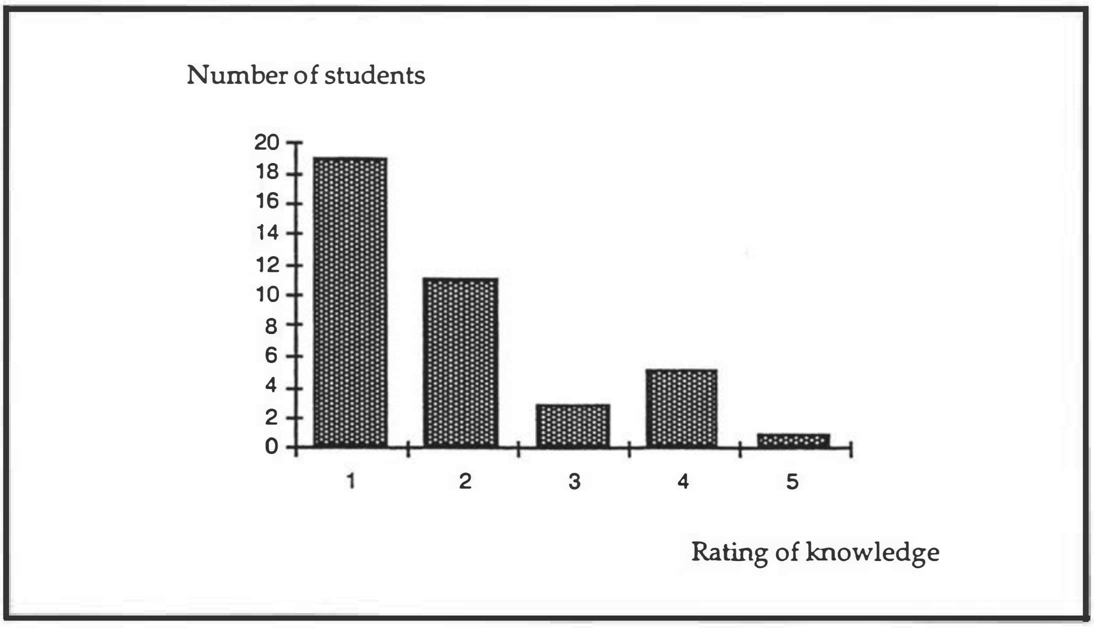


Figure 3.6 Student rating of domain knowledge

One of the goals of the survey was to find out whether there was a possible link between user characteristics and the kind of dialogue features required. All the students were non-specialists in the area of computing with a limited knowledge of expert systems. Their domain background varied considerably, however, and three groups could be distinguished. There was a large group of complete beginners (19) who knew nothing of the topic (henceforth known as group A). Their responses could be compared with the 11 students with some limited knowledge of the domain (group B). Finally, the 9 students who had some competence in the domain were aggregated into group C. Whilst these users were university students, some had extensive domain knowledge and were akin to experts.

Obviously, some students may have classified themselves wrongly, under-estimating or over-estimating their level of expertise. One way to check this was to examine the composition of each group and see in what faculty the students assigned to it had enrolled. Whilst this is, in itself, not definitive, it provides a crude test. One would expect, just from looking at the qualifications being sought, that the majority of Business and Arts students would be in group A with Horticulture students forming the nucleus of group C. This proved to be the case (see Table 3.1). All the Arts students and 64% of the Business students rated themselves as complete beginners. Of the 4 Business Studies students in group B, one had previously taken papers in Horticulture, whilst the only Business student in group C (enrolled for a Masterate), had already completed an Agriculture degree. Conversely, the Horticulture students, as anticipated, formed part of group B and the majority of group C. The students who cannot be so easily classified are the Agriculture students who have quite diverse backgrounds with some specialising in animal husbandry whilst others were more interested in the related area of soil management. A discussion of the major findings of the survey follows. (The complete questionnaire appears in Appendix A and the responses in Appendix B).

Faculty	Group A	Group B	Group C
Agriculture	5 26%	4 36%	2 22%
Arts	5 26%	0 0%	0 0%
Business	9 47%	4 36%	1 11%
Horticulture	0 0%	3 27%	6 67%

Table 3.1 Analysis of each group by faculty

To check that the students were actually able to use the expert system, they were asked whether they could understand what the expert system was doing. As can be seen from Figure 3.7, only five individuals felt ill at ease with the system (a 1 or 2, response). There seemed to be almost universal agreement that they could confidently interact with the system whatever their level of expertise.

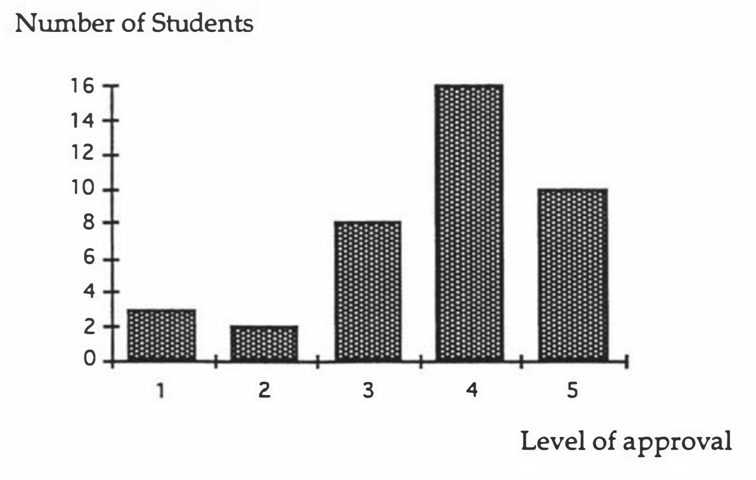


Figure 3.7 Question 17
"Do you feel that you know at each stage what the expert system is trying to do?"

3.4 Analysis of results

The students were asked to evaluate three methods of inputting information: English phrases and sentences (also referred to, in this thesis, as natural language), menus or direct manipulation, that is using a mouse to point and click on the screen. For this and all following questions a 1 or 2 response indicates dissatisfaction, whilst 3, 4 and 5 show increasing levels of approval. As the histogram shows (Figure 3.8), 19 of the 36 respondents were unhappy with inputting information using natural language.

When the results were analysed, an interesting difference between group A and the other two groups surfaced. Only 32% of group A responded in the satisfactory plus range (3 or over) whilst the figures for groups B and C were virtually identical at 55% and 56% respectively. Possibly the beginners felt that they would be lost for the words, not being familiar with the domain vocabulary. Menu input received a higher approval rating than natural language with 23 users answering with a 4 or 5 whilst direct manipulation was more popular again with 24 out of the 38 responses at the 5 level (Figure 3.9).

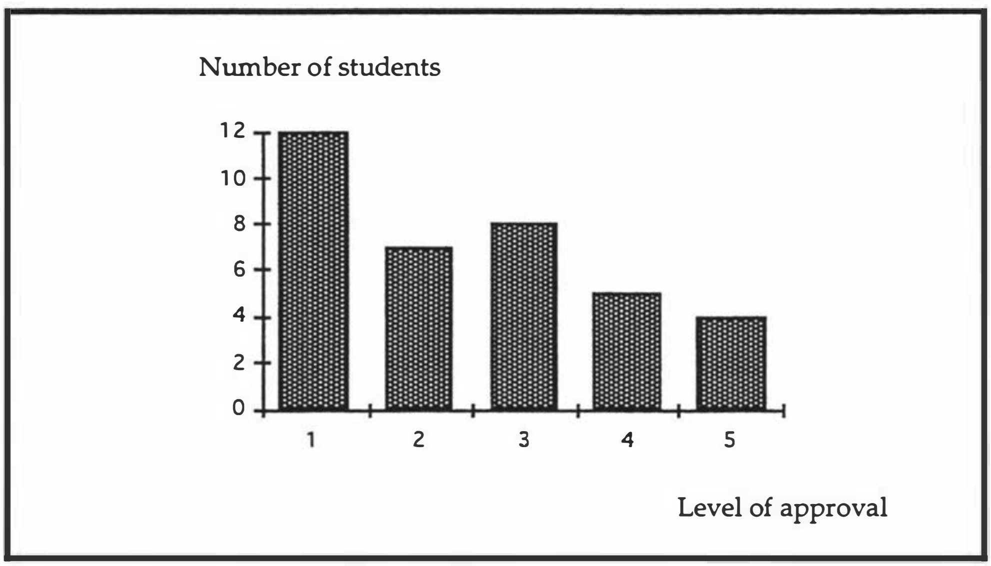


Figure 3.8 Use of English phrases and sentences to input information

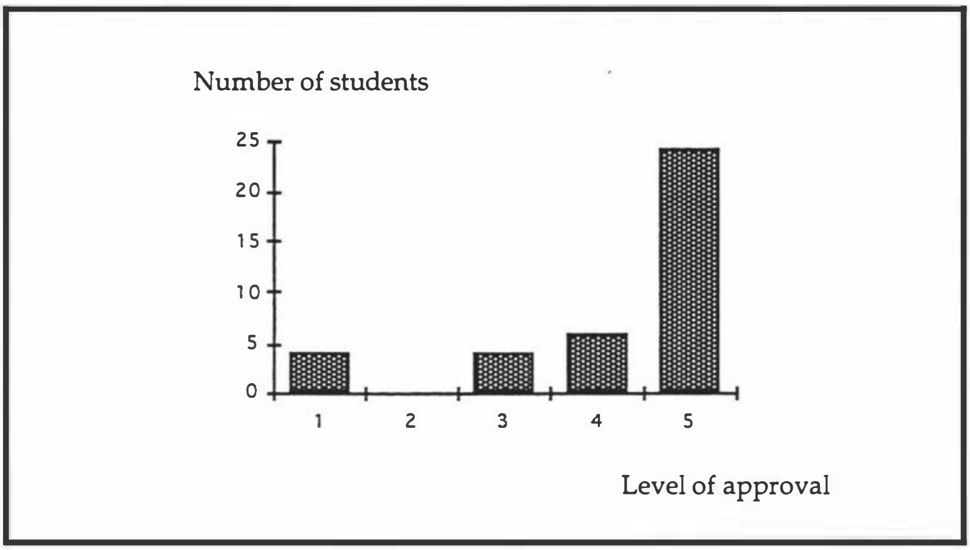


Figure 3.9 Use of direct manipulation to input information

It is clear that all groups felt at ease with both these modes of knowledge input in contrast to entering information using English (implicitly in the context of this study from the keyboard though English input via speech might be seen differently). Whilst groups A and C preferred direct manipulation input, Group B favoured menus. A clear picture of each group's response can be seen in Table 3.2. Unless otherwise stated, a table shows the percentage of each group responding at a satisfactory or higher level, that is in the 3 to 5 range.

	Group A	Group B	Group C
Alternatives			
English	32%	55%	56 %
Menus	68%	91 %	89%
DMI	84%	82 %	100 %

Table 3.2 Percentage of users satisfied with each presentation method

With regard to conclusions, such as the likelihood of certain diagnoses, respondents were asked whether they would like them to be presented in English phrases and sentences, graphically or as numerical values on a scale. The APPLE diagnosis system presents them graphically. In contrast to the results for information input, there was widespread support for conclusions in natural language with only one person answering below the 3 level (Figure 3.10).

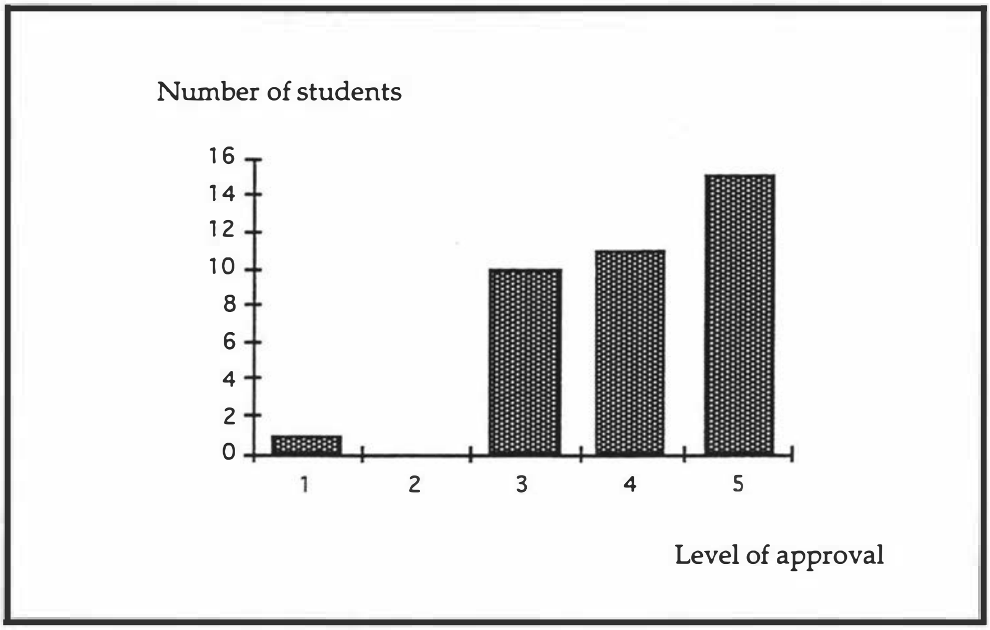


Figure 3.10 Presentation of conclusions in English phrases and sentences

There was similar approval for the graphical presentation of information but a more mixed response to the numerical option (Figure 3.11). When examining responses by group, it transpired that 100% of the members of group C answered with a 3 or above. The corresponding figures for groups A and B were 47% and 36% (see Table 3.3). It may be inferred that users with domain knowledge can interpret numerical output more easily than those who are less well-informed. Perhaps their grounding in the domain allowed them to translate easily between states of belief and their numerical equivalents. Group C, in fact, showed no clear preference for any particular method of information output, supporting all three alternatives equally. Those more expert in the domain seemed able to cope with all modes of information output and would have to be consulted about which options they would want in practice.

Whilst natural language input (by keyboard) is the least popular means of data input, it is seen as a perfectly acceptable option for output. Moore and Swartout (1990) developed a hypertext-style interface that allows a user (a student learning LISP) to point to some portion of the system's natural language explanation that needs to be clarified. A menu of relevant questions about that portion of the text is then displayed and the student selects the appropriate one. Whilst the rationale was to avoid some of the problems that could occur when parsing sentences input by students, the findings from this study indicate that such an approach matches user preferences. Explanations are given in English whilst direct manipulation techniques are used to highlight a problem area or select a question from a menu of alternatives.

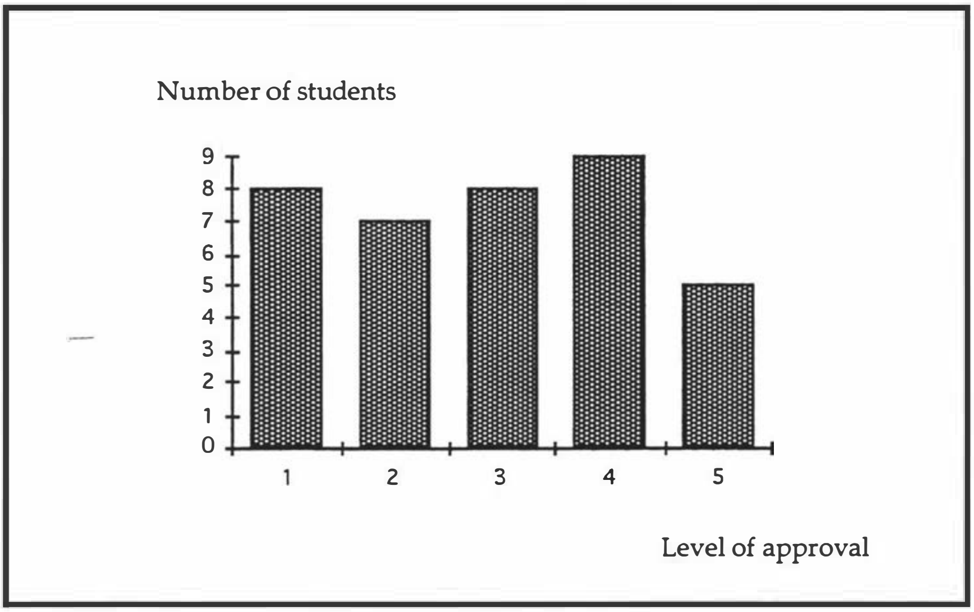


Figure 3.11 Presentation of conclusions by numerical table

	Group A	Group B	Group C
Alternatives			
English	84%	100 %	100 %
Graphics	84%	82 %	100 %
Numerical	47%	36 %	100 %

Table 3.3 Percentage of users satisfied with each output method

Some of the information that has to be input to a knowledge-based system is a matter of guesswork since data may be incomplete or the user may be uncertain. For this reason, answers often have to be qualified. Consequently, users were asked how they would prefer to indicate confidence in their answer to a question: using text, numerical input or graphically (that is moving a pointer along a linear scale). The graphical alternative was very well received (Figure 3.12) but there was less enthusiasm for the other two options, particularly numerical input which was rated by over a third of the respondents at 1 or 2 (Figure 3.13). Bearing in mind the previous findings about numerical input, it was hypothesised that the expert group would find the numerical option more acceptable than the other groups. This was substantiated by the analysis which showed that the percentage of users responding to this question with 3 or over was 63% for group A, 55% for group B and 78% for group C.

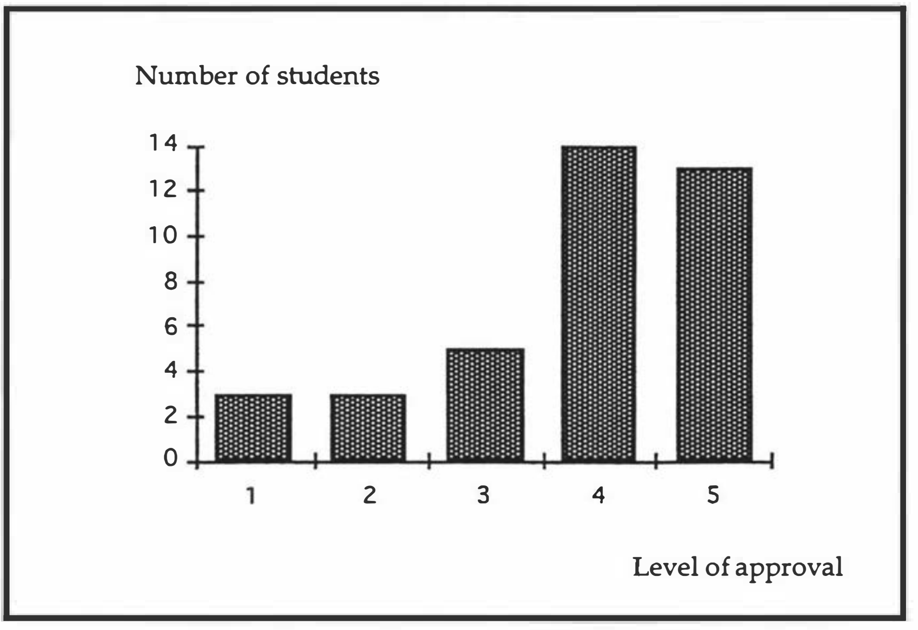


Figure 3.12 Indicating confidence in an answer by moving a pointer along a linear scale

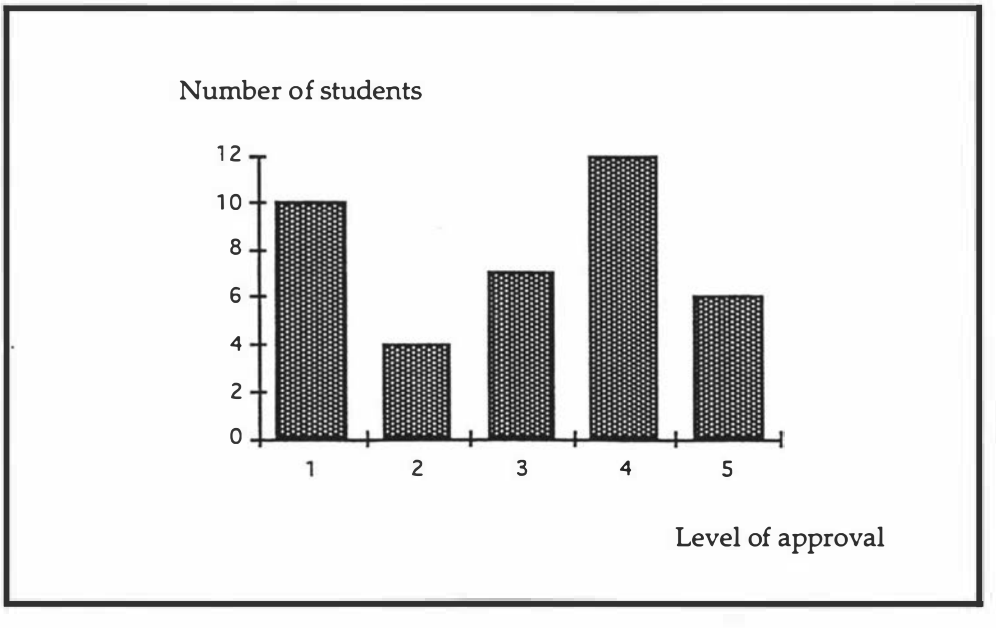


Figure 3.13 Indicating confidence in an answer by numerical input

To summarise the results for this question, groups A and B preferred to indicate confidence in their answer by dragging a pointer along a linear scale whilst for the members of group C this alternative rated equally with the text option (see Table 3.4).

	Group A	Group B	Group C
Alternatives			
Text	74%	55%	89%
Numerical	63%	55%	78%
Pointer	79%	82%	89%

Table 3.4 Evaluation of methods for indicating confidence

Explanations of both terminology and the problem solving capabilities of the knowledge-based system are provided by the APPLE system. The students were asked to rate the usefulness of these features. Consider first the "Clarify" option which provides definitions of the technical vocabulary used in the system. The histogram shows that there was considerable user support for this (Figure 3.14). A breakdown of the answers by group, however, indicates that members of group A were not quite so appreciative as the other users with only 58% responding with a 4 or 5 (see Table 3.5). This implies, as one would expect, that some of these complete beginners felt that definitions were not sufficient to enable them to understand what was happening.

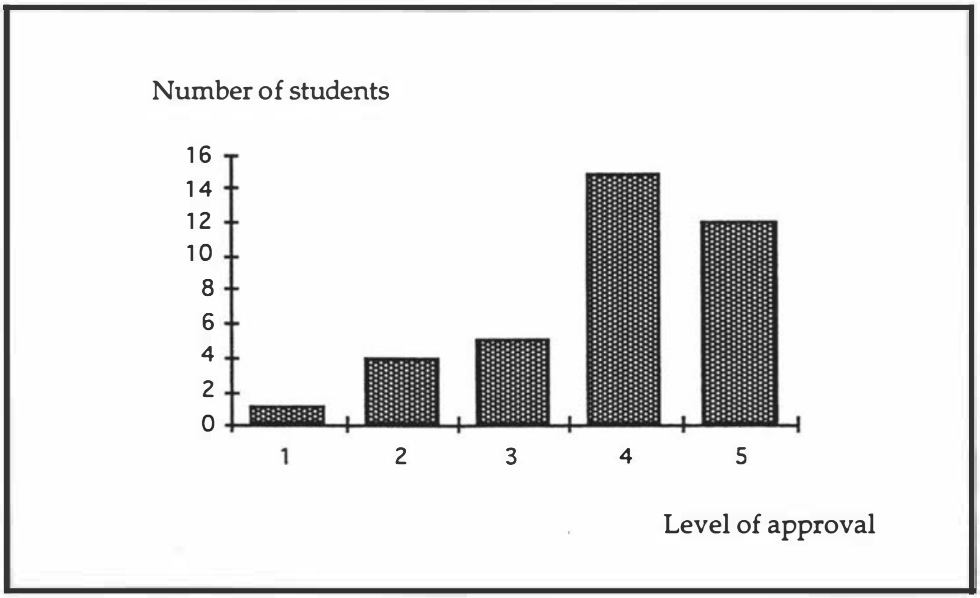


Figure 3.14 Helpfulness of Clarify

Response	Group A	Group B	Group C
Missing	10%	0%	0%
1, 2	21%	0%	11%
3	10%	18%	11%
4, 5	58%	82%	78%

Table 3.5 Usefulness of Clarify option

Overall, the results for the "Why" option (which allows a user to find out what hypothesis is currently being investigated) were very similar to those already described for "Clarify" (Figure 3.15). Group A's answers were almost identical to those for the previous question with Groups B and C a little more enthusiastic at the 4 and 5 level (Table 3.6). Taking the responses to both the "Why" and "Clarify" options together, group A members seemed a little more tentative in their support than other users. This is hardly surprising given their lack of background.

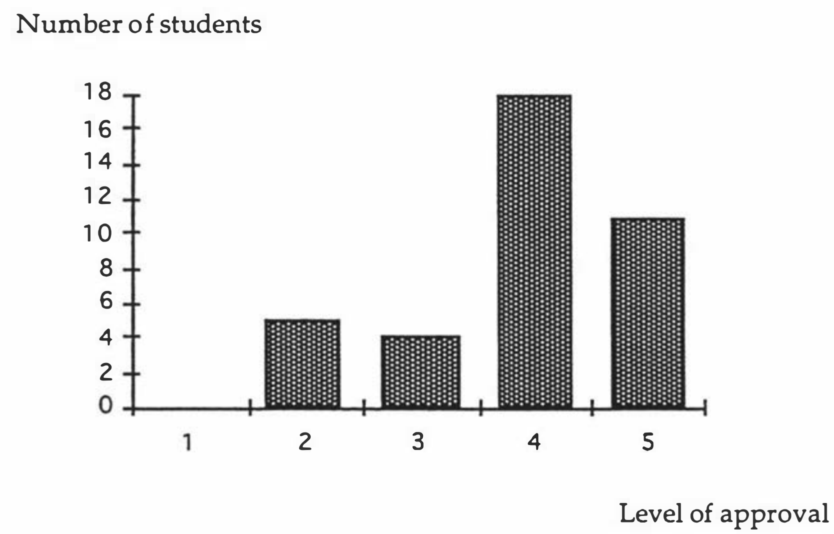


Figure 3.15 Helpfulness of Why

Response	Group A	Group B	Group C
Missing	5%	0%	0%
1, 2	21%	9%	0%
3	16%	0%	11%
4, 5	58%	91%	89%

Table 3.6 Usefulness of Why option

Four explanation features (see section 2.1.4) that were not provided by APPLE were also rated by the students:

- showing how a conclusion was reached;
- permitting interactive dialogue;
- justifying an answer using deep knowledge;
- providing examples.

These features are each discussed in their turn. Firstly, users were asked if, at the end of a session, they would like to see the reasoning behind the conclusions given. There was overwhelming support for such a function with no responses below 3 and thirty-two at the 5 level.

Secondly, Question 12 asked users whether they would like to engage in a dialogue with the system, asking questions whenever they wanted. This facility was suggested in view of the support for interactive dialogue in the literature (Moore and Swartout, 1989; Cawsey, 1993). Thirty-two people rated this feature at 3 or more (Figure 3.16), indicating that most users would welcome an addition of this kind to APPLE. It is not clear, though, whether users saw their side of the dialogue as involving keyboard or speech input. The most enthusiastic support for interactive dialogue (those who responded with a 4 or 5) was shown by group C (see Table 3.7). Overall, group C members seemed to welcome any opportunity to interact with the system and check its workings. Group B members on the other hand appeared to prefer the known ways of obtaining explanations, using "Clarify" and "Why", to the unknown possibilities of dialogue (see Tables 3.5 and 3.6). In contrast, the results for group A were exactly reversed. Interactive dialogue ranked higher than the other two alternatives. Perhaps these users wanted as much help as they could get.

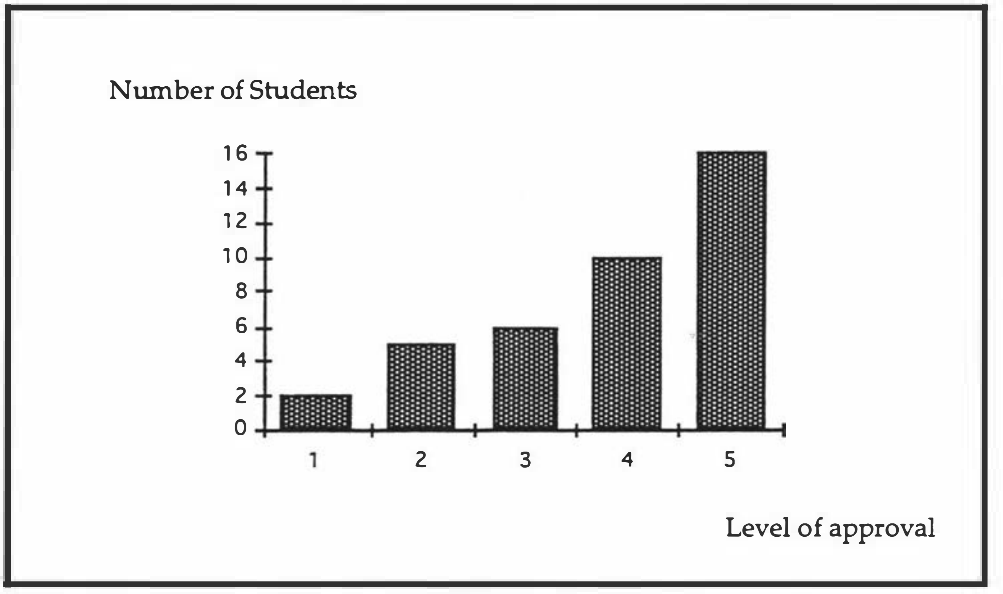


Figure 3.16 Question 12
"Would you like to engage in a dialogue, asking questions whenever you wanted to?"

	Group A	Group B	Group C
Response			
1, 2	16%	27%	11 %
3	16%	27%	0 %
4, 5	68%	46%	89 %

Table 3.7 Evaluation of dialogue option

Thirdly, explanations can be provided which justify an answer using a deep model of the domain. Information could, for example, be provided to explain why diseases occur. Students were asked, therefore, "Would you like answers that relate to the underlying causes of problems, such as relevant biochemical information?" Over two thirds of the students agreed strongly with this, that is they responded with a 4 or 5 (see Figure 3.17). Surprisingly, there was little difference between the results on a group basis (see Table 3.8).

Response	Group A	Group B	Group C
Missing	5%	0%	0%
1, 2	11%	18%	11%
3	16%	18%	11%
4, 5	68%	64%	78%

Table 3.8 Support for Biochemical option

We had believed in advance that group A would evince little interest in biochemical information in contrast to groups B and C. Presumably, group A's desire for biochemical knowledge reflects the need students felt to fill in background details. We had also expected that all the expert students would want to see something more than heuristics in explanations and had anticipated that every member of group C would respond with 3 or more. This was almost but not quite the case.

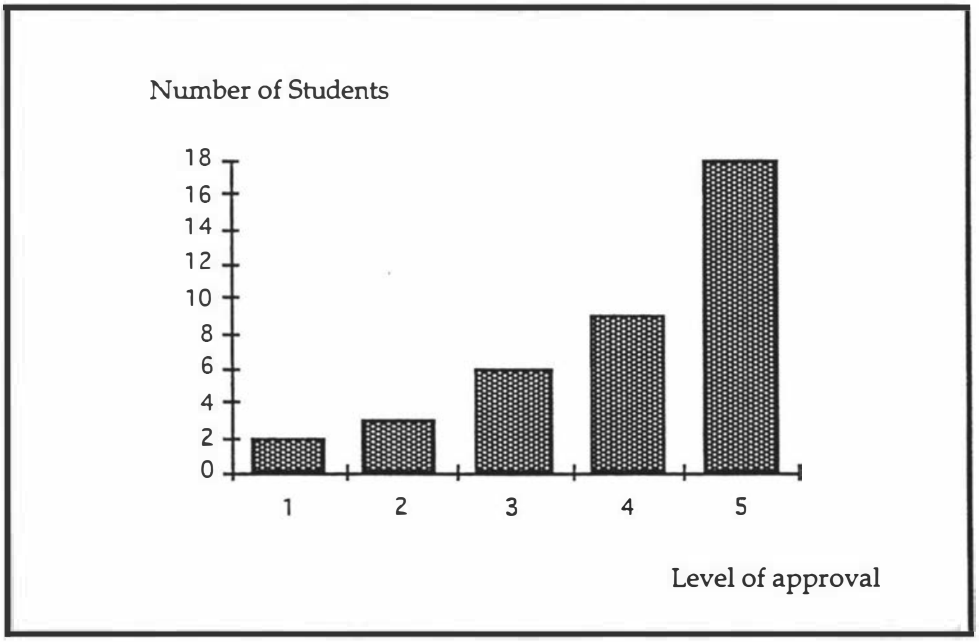


Figure 3.17 Question 15
"Would you like answers that related to the underlying causes of problems, such as relevant biochemical information?"

Fourthly, the literature suggests that users want the provision of examples (Gregor, 1991; Wognum and Mars, 1991). A useful source of knowledge for users in this domain is graphical information, for instance pictures of diseased leaves. These could be used when eliciting data from students and to make explanations more comprehensible. Consequently, students were asked "Would you find pictorial information like pictures of diseased leaves helpful?" The responses to this were very positive with 30 students responding at the 4 or 5 level. Clearly, using a database of pictorial information for purposes of explanation would provide an added dimension that users would find helpful. We had speculated that this information would be of least value to group A and most help to those more expert who might prefer to work with the information in its most natural form. This theory was confirmed to some extent as everyone in group C responded at the 4 or 5 level (see Table 3.9). Again, though, group A showed a remarkable enthusiasm for information. Looking at their answers overall, the students in this group seemed to rate more highly features that were proposed but not incorporated into the system. They appeared to hope that such features would provide them with information that would make the system more comprehensible.

Response	Group A	Group B	Group C
1, 2	10%	18%	0%
3	16%	18%	0%
4, 5	74%	64%	100%

Table 3.9 Support for pictorial option

Overall, some caution must be exercised when interpreting the results that relate to explanation facilities; they show what explanation features users say they like rather than what they would use. In particular, the results for group A must be discounted to some extent because they were clearly trying to find out more about a domain of which they had no knowledge. Gregor (1992), it should be noted, observes that subjects with a prior knowledge of the subject use the explanation features more than the others. Further analysis of the results in tables 3.5 to 3.9 reveals that, with one exception, members of group C display a higher level of interest in explanation facilities than members of other groups (Table 3.10). This indicates, as does Gregor's research, that enthusiasm for explanation requirements may be related to a user's level of expertise.

Feature	Group A	Group B	Group C
Clarify	68.5%	100%	89%
Why	74%	91%	100%
Interactive dialogue	84%	73%	89%
Deep model	84%	82%	89%
Examples	90%	82%	100%

Table 3.10 Percentage of each group responding at the 3-5 level

If group A are eliminated from further consideration because of their lack of domain knowledge, it is possible to re-consider the results for groups B and C. These are the users who can be described as having a limited and good knowledge respectively of the domain. Boxplots can be used to compare the results for each group. With interval data, boxplots show the overall level of values, the spread of the data, outliers, how the levels of the variables compare and how the spreads of the variables compare. Boxplots are based on the median; unlike measures that use the mean they not affected by outliers. For this reason they are a suitable tool for analysing exploratory data.

The results (see appendix C) indicated that the differences between the two groups with regard to menu input, numeric presentation of output and interactive dialogue were statistically significant. A Chi-square test also showed a relationship between level of expertise and whether or not Bio-Chemical information should be presented to users of APPLE. These findings indicate that user-system interaction and presentation of information may be related to a user's level of domain expertise. Further studies would have to be undertaken to confirm whether this is the case. In the context of this thesis, the findings re-inforce the advice from the HCI literature about the importance of user modelling. If there are user groups with differing levels of expertise, the knowledge engineer should determine whether they have their own requirements for interaction and presentation of material during the analysis phase.

3.5 Conclusion

An initial survey was undertaken to determine what features might be required in a knowledge-based system to make it usable. Thirty-nine students were asked to run the APPLE knowledge-based system and answer questions about methods of information input and output, and explanation facilities as well. Overall preferences were noted as well as group differences. Menus and direct manipulation were both well supported as means of information input whilst natural language and graphical ways of presenting output also received a high approval rating. Moving a pointer along a linear scale to indicate confidence in an answer was far more popular than the text or numerical alternatives. With regard to explanation facilities, the "Clarify" and "Why" options were found very useful by members of groups B and C. There was also general agreement that causal knowledge (biochemical explanations), examples (pictorial representations of the domain) and detailed explanations of how conclusions were reached would be helpful. The results of this study indicate that users are concerned about interface and explanation facilities.

On the whole, the more expert group felt confident about handling information in virtually every form (see Tables 3.2, 3.3, 3.4). With the exception of the natural language input option, there was never less than 78% of group C who responded in the 3 to 5 range. It is hard to avoid the conclusion that the more expert group can usually cope quite easily with whatever is provided (personal communications from Dan Diaper and Sunil Vadera would tend to support this). This does not mean, however, that they would not have preferences for a particular application if consulted. Such flexibility was not displayed by users with little or no domain knowledge. These users had their likes and dislikes which were not always identical. Group B students, for instance, rated menus very highly, perhaps preferring the structured way of interacting. The possibility undoubtedly exists that, at times, there is a link between users' domain background and the way they wish to communicate with a knowledge base.

To conclude, a study was undertaken to see if users had strong views about the interface of a knowledge-based system. From the results presented, some trends are clearly discernible such as the overwhelming support for a direct manipulation interface. The presentation of information in its natural form, pictures of apple trees, for instance, was also welcomed. On some occasions, though, there appeared to be a relationship between the users' level of expertise and preferred interface feature; outputting information in a numerical form, for example, was acceptable only to the most expert users. Groups with different levels of expertise may well have preferences that have to be elicited. Whilst these results are from one survey only, they confirm the emphasis of researchers on the need for easy to use systems, cognitive consistency with the task and the grouping of users on the basis of their domain expertise. The knowledge engineer should take these factors into account during the analysis phase of a knowledge-based system. Another factor that has to be considered, although it was not relevant in this study, is the computing experience of users. Given the importance of these issues, at what point in the system development process should they be considered? Before answering this question, the many life cycles for knowledge-based system development must be examined to see how these crucial activities have been tackled.

Chapter 4

A Review of Knowledge-based Systems Life Cycles

The development of knowledge-based systems is no longer just the province of researchers in Artificial Intelligence. Once the technology was established, many organisations became interested in building commercially viable products. To achieve this goal, the application of software engineering principles has been suggested, revolving around the selection of an appropriate life cycle. In this chapter several approaches to expert system development are described with emphasis on the analysis phase. It is at this stage that, as previously argued, user concerns should be taken into account. The major debate about the role of prototyping is also addressed. Finally, some important principles for knowledge-based systems development are identified.

4.1 Software engineering paradigms

Many difficulties have beset software development; frequently the completed product has not met user requirements whilst exceeding both time and budgetary constraints (Hickman *et al.*, 1989; Pressman, 1992). Software engineering techniques centred upon a systems life cycle have been introduced to remedy this situation. The classical systems development life cycle is the waterfall model, a sequential approach that involves working in turn through analysis, design, coding, testing and maintenance stages (Pressman, 1992). This stepwise methodology ensures that the tasks are clearly defined and each one is accomplished before progressing to the next. It fosters a systematic approach but at the expense of flexibility. One major criticism is that software development is usually an iterative rather than a linear process. Another is the difficulty that customers have in expressing their requirements. Feedback provided by users comes too late in the stage to impact on the analysis phase (Gonzalez and Dankel, 1993).

One alternative to the waterfall model is prototyping where a small scale version of the final system is built for users to check that their requirements have been met. Another is Boehm's (1988) spiral model which provides a framework that allows for iteration and prototyping as well as an analysis of the risks. During each cycle (rather than stage), the objectives are identified, the risks evaluated, relevant outputs developed and a decision made about whether or not to proceed. Pressman (1992) comments that the spiral model paradigm for software engineering is currently the most realistic approach to the development for large scale systems.

Whatever paradigm is selected, Pressman (1992) notes that there are three generic phases: definition, development and maintenance. The definition stage, itself, comprises the three steps of system analysis, software project planning and requirements analysis. Essentially, the role that the software will play is identified, its scope defined and the detailed specification produced. Development involves software design, coding and testing. Design entails describing the algorithms, data structures and interface characteristics of the future system. Programming and testing can then take place. Lastly, there is the maintenance phase which deals with three types of change: correction, adaptation and enhancement.

These definitions from the software engineering literature are a useful starting point for a discussion of life cycle issues in knowledge-based systems.

4.2 Life cycle issues in knowledge-based systems

There is an on-going debate about the life cycle of an expert system. Historically, AI researchers implemented systems without any regard to analysis and design as they were more concerned with testing out the technology (Sell, 1985). Sell observed that expert systems are pieces of software that should be subject to the same engineering discipline as other pieces of software. According to him, without using appropriate methods and techniques, expert systems have uncertain foundations, are difficult to modify and maintain, present an awkward user interface and ignore resource constraints. Nonetheless, his discussion of the life cycle is rather informal. He drew attention to the personnel requirements of the development team, suggesting that there were three different roles to fill, those of the knowledge engineer, the domain expert and the implementor. Decisions must then be made about hardware and software; the selection of a suitable machine and appropriate expert system building tools or shells that will run on it. Finally, he discussed the problems of knowledge acquisition and knowledge representation. The validation of the knowledge base and the provision of facilities such as explanation and justification were mentioned in later chapters of his book but were not integrated into an overall framework. His book, though, is important because it mentions problems with the interface and tries to bring the principles of software engineering to bear on the problem. This can be seen as a somewhat skeletal attempt to define the stages of an expert system life cycle. It was not the first, however, to discuss problems that arise when building expert systems.

Earlier, Quinlan (1982) identified what he described as three design issues facing knowledge engineers: the representation of knowledge so that it could be used by the system and understood by human beings, the selection of an appropriate architecture for problem solving, and the acquisition and testing of knowledge to ensure internal consistency and completeness. Whilst raising important questions, this approach still does not deal with the software engineering issues that a life cycle usually addresses.

Hayes-Roth, Waterman and Lenat (1983) were advocates of a more systematic approach in the text "Building Expert Systems". They described (p.24) the principle stages in the evolution of an expert system as follows:

- *Identification*, determining problem characteristics;
- *Conceptualisation*, finding concepts to represent knowledge;
- *Formalization*, designing structures to organise knowledge;
- *Implementation*, formulating rules that embody knowledge;
- *Testing*, validating rules that embody knowledge.

This methodology for building expert systems embraces many activities from identifying the problem area and defining its scope to evaluating the performance of the system. During the implementation stage a prototype program is developed which is later tested. Problems that arise may need to be solved by returning to an earlier phase such as conceptualisation to re-formulate the key concepts. Consequently, this methodology includes both iterating round the stages of the life cycle and prototyping a working model of at least some parts of the system. In an elaboration of this methodology, Buchanan *et al.* (1983) suggested that user issues should be dealt with in the testing phase. Once the first prototype had demonstrated the feasibility of the approach, a friendly interface should be added so that the intended users could experiment with it. This also assisted the knowledge engineer with system development and debugging. No advice was given, however, about how to develop this friendly interface. Overall, the approach seems heavily dependent on prototyping.

4.2.1 The role of prototyping

Waterman (1986) clarified the role of a prototype in knowledge-based systems development showing that it could be used for many different purposes:

- Demonstration prototype: a system that can solve parts of a problem and is used to show the potential value of an expert system to management and prospective sponsors.

- Research prototype: a system with functions for handling most areas of a problem, although some areas may require research in the problem domain.

- Field prototype: a system supporting all functional areas of the problem that can be tested with end users for effectiveness and utility.

Prototyping is seen as conferring many advantages. A prototype is a tangible product which can be used to show the feasibility of a project and obtain management support. It can reveal gaps that require further research and allows users to comment on the functionality and usability of the proposed system.

Because of the iterative nature of knowledge-based development, many others have based their expert system life cycle around prototyping (Harmon and King, 1985; Bobrow *et al.*, 1986; Martin and Oxman, 1988; Guida and Tasso, 1989; Weitzel and Kerschberg, 1989; Roberts, 1990; Sacerdoti, 1991; Turban, 1992; Gonzalez and Dankel, 1993; Klut and Eloff, 1993). Often rapid prototyping is advocated. With rapid prototyping a small-scale system is quickly designed, built, and tested. The expert and users assess the performance of the system and, if improvement is needed, the system is modified. The prototyping phase is embedded within the life cycle.

Harmon and King (1985) distinguished between the approach used to build small and large expert systems. Small expert systems, those which incorporate about 75 to 200 rules, can be built by non-specialists if suitable tools are chosen. A small application can easily be prototyped until it handles all cases. With large-scale development, that is when a team of people is required, a six step process should be followed: selection of the appropriate problem, development of the prototype system, development of the complete system, evaluation of the system, integration of the system and maintenance. The prototype developed in the second phase is a small version of the system that can be used to test out assumptions about the encoding of facts and the inference strategies of the expert. Assuming that a prototype can be built that works successfully on case studies, the knowledge engineer expands the prototype into a full system. The basic design of the knowledge base may be reviewed. Rapid prototyping is used to determine the basic structure of the system but further work and alterations will take place at this stage. Once this has been completed the interface has to be tailored to the users. Harmon and King thought that considerable attention should be paid to providing explanations for the user about the system. Displays that allow the user to follow the system's reasoning process were seen as a key to winning user support. They highlighted the danger of producing a system that works but is unacceptable to users.

Writing at much the same time, Partridge (1986), believed that the problems of dealing with AI problems could best be served using an approach called the "run-debug-edit cycle". This methodology is similar to the rapid prototyping paradigm. The run-debug-edit cycle aims to get a good initial approximation off the drawing board and onto the computer. This is an incremental development approach which results in the evolution of a machine-executable problem specification. Whilst Partridge saw the need for some requirements analysis, he argued that intangible human needs were part of the essence of the problem. He took an essentially pessimistic outlook, saying, "In AI we start wherever we can and hope to emerge with a behaviourally adequate problem" (Partridge, 1986, p.129).

The perils of developing knowledge-based systems were also dealt with by Bobrow *et al.*, (1986). They provided guidelines for developing expert systems that should meet the expectations of users. Their life cycle was a refinement of that proposed by Hayes-Roth *et al.* (1983). After problem identification and conceptualisation, a prototyping stage allowed *inter alia* the development of a suitable user interface. Testing and redefinition then followed and the life cycle concluded with a maintenance phase.

Rolston (1988) advocated taking advantage of the work done in software engineering even though there are significant differences between traditional software and expert systems. The expert system development life cycle that he suggested is an iterative version of the traditional life cycle. It has the following stages: problem selection, prototype construction, formalization, implementation, evaluation and long-term evolution. Rolston claimed that it has several unique features:

- the customer and domain expert are involved throughout the process;
- frequent demonstrations of work to date are encouraged;
- change is viewed as healthy and is particularly encouraged during the prototyping phase.

Prototyping is seen as having three functions. First, it helps the knowledge engineer to gain a deeper understanding of the nature and scope of the problem. Secondly, it can be used to demonstrate the system functionality. Thirdly, a prototype can be developed to test the initial design decisions. Essentially, prototyping helps the knowledge engineer and the customer to understand the problem. It can reveal unresolved issues that lead to the modification or even re-implementation of the prototype. Only when a satisfactory prototype has been developed, should a decision be made about proceeding. At the formalization stage, all necessary planning should be done before moving from a specification to design, otherwise, Rolston warned the readers to expect what he terms a "kludge". Obviously, his life cycle aims to ensure that expert systems development is a disciplined process. There is no mention, however, of the interface. Possibly, the input from the customer ensures that this matter is addressed.

Weitzel and Kerschberg (1989), who developed the Knowledge-Based-System Development Life Cycle (KBSDLC), believed that previous prototyping methodologies did not address the challenges of building knowledge-based systems. The KBSDLC can be used in two ways. With rapid prototyping the design specification for a standard implementation of the knowledge base is provided. Evolutionary prototyping can also take place, that is the final prototyped expert system is kept as the production system. The emphasis in

KBSDLC is on the elicitation of knowledge and the end-user is not even mentioned.

In a review of expert system life cycles, the claim was made that there was a lack of sound and general methodologies for large-scale expert system development (Guida and Tasso, 1989). The authors went on to describe the characteristics of an expert systems development methodology. It should be modular, complete, effective, efficient, practical and flexible. The proposed life cycle has a typical waterfall structure with five phases. A plausibility study is followed by a demonstration prototype construction. This demonstration prototype allows user requirements to be identified at an early stage. As with Harmon and King (1985) this stage leads on to a full prototype construction. Then the target system is implemented and the operation and maintenance phase is entered. There is not such a heavy dependence on prototyping as there is with Harmon and King (1985), since phase two, a demonstration prototype, can be omitted if the plausibility study shows there is no need for the demonstrator. The full prototype of phase three is developed in accordance with either the initial plausibility report or the revised version that incorporates alterations that arise from experience with the demonstrator. An empty system, that is a skeletal system with an empty knowledge base, is fully developed and tested. The user interface is considered at this point. This prototype is usually completely different from the demonstrator. Evolutionary prototyping takes place if the target system that is produced in phase four is merely a refinement of the prototype. The prototype may need to be modified to a greater or lesser extent, though, because of the target environment. Overall, prototyping is used in a very flexible fashion in this life cycle. In the early stages it can help refine the users' requirement as identified at phase one or even be omitted. Subsequently the prototyping that takes place may be but does not have to be evolutionary.

Turban (1992), too, recognised the value of prototyping, believing that rapid prototyping is essential when developing large systems because of the high costs of a poorly structured and, hence, unused expert system. Moreover, he pointed out that prototyping allows the developers two distinct approaches to system development. Following the evolutionary model, a prototype becomes the interim working version which is modified several times until it is complete. The other option involves developing a model which is thrown away and a new design drawn up. He also gave a comprehensive list of advantages conferred by rapid prototyping (see p.423). As well as those mentioned previously, he cited the following:

- it allows the possibility of an early midcourse correction of the project based on feedback from managers, consulting experts and potential users;

- it might provide a system with enough utility [sic] that it can be put in the field on an extended basis;

- it provides an accelerated process of knowledge acquisition;

- it provides information about the initial definition of the problem domain;

- it helps to sustain the expert's interest.

Whilst some of these benefits may be obtained when using prototyping, this will not always be the case. Perhaps Turban somewhat overstated the case. There are situations where extensive interviewing of many experts has to take place (Todd *et al.*, 1993) and prototyping does little to speed up the process of knowledge acquisition. It is not quite clear either what is meant by an early midcourse correction. Nonetheless, given this level of support for prototyping it is easy to see why Larry O'Brien (1993) recently claimed that the AI communities are leaders in the key area of rapid prototyping.

To summarise, prototyping is often used to assist with the elicitation of expertise and/or user requirements. Since the development of knowledge-based systems software is not a serial process, prototyping allows for any necessary iteration to take place. Those parts of the software model that are correct are kept whilst others are discarded. Another prototype is produced and so on until one becomes the working system or the specification for it. When the end result is the system itself, there may be serious doubts about its structure and robustness. Payne and McArthur (1990), who themselves recommend prototyping, outlined the problems that arise once an initial system is in place. Subsequent development is slow because the easy portion of the problem has been tackled leaving difficult paradoxical situations to be resolved. It is difficult, therefore, using the prototyping paradigm to predict how long the process of building a system will take and, consequently, there is no means of producing a realistic costing. A survey of 250 companies using prototyping in the data processing environment showed that whilst user involvement was one of its major benefits, a common problem was the large number of iterations of the prototype which increased the duration of the project (Carey and Currey, 1989). Moreover, an expert system has to represent knowledge correctly as well as meet the needs of both the organisation and the users. It is hard to see how the prototyping process can easily accommodate the elicitation of expertise as well as organisational and user requirements. In practice, prototyping appears to be used as a substitute for analysis. It was to avoid problems of this kind that the KADS methodology was developed (Hickman *et al.*, 1989).

The developers of KADS, the major deliverable of Esprit project 1098, were highly critical of rapid prototyping. The following comments about prototyping have been made over the years:

> *as a development paradigm it suffers from the fundamental flaw that the design of the prototype all too often evolves into the design for the operational system. The original design may well be appropriate for the prototype but completely inadequate for the full system and may well require costly and expensive re-writes.*
>
> (Hickman *et al.*, 1989, p.16)

> *Rabid rapid prototyping is an underestimation of the knowledge acquisition problem.*
>
> (Wielinga *et al.*,1987, p.102)

> *Rapid prototyping is a costly ad hoc development which often leads to poorly specified, poorly designed systems that do not address the real needs of the application.*
>
> (Touche Ross, 1989, p.1)

That there is more than a little substance to their claims can be seen from the confession made by Weitzel and Kerschberg (1989), developers of KBSDLC, who acknowledged that when developing Medclaim, a knowledge-based system for processing medical claims, they had to backtrack from the coding phase to the problem definition stage.

4.2.2 The KADS methodology

KADS, originally an acronym for Knowledge Acquisition and Documentation Structuring but now the name of a methodology, had its origins in an Esprit pilot project that started in 1983 (Hickman *et al.,* 1989). It was called "A methodology for the design of knowledge-based systems". Hickman *et al.* note that the initial concept was way ahead of its time with its emphasis on software engineering techniques when most AI work of a purely commercial nature was *ad hoc.* There seems to be little in the literature of the topic to contradict this view. The only comparable work for that period was described in the book "Building Expert Systems" by Hayes-Roth *et al.* (1983).

After the success of the initial pilot project, there were two more Esprit projects: P1098 (KADS-I) and P5248 (KADS-II). Simon Hayward (1986), manager of the first major Esprit project, firmly believed that commercial expert system development demanded more than an approach based on rapid prototyping. He observed that the AI community traditionally experiments and prototypes, taking the attitude "it doesn't matter much where you start, try something, see if it works and keep modifying it until you get something good enough for your purpose" (Hayward, 1986, p.196). At this early stage of Esprit project 1098, a skeletal life cycle was followed with four phases: knowledge acquisition, system design and implementation, testing and operational use.

Subsequently, the KADS life cycle described was a modified version of the waterfall life cycle with the following stages: analysis, design, implementation, installation, use, maintenance and knowledge refinement (Hickman *et al.,* 1989). An alternative version, the "Client-Oriented Normative Control Hierarchy" known as CONCH, was also proposed as a possible life cycle. It differs from the waterfall version in its adherence to Boehm's spiral life cycle paradigm but, in essentials, follows the KADS philosophy. Whatever framework is employed, prototyping can be used for specific purposes only and not as a methodology in its own right (Hickman *et al.,* 1989). It can be used, for instance, to investigate the appropriateness of a particular AI method or user interface. In these cases, prototyping should follow a life cycle of its own, to wit a definition of aims, design, implementation, use and evaluation. There have clearly been changes to KADS over the years but the researchers have always adhered to the principle of rigorous analysis before any design decision has been made. KADS is so important to the research presented here that its terminology and relevant models will all be described in some detail.

As the difficulties that face the knowledge engineer are formidable, KADS is based on several principles to assist with the construction of an expert system (Wielinga *et al.*, 1991, p.3):

- the introduction of intermediate models to cope with the complexity of the process;
- the use of the KADS four layer framework for modelling the required expertise;
- the reuse of partial models as templates to support top-down knowledge acquisition;
- the differentiation of simple models into more complex ones;
- structure preserving transformation of the expertise model into a design model and system code.

Several intermediate models are produced in the analysis phase. Initially, the organisational, application and task models are defined. The organisational model provides an analysis of the environment in which the expert system will function whilst the application model describes the problem that has to be solved. Only when these have been completed can the task model, which identifies the necessary functions of the knowledge-based system and the tasks it will perform, be specified. This real life task, as it is known, is further decomposed into a number of primitive tasks which are allocated by the knowledge engineer either to the user or system. To do this properly an analysis of user abilities and requirements is needed as well as knowledge of the potential capabilities and limitations of the system. From this a model of the required user-system cooperation, the model of cooperation, can be built up. This is merged with the model of expertise which specifies the problem solving behaviour of the knowledge-based system (and not the know-how of the expert) to form the conceptual model. The end product is not described formally but uses terms that are subjective and relative to the cognitive framework of the human user. Since the conceptual model is implementation dependent the design model has to take into account practicalities such as performance, hardware and software. Should there be the need to change requirements or build a new user interface, the conceptual model guides this process (Hickman *et al.*, 1989).

In the KADS methodology, there is great emphasis on analysis as a separate stage in the life cycle. It is seen as such a complex process that three different streams are suggested. The external stream handles the requirements of the organisation whilst the analysis of expertise and the development of the conceptual model are carried out in the internal stream. The last of the three streams proposed, the modality stream, aims to describe the different ways a system could be used. For each possible division of labour between the user and the system, a model of cooperation is produced. Hickman *et al.* (1989) note that the interrelationships between the streams of activity become very complex and hard to document.

The model of expertise produced as a result of this phase is independent of current technology and any particular implementation. Four layers that describe expertise are distinguished (Table 4.1). Whilst it is usual to incorporate knowledge of the domain this model goes much further. The inference level states what inferences can be made from the domain knowledge not how or when they are made, whilst the task layer defines ways in which knowledge sources can be combined to achieve a particular goal. The bottom three levels are always required but the strategic information is only needed for highly flexible systems which will deal with new kinds of problems. Models of this kind make it possible for a system to be checked at the design stage in a methodical way.

Layer	Elements	Relation to lower layer
Strategy	Plans, meta rules	controls
Task	Goals, task structures	applies
Inference	Meta classes, knowledge sources	describes
Domain	Concepts, relations	

Table 4.1 Layers of a model of expertise (Hickman *et al.*, 1989)

Interpretation models, which specify the inference and task levels of the conceptual model for elementary problem solving tasks, have been developed. The full taxonomy of these is shown in Figure 4.1.

- system analysis
  - identify
    - classify
      - simple classify
      - diagnosis
        - single fault diagnosis
  - heuristic classification
  - systematic diagnosis
    - causal tracing
    - localisation
      - multiple fault diagnosis
    - assessment
    - monitor
  - predict
    - prediction of behaviour
    - prediction of values
- system synthesis
  - design
    - hierarchical design
    - configuration
  - planning
  - modelling

Figure 4.1 KADS Library of generic tasks (Hickman *et al.*, 1989)

At the highest level, the major distinction is between analysis and synthesis. Analysis tasks involve establishing the unknown properties of objects within a domain as with classification, diagnosis and monitoring. Synthesis tasks on the other hand are concerned with structuring a possible object within a domain. As can be seen synthesis can be sub-divided into design, planning and modelling tasks. The output of the design process is a description of components and structure whilst planning produces a description of activities related in time. One of these models can be selected to drive the knowledge acquisition process and a domain dependent model produced. These models of the problem solving process are building blocks that allow the knowledge engineer to deal with the complexity of the construction process. By instantiating an interpretation model with domain knowledge, the model of expertise can be developed.

Even a comprehensive collection of interpretation models does not solve all the knowledge engineer's problems. It is possible (but unlikely) according to Breuker *et al.* (1987) that there is no model for the task. It is more likely that a real world problem will not automatically map to just one of these tasks. A real life model is a composite of generic tasks and it may be difficult to select the elementary tasks and link them. A further problem they mention is the degradation of one model into another. For example, what appears to be a monitoring problem can turn to diagnosis if it proves necessary to find the cause of a recurrent problem.

There is an even greater problem, though. How can the knowledge engineer decide at the start of a project what the problem solving task is and select a suitable generic model? Hickman *et al.* (1989) state that in the early stages of analysis it is not possible to make this decision. It is most important in the first instance to model the organisation's requirements. The resulting domain and inference layers may suggest an appropriate interpretation model. For instance, if the domain structures are predominantly "parts of" hierarchies then diagnosis is indicated. Inference actions, too, provide a clue. *Assess, specify* and *decompose* activities indicate analytic processes whilst *build, design* and *plan* point to synthesis. Despite this advice, it is accepted that choosing an initial model is not easy given familial resemblances such as exist, for example, between simple classification and assessment.

The knowledge engineer does not only have to analyse the expertise, however, but must also elicit and model the user requirements. Busche (1989) contributed the work on user modelling to the KADS project. She believes that user models can play varying roles: determining the interaction presentation style, controlling the semantics of interaction, as input to the system for task performance and controlling the system modality. The first two of these have already been discussed in detail (section 2.2.3).

The emphasis is on tailoring systems to users. A decision can be made to provide each group of users with a presentation style appropriate to their computing background. Moreover, user groups with different areas of responsibility can be given access to different subfunctions. Busche describes a prototype system for error diagnosis and repair used by two groups: operators and technicians. The solutions offered by the system to operators are less sophisticated than those for the technicians. The knowledge-based system has to know the type of user before providing the instructions for repairing the device.

User modelling is also needed when there is a specific distribution of labour between the system and each user group. In the repair example, both the operator and the technician receive instructions after a fault has been diagnosed. At some point, however, a technician is allowed to make suggestions about the cause of a fault whilst this task is performed by the computer when the user is an operator. Hence the task distributions are different and two separate modalities defined. This may also be the case when users have different goals - some wishing to learn about the domain and others to obtain answers.

Finally, user modelling can support task performance, for instance in consultation systems where answers are very dependent on personal information. Advice about superannuation is obviously linked to personal circumstances and expectations. It is probable that the knowledge-based system will have to build a model of the user before proffering advice.

Busche (1989) indicates how user modelling can be integrated into the analysis phase of the KADS life cycle. She suggests that the requirements of the user, both for functionality and interface requirements, should take place in the external stream. The different distributions of labour can be identified in the third stream for system modality with any necessary user task analysis in the internal stream.

KADS has been described in some detail. This methodology has the advantages that is does not get prematurely physical or too dependent on experimenting in order to produce the correct product. Prototyping is still seen as important but not the over-riding development tool. Finally, there is the emphasis on finding out about the user and actually modelling the user-system co-operation.

KADS has become widely accepted with Wetter (1992, p.VI) stating, in the preface to the proceedings of a conference on current developments in knowledge acquisition, that KADS currently is the most influential modelling approach. Georges (1992) also mentions the move by the commercial world especially in Europe towards sound comprehensive approaches away from *ad hoc* methods that offer fast results in the short run but break down eventually due to lack of generality and reusability.

In the light of this success, the stated aim of the KADS-II project is to develop a comprehensive method for knowledge-based systems development that will become the *de facto* standard for the European IT industry (Porter, 1992; Schreiber *et al.*, 1993; Wielinga *et al.*, 1993). Whereas models are viewed in KADS-I as intermediate results in the development process, in KADS-II they are seen as evolving and persistent products that enable them to be realised computationally. There are now six models: the organisational model, the task model, the agent model, the communication model, the expertise model and the design model. In effect, the conceptual model has been decomposed into its composite parts and the model of cooperation separated into the communication and agent models. The agent model allows for the explicit modelling of agents whilst the communication model describes the interaction between them. Building on the original KADS project, the intention is to develop a conceptual modelling language that allows the structure and the expected contents of a conceptual model to be defined. This approach whilst facilitating implementation seems to run counter to the principle of having an analysis phase that does not take into account computerisation. The move to design seems a little premature.

4.2.3 An alternative to KADS

There have been other approaches that allow a systematic approach to analysis. The developers of Keats, the Knowledge Engineer's Assistant, provide a knowledge engineering methodology based on the iterative refinement of qualitatively and teleologically different models (Motta *et al.*, 1988). The authors specifically disclaim any intention of providing a life cycle model for managing the development of knowledge-based projects. Their model-based approach aims to describe the nature of the knowledge-transformation activities which the knowledge engineer has to perform when building knowledge-based systems. Their broadly described activities include problem conceptualisation, knowledge encoding and debugging. Conceptualisation is seen as producing an abstract, machine-independent representation of the problem. The resultant conceptual model can be seen as a representation of the expert's knowledge and may have to be heavily refined at the knowledge encoding stage. Motta *et al.* (1988) note the similarity between their work and that of the KADS team. They comment, though, that in their approach the conceptual model is kept apart from the architecture of the end system whereas in KADS the interpretation models provide the architecture of the prospective system. This allows the Keats conceptual model to be driven purely by psychological considerations. Whilst this clean separation of knowledge and program is seen to be advantageous, the authors note that the process of developing a system can be very time-consuming when the structure of the conceptual model is very different from the final architecture of the system. In the Keats manifesto, Motta *et al.* (1988) stress their interest in understanding the nature of the knowledge engineering process rather than in its technological, organisational and environmental aspects.

4.2.4 User-centred approaches

Whilst the developers of Keats chose not to deal with organisational and end-user issues, there are those who concentrated their attention on these aspects. POMESS, A People Orientated Methodology for Expert System Specification, was designed to combat the seat-of-the-pants engineering based on rapid prototyping (Diaper, 1987; Diaper, 1988). Diaper wishes to direct attention from the technology to the organisation and users. An organisational model can identify both the tasks suitable for organisation and the people who are involved, the domain expert(s), direct potential users of a system and the indirect users. The task analysis phase uses a method known as Task Analysis for Knowledge Description (TAKD) which starts with a detailed observation of a task. Decisions about the type of interface necessary for a simulation can be made once the TAKD process has been completed. During the knowledge harvesting phase, the Wizard of Oz technique is used to simulate consultations between a potential user and the system, with the domain expert assisted by a typist mimicking the system. A prototype system is developed and, once completed, makes the expert redundant. The dialogues can all be kept, analysed and the interface designed from them. In order to make the simulation process even more effective, Diaper suggests that the user is initially provided with the most powerful interface possible and that the dialogue data can be used to select the most desired features. Once the prototype and the interface have been developed, the working system is evaluated. The underlying TAKD approach has been criticised by Benyon (1992) for embodying current practices in future systems. Moreover, whilst user needs are supposed to emerge from the Wizard of Oz process, this is an *ad hoc* way of eliciting them. Other knowledge engineers may not be able to use this approach successfully. Overall, though, Diaper displays a welcome enthusiasm for meeting user needs.

A related approach, but one not formalised specifically in a knowledge-based systems life cycle, is suggested by Johnson (1989). When developing any piece of software, it is seen as essential to model the different types of knowledge required by users to perform a task (Task Knowledge Structures). The Adept project, Advanced Design Environment for Prototyping with Tasks, is based on this approach (Markopoulos *et al.*, 1992). The failure of the rapid prototyping paradigm to incorporate any theory or design principles is noted by these authors. Mistakes can still be repeated in subsequent iterations of the prototyping and evaluation cycles. Instead, an approach is advocated that provides a framework for supporting task-based interface design. Prototyping is used in a more disciplinary fashion since models of the user and the task guide the design process. This work is significant for its use of task and user models and its criticism of rapid prototyping as a methodology for designing interfaces. Again user interests are at the forefront.

Like Diaper, Basden (1989) sees existing methodologies as technology centred. This makes the whole process obscure to the client, the deliverable often does not meet the client's real needs and the milestones of the project are usually incomprehensible. To avoid this technological bias, he proposes a client-centred methodology whose stages are usually deliverables rather than activities. These stages have names meant to be self-explanatory: start, skeleton system, demonstration system, working system, usable system, saleable system and system embedded in use. Except for the early stages, there is an indefinite number of cycles involving knowledge acquisition, knowledge representation and validation until all relevant issues have been resolved. Whilst the working system satisfies the experts, the usable system is meant to satisfy the end-users. Although described in the methodology as a separate stage, Basden expects it in practice to overlap with stage four. It is pulled out as a step in its own right since usability is so important. Here the cycles of knowledge acquisition, implementation and validation concern knowledge from the end-user and not the expert. This allows suitable help and explanation facilities to be developed. Basden, like the developers of KADS, recognises the importance of the organisation and the ultimate users of the system. Unfortunately, as with KADS, there is little detailed information about integrating user concerns within the methodology. For example, it is not clear how the workable system can be developed in parallel with the usable system. How this goal can be achieved is not explained in any detail. Moreover, the client-centred approach is heavily dependent on prototyping with all its attendant problems. Without the usual framework of staged development provided by a life cycle (analysis, design, implementation, etc), it can be hard for a knowledge engineer to decide what to do when.

A human factors perspective on expert system design is also taken by Howey, Wilson and Hannigan (1989). It is interesting to note that they do not have an analysis phase *per se* although one of their main goals is to determine system functionality according to the needs of the users. Within their design process they have the three stages of specification, development and evaluation. Whilst prototyping does not appear as a separate phase, there is an underlying assumption that a prototype will be built for users to test. The approach is iterative and the specification of the system can be modified as a result of the evaluation process.

An overview of the human factors input to knowledge-based design is shown in Figure 4.2. There are two separate streams, one that deals with users and another that concentrates on expertise. This allows the knowledge engineer to identify the functions and interface characteristics that are desirable in the system from the users' perspective. The knowledge-based system is then presumably prototyped and evaluated. This approach has the advantages of meeting user needs and providing a sound foundation for prototyping. Unfortunately, it fails to distinguish between analysis and design which are seen, essentially, as one process. The iterative nature of the methodology means that the functions of the system may be determined ultimately by prototyping and not by a rigorous analysis phase. Even though there is a commitment to integrating user factors into a knowledge-based systems life cycle, the required outputs are not specified in sufficient detail to help the knowledge engineer achieve this goal. For example, exactly what information should be obtained about users? There is also the underlying assumption that the elicitation of expertise is straightforward. Finally, the views of the organisation seem to be overlooked.

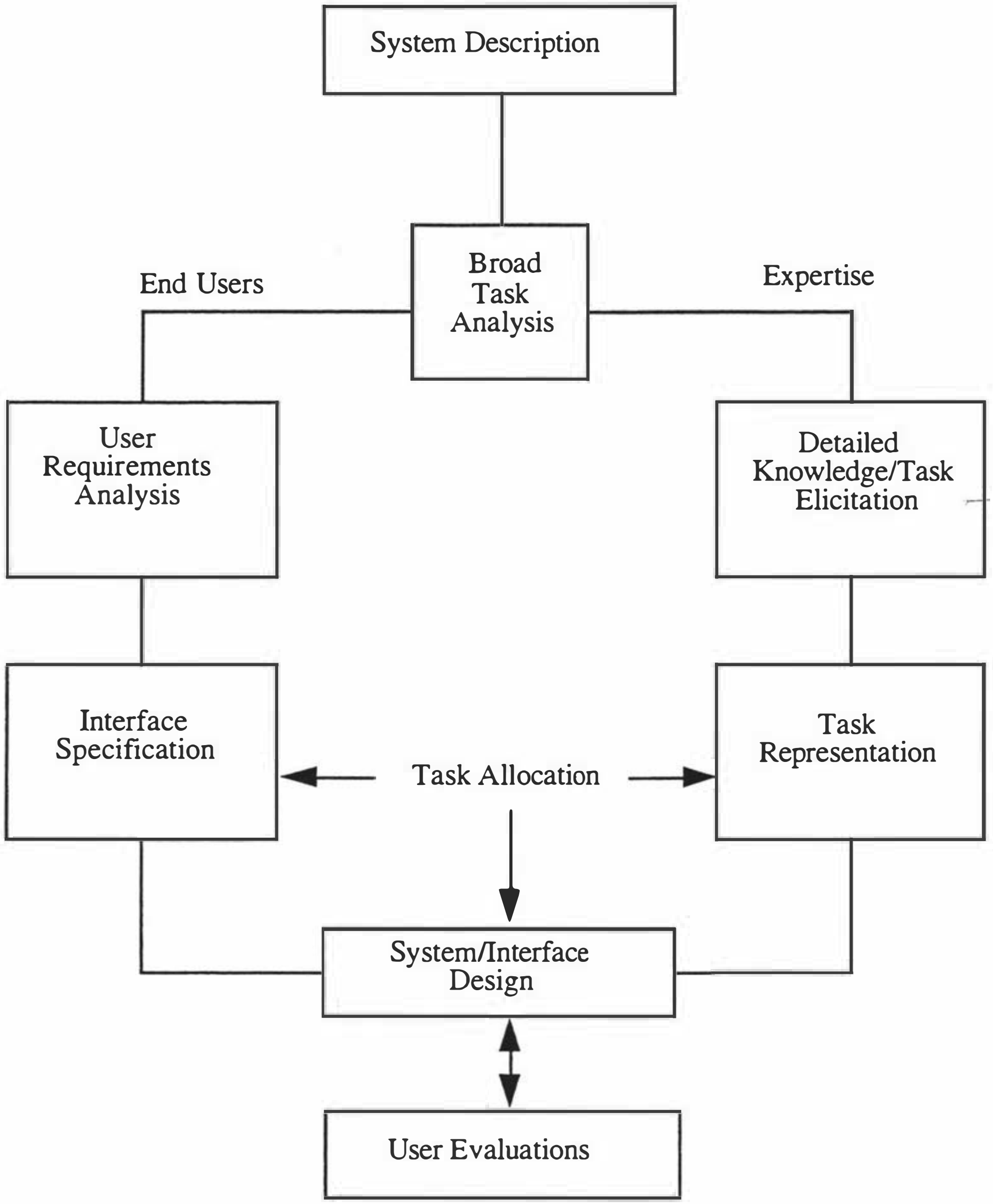


Figure 4.2 Human factors inputs to knowledge-based design (Howey *et al.*, 1989)

Kemp and Kemp (1991b) also stress the importance of obtaining user cooperation for a project. They illustrate diagrammatically the interplay required between the knowledge engineer, the experts, the users and the organisation to produce a workable specification. The double-headed arrows in Figure 4.3 indicate that analysis is a two way process with the knowledge engineer constantly refining and clarifying his/her understanding of each area. Much iteration is needed to ensure that all needs are met.

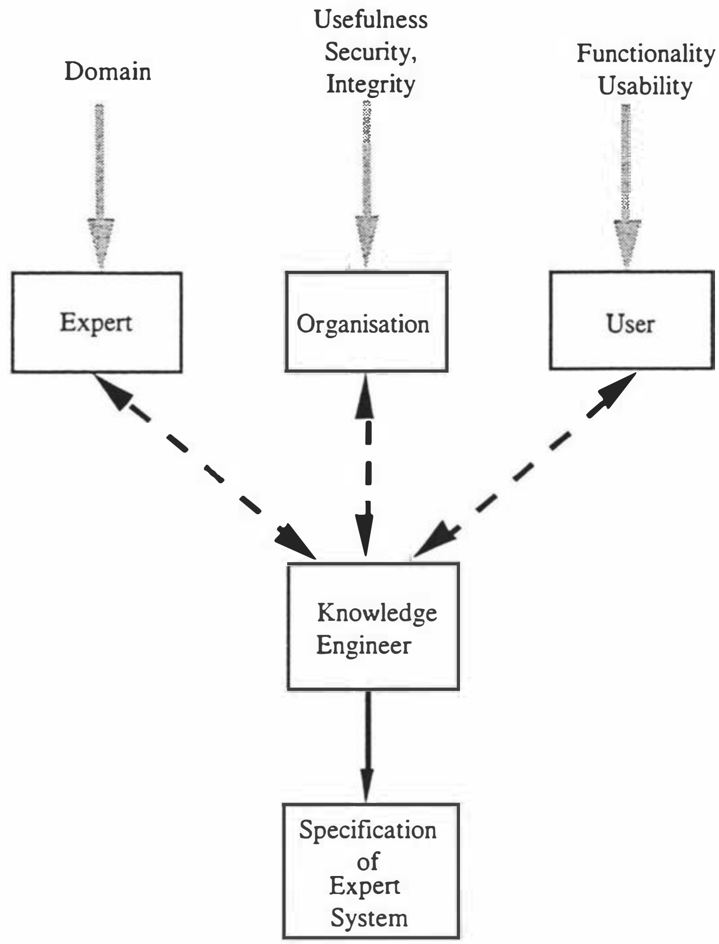


Figure 4.3 Components of analysis (Kemp and Kemp, 1991b)

4.3 Conclusion

The foregoing analysis of expert systems life cycles illustrates the reliance on prototyping within the knowledge-based systems community. It has been argued that this is a weakness rather than a strength. In many cases there seems to be no clear distinction between what Pressman terms the definition and development phases of a life cycle: see, for example, Harmon and King (1985) and Turban (1992). At the heart of developing a successful expert system is the elicitation, analysis and modelling of the required knowledge. Substituting prototyping for a thorough understanding of the problem can cause many problems, meeting needs in the short but not the long term, for instance. The mission of the KADS projects is to avoid the pitfalls of rapid prototyping and develop a comprehensive, commercially viable methodology. Prototyping is seen as tool with limited value but not the centrepiece of a life cycle. Over the years, researchers on the KADS project also realised the importance of organisational and user issues. The interface, for example, should not just be tacked on once a suitable prototype had been developed but considered as soon as possible. Hickman *et al.* (1989) though, comments on the problem of linking all relevant strands within the analysis phase. Even if there are some practical problems with the KADS approach, certain invaluable guidelines can be distinguished. Firstly, the initial representation of knowledge should be free from implementation considerations; intermediate models can be used for this purpose. Definition of the system, consequently, is clearly separated from design and coding issues. Secondly, extensive knowledge acquisition should take place before specifying a system. Thirdly, this should be a top down process driven by the selection of the appropriate interpretation model. Fourthly, human factors are important and cannot be left until the implementation stage. As a framework for large scale system development KADS undoubtedly has much to offer.

Part 2

FOCUS and the Enrolment Case Study

Chapter 5

FOCUS

In this chapter, the FOCUS framework, for use in the analysis phase of a knowledge-based system life cycle, is introduced. Its principal deliverables and the reasons for developing FOCUS are discussed. The five stages: *Problem specification*, *Preliminary analysis*, *User analysis*, *Functional specification* and *Detailed analysis* are fully described. The inputs, activities and outputs of each phase are all detailed. Finally, the way that the knowledge engineer can use the major outputs of FOCUS for interface design is outlined.

5.1 Introduction

An important concern in the design of interactive systems is meeting the users' objectives. As already noted, if user requirements are ignored then programs may be seldom or never used. This situation can be improved if relevant interaction issues are considered in the analysis phase of an expert system. There are two separate but related issues. Firstly, users expect that the system will deliver the functionality they require. Secondly, good communication with the system is essential so that users can run the program, understand its problem solving capabilities and ask questions about the underlying domain. Tailoring a system to satisfy users can be a difficult task when there are differences in computer background and/or domain knowledge. To assist with this process, FOCUS can be used in the analysis phase of an expert system life cycle (Kemp, 1993). This approach ensures that the required "**F**uncti**O**ns and **C**ommunication facilities for **US**ers" are elicited. Relevant user modelling as well as domain analysis is necessary.

FOCUS is based on the model building approach advocated by the KADS team (section 4.2.2). Wielinga *et al.* (1991) specify the series of models that should be built (Figure 5.1). The knowledge engineer starts by developing the organisation model which considers the problems and objectives of management. Then he or she can move through to the application and task models that define the functions of the system and the way(s) in which the organisation's goals can be achieved, respectively. The model of expertise and model of cooperation can then be completed. The model of expertise has four layers which specify the problem solving expertise needed to solve a particular problem whilst the model of cooperation indicates which agent, the user or the system, carries out a task. These last two models are fused into the conceptual model. Only at the design stage are implementation details considered and the design model adapted from the conceptual model in the light of these.

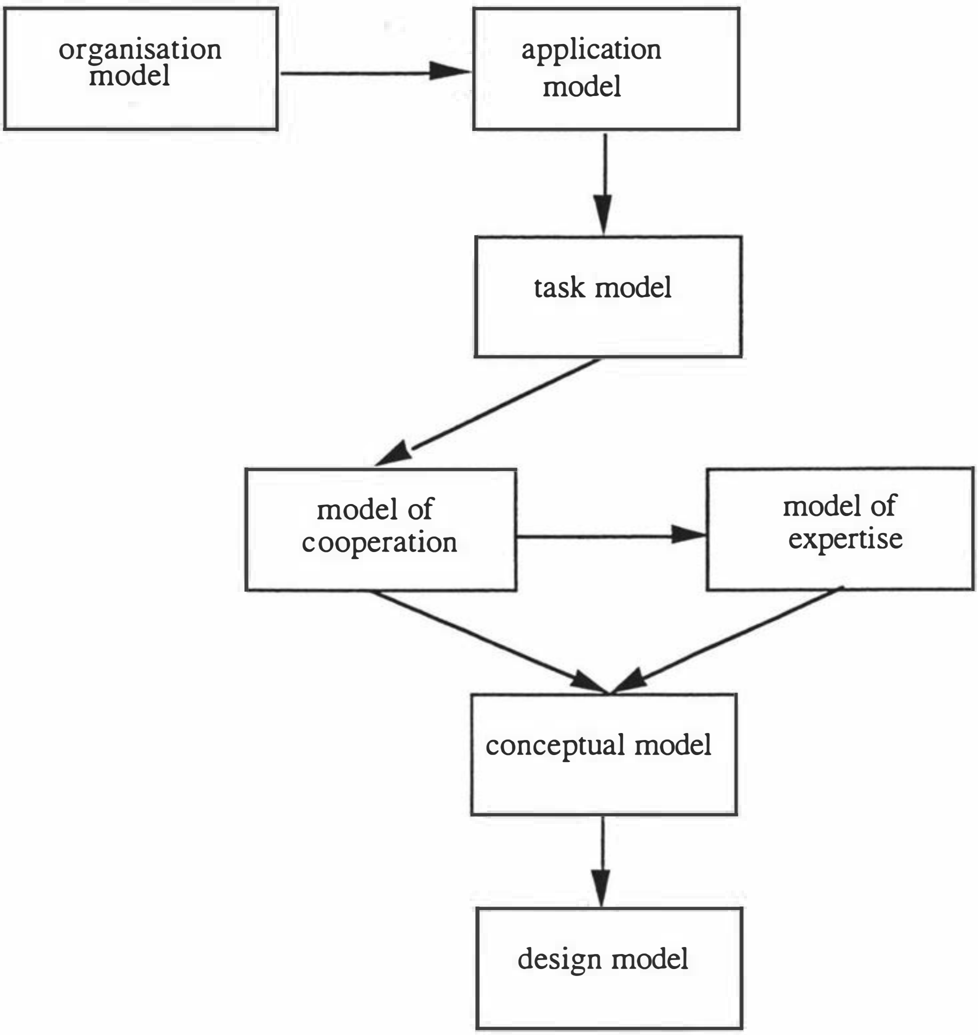


Figure 5.1 Intermediate models (Wielinga *et al.*, 1991)

All the models except that for design are developed during the analysis phase of a knowledge-based system life cycle. Hickman *et al.* (1989) split the analysis phase into three strands: the external stream concerned with the organisation's requirements; the internal stream, dedicated to modelling the expertise, and the modality stream which is concerned essentially with user-system cooperation. Since the role that user modelling should play in the analysis phase of the

KADS life cycle is not clear, Busche (1989) suggests that the functional and interface requirements of users should be identified in the external stream. On the other hand she states that communication issues should be handled in the modality stream.

There are many fruitful ideas in these works but overall there is fragmentation and no clearly defined way of ensuring that all user concerns are met. Even if the appropriate user requirements can be ascertained there are still two crucial questions to answer. Firstly, what information is needed from the analysis phase to assist with the design of the interface? Secondly, how can the knowledge engineer ensure that the relevant expertise for explanation is elicited? It is argued that there should not be three different strands of analysis but that they should be integrated to ensure the generation of the required outputs.

Consequently, FOCUS has been developed to ensure that during the analysis phase organisation, user and domain modelling are interlinked (Figure 5.2). Three deliverables need to be produced: the organisation, task and conceptual models.

This simplified approach follows Porter (1992) who suggests omitting the application model and refers to the model of communication instead of the model of cooperation. The organisation model contains all the information about a knowledge-based problem from an institution's perspective. If a decision is made to build a knowledge-based system, the task model shows the functions required to achieve the organisation's objectives. The conceptual model is itself composed of three parts, that is the model of expertise, the model of communication and user requirements. Whilst the model of expertise specifies the system's problem solving knowledge, the model of communication shows what subtasks are performed by the user and which by the computer. Finally, the user requirements model describes both the functionality desired by the users and how their interface needs can be met.

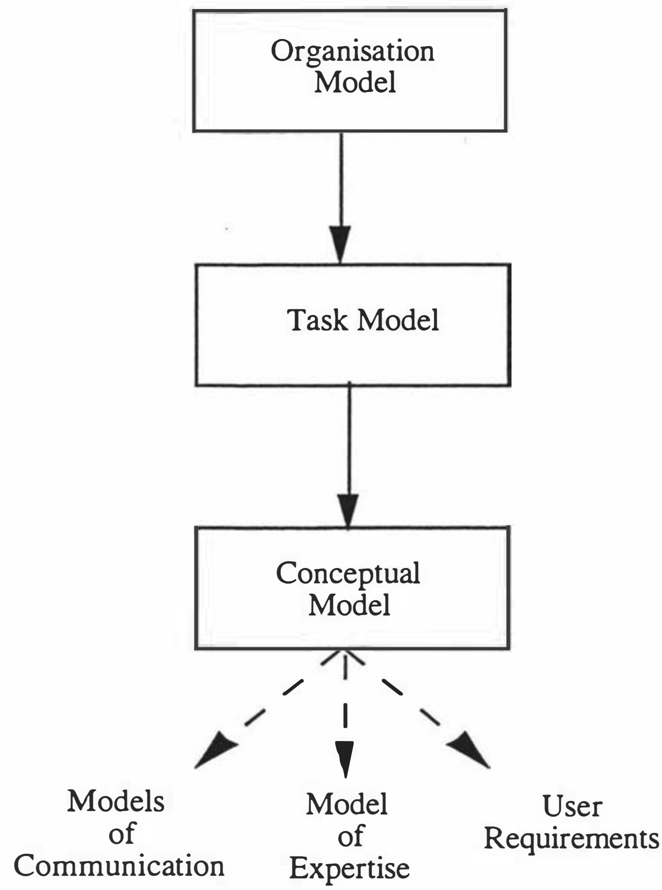
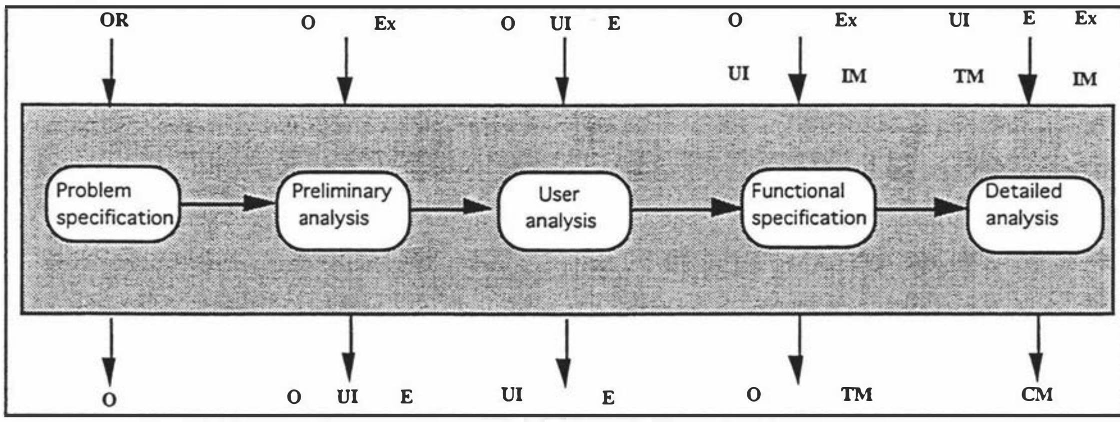


Figure 5.2 FOCUS deliverables

5.2 The FOCUS framework

FOCUS (see Figure 5.3) identifies the main stages of the analysis phase that are necessary to produce the conceptual model: *Problem specification, Preliminary analysis, User analysis, Functional specification* and *Detailed analysis*. The major inputs and outputs which connect the stages are also shown. It can be seen that, starting from the organisational requirements, it is possible via a series of deliverables (such as the organisation model), to produce the conceptual model. Other important flows include various sources of information such as documents and the knowledge of experts (Ex), user issues (UI), the model of expertise (E) and the task model (TM). Models go through many revisions. Any flow that is both input to and output from a process is altered during that stage (for example, user issues in the process *User analysis*). As usual with all life cycle activities, some iteration may be needed. If an organisational requirement did not emerge until the end of the *Preliminary analysis* phase, it would be necessary to return to *Problem specification* and discuss this issue with management. The existing organisation model would have to be extended and

Preliminary analysis entered again. Nonetheless, activities have been structured to minimise iteration as much as possible. Take the first three activities, one focuses on the organisation, one on the analysis of expertise and one on the user.



Key :

OR	Organisational requirements	E	Model of Expertise
O	Organisation model	IM	Interpretation Models
Ex	Expertise	TM	Task Model
UI	User Issues	CM	Conceptual Model

Figure 5.3 The FOCUS framework

The main goals of FOCUS are:

- to integrate organisational, user and expertise issues. This should be achieved without the fragmentation implicit in the KADS approach.

- to enable the knowledge engineer to identify the principal activities of the analysis phase and the way they relate to each other through inputs and outputs. The components of the conceptual model can be cross-checked to ensure as far as possible that no important requirement has been omitted.

- to take account of the iterative nature of knowledge-based development. Analysis takes place over a period of time and models are extended and refined accordingly.

- to show clearly the move over time from the organisational to the problem solving perspective;

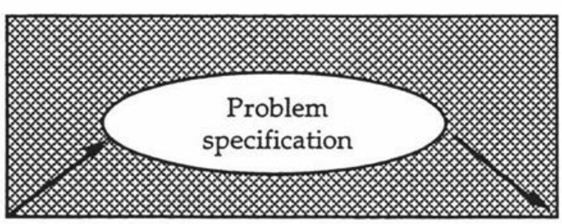
- to allow important human-computer interaction issues to be dealt with since FOCUS assists with the identification of user tasks and specifies how the user can communicate effectively with the intended system;

- to lead to the design phase for both knowledge base and interface design. The model of expertise can be translated into any suitable form of knowledge representation whilst the model of communication reveals when interaction between the user and computer occurs.

FOCUS has many possible advantages for the knowledge engineer developing an interactive system. Firstly, it takes account of the diversity of users, that is users with differing domain and interface requirements. Secondly, the user is viewed not in isolation but in the context of the organisation to make sure that this perspective, too, is incorporated. Thirdly, much information is generated that can be used for project management purposes. Decisions, for example, about whether to continue with the project can be based on the information held in the organisation model.

To show how the objectives of FOCUS can be met, a detailed account of the major inputs, activities and outputs of each stage follows. How this approach can be applied is demonstrated in chapters 6 and 7.

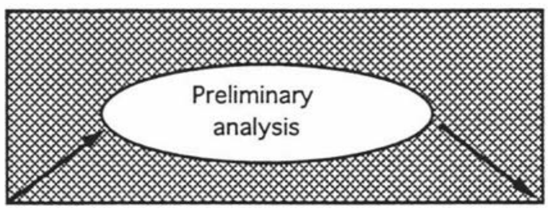
5.2.1 Problem specification



Input	**Activity**	**Output**
Organisational requirements	Discuss: problem area, suitability of kbs approach	*Organisation model* : problems of management objectives of management sources of expertise evaluation of sources identification of possible users

The *Problem specification* stage begins when a problem arises within an organisation and a solution is proposed which involves knowledge-based technology. A knowledge engineer is selected to see if a solution can be found. Discussions with the relevant members of staff take place to elicit their problems and the reasons for wanting a knowledge-based system approach. If it becomes apparent that the technology is not suitable then the matter would probably end here. Otherwise the various sources of expertise such as relevant documents and available experts are identified. As they cannot all be consulted at once, the knowledge engineer has to decide what sources to consider first for the *Preliminary analysis.* The order will depend on advice given by the staff members concerned and the experience of the knowledge engineer. The different groups that may use the system also have to be identified. All these outputs are described in the organisation model.

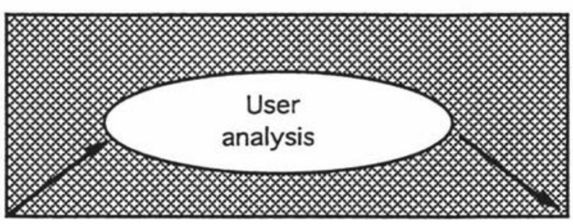
5.2.2 Preliminary analysis



Input	Activity	Output
Organisation model	Elicit knowledge	*Organisation model*
Selected sources of expertise		background to problem
		current procedures
		alternative courses of action
		Model of expertise
		lexicon
		concepts
		attributes
		relationships
		User issues
		problem areas
		existing situation

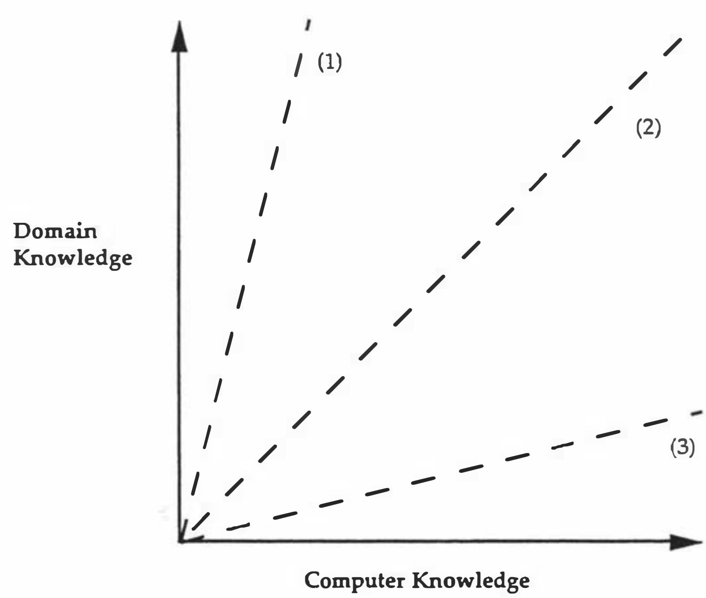
The over-riding goal of this stage is to see whether one or more possible solutions can be identified. Knowledge about the domain and the organisation has to be acquired. The knowledge elicitation process can be carried out using any appropriate techniques. The knowledge engineer then sifts through the facts obtained from documents and experts to build up an initial picture of what is happening. Verification of the knowledge obtained is important at this and any other time that knowledge elicitation takes place. The elicited expertise should be recorded, a lexicon of important terms built up and the domain layer of the model of expertise started. Once an initial picture of the situation has been built up, a decision can be taken about whether to proceed. As Wielinga *et al.* (1991) point out, analysis can reveal problems that the organisation can solve without resort to computerisation. Investigating a problem can, of itself, lead to other ways of improving the situation and the project may come to an end. Approval to go ahead, though, can only be tentative as there may be other pitfalls along the way. If provisional approval is given, one or more courses of action can be specified and the probable users of the system identified. In order to plan for the next stage, *User analysis*, the knowledge engineer should find out about the current situation of the users in the organisation and identify possible problem areas.

5.2.3 User analysis



Input	**Activity**	**Output**
Organisation model	Elicit user requirements	*User issues*
Model of expertise	Extend model of expertise	user requirements
User issues		domain background
		computing background
		Model of expertise
		concepts
		attributes
		relationships

To a great extent, the organisation's needs have been dominant up to this point. During *User analysis* the knowledge engineer attends to the concerns of users and their view of the situation within the organisation. The user perspective may be quite different from the administration's but no less valuable. Information from this source not only helps the knowledge engineer to fill out the model of expertise but also indicates what functions (if any) users require and whether they would support the introduction of a computerised system. Both the computing skills of the users and their domain background may also need to be ascertained. The intended users of the system might have quite diverse backgrounds, some knowing a great deal about computers and little about the domain and *vice versa*. Others again may be experienced with both (Figure 5.4). The needs of these groups will probably be quite different and would have to be discussed at the next stage. Information is gathered during *User analysis* so that a realistic appraisal can be made of user needs before taking irrevocable decisions about the knowledge-based system. The success of this undertaking depends upon the careful preparation by the knowledge engineer of any required questionnaires, interviews, etc. Without user support the viability of the project has to be seriously questioned.

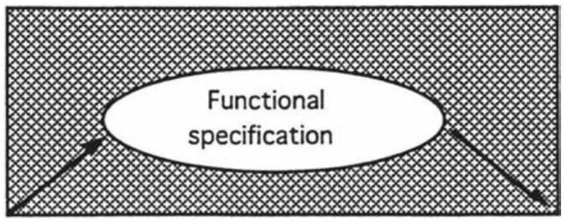


Key

1. Novice computer user but domain expert.
2. Similar level of experience with both domain and computer.
3. Domain novice but expert computer user.

Figure 5.4 Domain knowledge versus computing background

5.2.4 Functional specification



Input	Activity	Output
Organisation model	Determine required functions	*Organisation model* :
Model of expertise	Match IM model with problem	intended users
User issues		functions
Interpretation models		impact on environment
		problems to be solved
		interface requirements
		criteria for evaluation
		Task model

The final decision about the future of the project is made during the *Functional specification* stage. It could still be necessary to rule out knowledge-based technology if, for instance, the hostility of the intended users is seen as an insurmountable barrier. All the relevant information about the organisation, the users and the domain has to be carefully considered before deciding whether to go on. Ignoring any one of these can doom a project to failure. If a decision to go ahead is made, then the intended users of the system and its main functions can be specified. A system may have to meet the needs of two or more distinct groups, each with its own domain profile: see Busche (1989) for an example of a diagnostic system developed for both novice and experienced technicians. In these circumstances, the requirements for each group should be specified. The problems the system will solve and the anticipated impact on the organisation should also be described.

Another important decision is whether to cater for differences in user computing background by allowing for more than one physical interface. Once this decision is made, the criteria for evaluating the usability of the knowledge base should be specified. Focusing on this issue from the outset should allow the subsequent development of any appropriate interfaces. An enormous number of checks can be made but those most relevant to the application should be highlighted.

The problem solving capabilities of the system are shown in a task model, a high level description of the activities required to accomplish the objectives of the system. Figure 5.5 shows the task model for a knowledge-based system that would diagnose and rectify faults in an audio system. The real life task is decomposed into the necessary sub-tasks. Although it is only skeletal at this point, this model is of central importance. Firstly, it forces the knowledge engineer to identify the main problem solving activities involved (for example, diagnosis and monitoring) and choose the relevant template from the KADS library of interpretation models to drive the knowledge elicitation and modelling process. Secondly, as more knowledge is obtained during *Detailed analysis*, the task model is extended and, when completed, becomes the basis for the model(s) of communication.

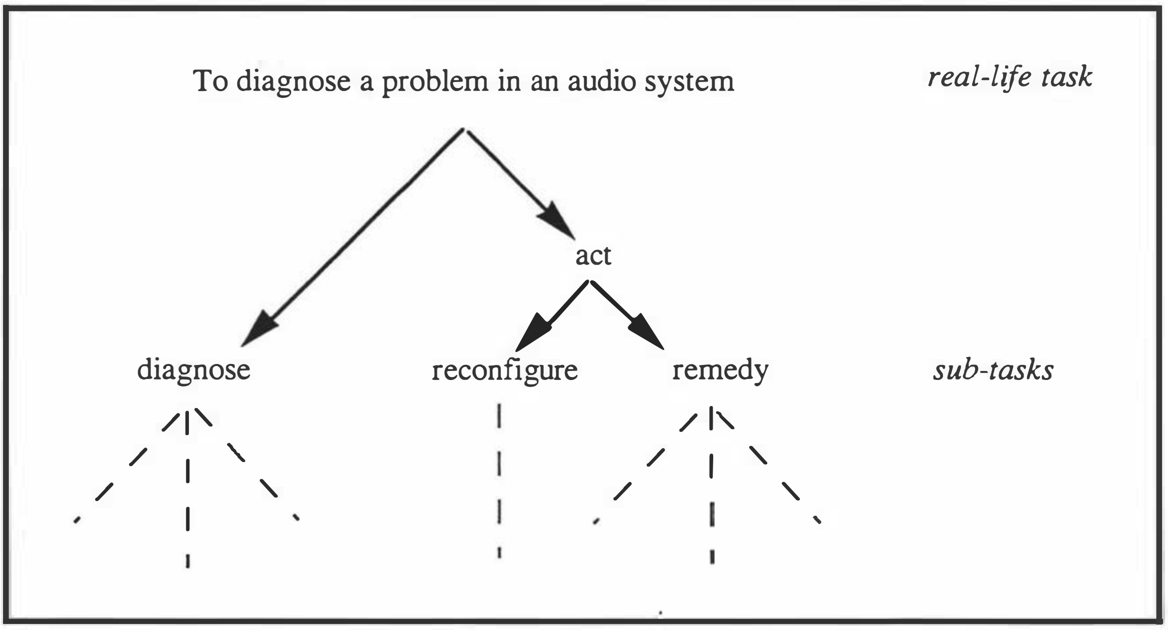
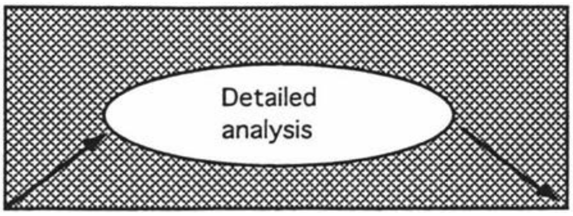


Figure 5.5 Task model (Wielinga et al., 1991)

5.2.5 Detailed analysis



Input	**Activity**	**Output**
Expertise	Elicit knowledge	*Model of expertise*
Task model	Analyse knowledge	domain layer
Model of expertise	Interpret knowledge	inference layer
User requirements	Detailed task analysis	task layer
Interpretation models	Model users	
	Model communication	*Models of communication*
	Model knowledge	user-system cooperation
	Validate knowledge	
		User requirements
		user interface requirements
		user models
		explanation needs

During the *Detailed analysis* stage, further elicitation of both domain knowledge and user needs takes place. As this proceeds, the task model is extended whilst the model of expertise and user requirements are developed incrementally. A model of communication has to be built for each user group to show which party carries out each task. The various components of the conceptual model and the inter-relationships between them are shown in Figure 5.6. The knowledge engineer should check that the models accurately represent the expertise that has been elicited as well as meeting the organisation's requirements.

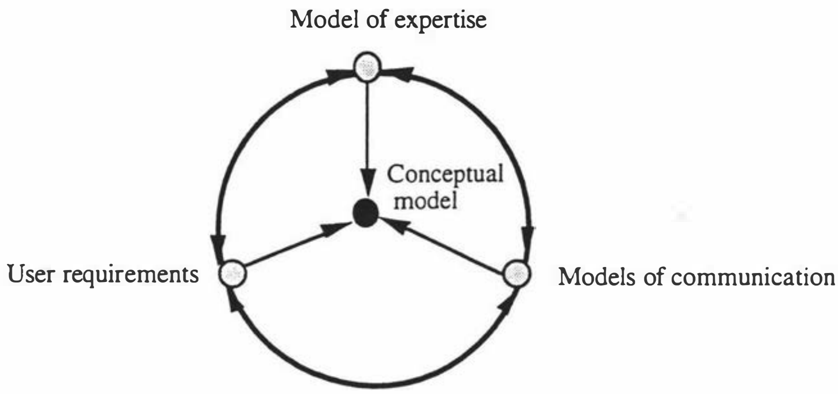


Figure 5.6 Components of the conceptual model

The model of expertise

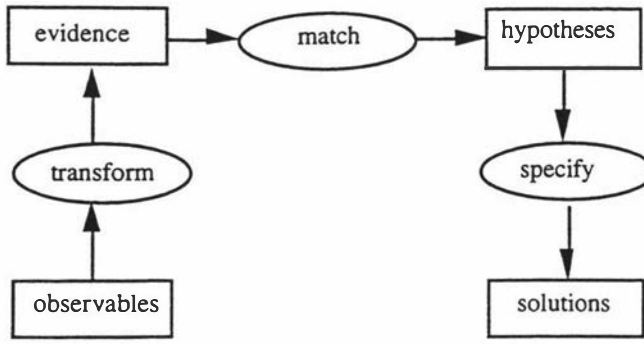
Once the important organisational decisions have been taken, the remainder of the relevant domain knowledge has to be obtained from experts, documents, etc. In accordance with the principle of openness, the knowledge engineer may use any appropriate knowledge elicitation techniques including prototyping to ensure that all relevant domain information is collected from experts. This process is guided by the use of interpretation models from the KADS library. For example, if a knowledge-based system for diagnosing faults in equipment is being developed, the relevant template can help the knowledge engineer to decide what information needs to be collected as well as how to structure the problem solving process. As expertise is obtained, it is incorporated into the domain layer whilst the inference and task layers are built up. A simplified example of these levels is shown in Figures 5.7, 5.8 and 5.9 (Hickman *et al.*, 1989).

Concepts : fever, spots, stomach ache, headache, mouth pain, food intake, gum disease, etc.

Relations

alcohol consumption can cause liver sclerosis
mouth pain can cause unchewed food
unchewed food can cause indigestion
pneumococcea is a bacterial infection

Figure 5.7 Domain layer for medical diagnosis (Hickman *et al.*, 1989)



Observable: temperature of 41 degrees Celsius
evidence: high fever
hypothesis: bacterial infection
solutions: pneumonia caused by pneumococcea glomeruleonephris

Figure 5.8 Inference layer for medical diagnosis (Hickman *et al.*, 1989)

Solution driven

- specify (solutions)
- match (hypotheses)
- transform (evidence)
- obtain_data

Figure 5.9 Task layer for medical diagnosis (Hickman *et al.*, 1989)

If users are divided into two or more distinct groups, each with their own domain requirements, then each will need their own view of the system. Whilst each user view is actually defined in the model of communication, the model of expertise will have to support multiple ways of using the knowledge base. The knowledge engineer has to ensure that the domain layer includes all relevant static knowledge. For each model of communication there also needs to be an equivalent inference and task diagram. These will probably be variations on the template selected from the library of interpretation models. Only one program will ultimately be implemented for a given project and it will bring together the requirements of each group as specified in their relevant inference and task layers.

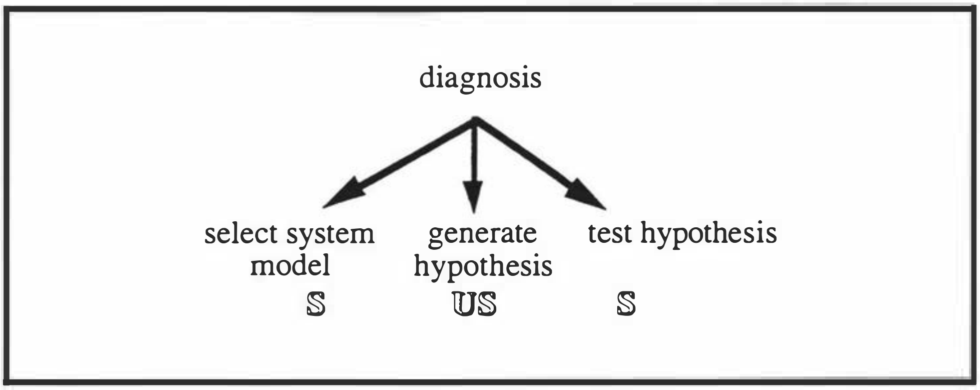
The domain layer does not only contain non-procedural knowledge but also incorporates any model of the user needed for problem solving purposes. Furthermore, once the explanation needs of the users have been ascertained then any information needed for answering questions (support knowledge) has to be added to the domain layer.

Model of communication

A model of communication acts as a bridge defining how the users can interact with a specified system. It can be viewed as an extension of the task model which shows the responsibilities of each agent. The task model, though, has to be completed before the model of communication can be started. A separate model of communication is then produced for each version of the system. Decisions about who carries out which activities are made in view of the associated user task model.

A model of communication is developed in the following manner. On every branch of the task model, the knowledge engineer indicates whether the activity should be undertaken by the user or the computer (see Figure 5.10). Any activity that is jointly undertaken is further decomposed until one party has the responsibility for each action. Problems may arise allocating tasks that indicate the need for further knowledge elicitation. The additional domain information has to be recorded in the model of expertise. The task model and subsequently the model of communication may have to be modified as a result of this process. The preceding discussion assumes that only one model of communication is required. When more than one has to be constructed, the relevant subset of the task model acts as the basis for each model of communication.

Once the task allocation has been completed, the knowledge engineer can see quickly where data input and output occur. A check should be made against the user task model to see that the sequence of activities is logical. It is also possible to identify from an examination of the model of communication where users may wish to ask questions. The selection of appropriate explanation features can then be finalised.



Key
𝕌𝕊 - user and system
𝕊 - system
𝕌 - user

Figure 5.10 Model of communication

User requirements

Once the functions of the system have been specified, further analysis determines which user models should be developed. The full set of models that may be needed includes the user task model, the model of user knowledge, the model of user characteristics and the problem solving model of the user. Whilst these have already been discussed previously (sections 2.2.2 and 4.2.2), a brief definition of each is given here for completeness. A user task model describes how users currently carry out a task whilst the model of user knowledge records their level of domain comprehension. The model of user characteristics describes the users' familiarity with computers and experience with interaction styles. Finally, the problem solving model of a user is generally only needed in advisory system and describes the aspects needed for a knowledge base to come to a conclusion. Why does the knowledge engineer have to develop these models? Let us take first the simple case where there is a homogeneous group of users and only one version of the knowledge-based system is being developed.

It is essential to make sure that the problem solving strategy of the system does not differ markedly from the way that users themselves would solve the problem or expect the problem to be solved. Arbitrary sequences of questions may confuse the user. Since cognitive consistency is vital, a stereotype task model should be built of its intended users. Techniques such as TAKD (Diaper, 1989), KAT (Johnson, 1992) and prototyping may also prove useful at this point. Until the knowledge engineer has some idea of how users picture the domain, the model of communication cannot be started.

The knowledge-based system should not only be easy to use but the language of the system should reflect the natural idiom of the user. Users have to enter data and understand conclusions and recommendations as well as asking questions and assimilating explanations. Consequently, a stereotype model of users' knowledge may be needed for dialogue generation.

The explanation needs of users must also be determined. There are two quite distinct issues. Firstly, the appropriate explanation features have to be selected. As Wognum and Mars (1991) point out, users appear to be happy with a subset of facilities that meet their user requirements. The knowledge engineer has to ensure that the appropriate subset is chosen. The model of communication and the model of user knowledge can both be taken into account at this point. Once suitable explanation functions have been proposed, any relevant support knowledge should be identified so that it can be incorporated into the domain layer. Explanations, it must be noted, have to be phrased in accordance with the model of user knowledge.

A knowledge-based system may need information about the user in advisory or teaching systems. For example, an advisory system for determining the best superannuation strategy for an individual must know a great deal about that user and his or her future plans. A description of the kinds of user information needed by the expert system should be included in the problem solving model of the user. Such models are application-dependent since the information is that needed for solving specific problems.

Finally, it is also possible during *Detailed analysis* to choose the interface presentation style even though detailed screen design is done later. A stereotype model of user characteristics based on information collected in both this and the *User analysis* stage will help the knowledge engineer to choose whether to adopt one or more of the following styles: graphical, text, form fill, command line or natural language interface. Obviously, the task that is to be done also has some bearing on this decision. For example, if a knowledge base has been developed to teach people how to operate a complex piece of equipment, then a graphical representation of the interface is indicated.

The situation may arise where the intended users have very different domain and/or computing backgrounds. How does FOCUS cope with this? In an extreme case, a knowledge-based system can be built to cater for the needs of each user. Individual, executable models might be necessary. These would be embedded within the knowledge base and updated to ensure that they were responsive to user changes in computing or domain experience. On the other hand, where discrete groups can be identified, stereotype models would once again be required.

Where only the domain background is quite different, a set of stereotype models should be developed for users of each version of the knowledge-based system identified in the *Functional specification.* This set must include the user task model and a model of user knowledge with, if necessary, a problem-solving model of the user. These can be used to help the knowledge engineer select the required subset of explanation features for each group. Information can then be presented to users in the most appropriate form (graphics may be the preferred medium of communication for one group and text for another). Whilst there is only one model of user characteristics, this is no guarantee that there will only be one interface. In some circumstances, it becomes clear from the inspection of the user task model and model of knowledge for each group that information has to be presented to users in quite different ways.

What happens when the computing background of the users varies? In this context there are clearly two situations: when there is only one version of the system and when there are more. In the case where only one set of functions is being provided, a model of user characteristics still has to be developed for each group with its own distinct computing needs. The demands of the task may over-ride any other considerations and only one interface presentation style is appropriate. Nevertheless, the models of user characteristics are not wasted as decisions can be made about the extent of training and help facilities to be provided. If the computing background of users is the only determinant of the presentation style then the knowledge engineer can ensure that the functions are delivered to each group in the most suitable way. For instance, if some of the users are blind then a natural language voice input interface may be recommended for them whilst others would be required to use text.

Interface issues are more complicated when groups have dissimilar domain and computing backgrounds. Let us assume that two versions of the system are going to be developed and that there is a group of naive computer users and another group who are more skilled. All relevant models have to be developed. When this is done decisions can be made about the presentation style. Various options are open to the knowledge engineer at this point. Again the situation arises that there is in practice only one way to present information in a meaningful way (Figure 5.11). This is often the case with graphical interfaces which represent equipment in a diagrammatic form so that users can indicate the area where there appears to be a problem.

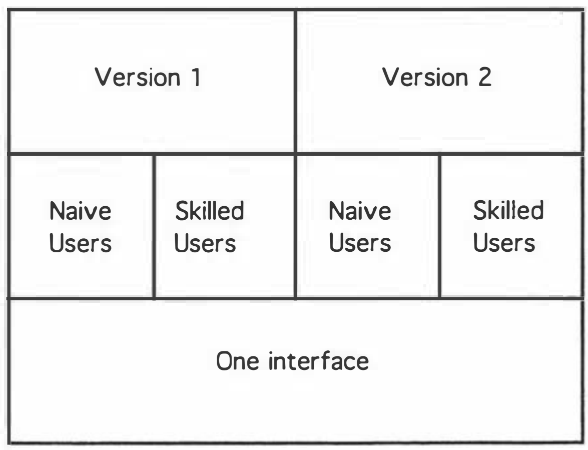


Figure 5.11 One interface presentation style

Alternatively, all naive users may be provided with one interface style and all skilled users with another (Figure 5.12). This is very flexible as it allows people to transfer from one interface style to another if their computing skills improve over time.

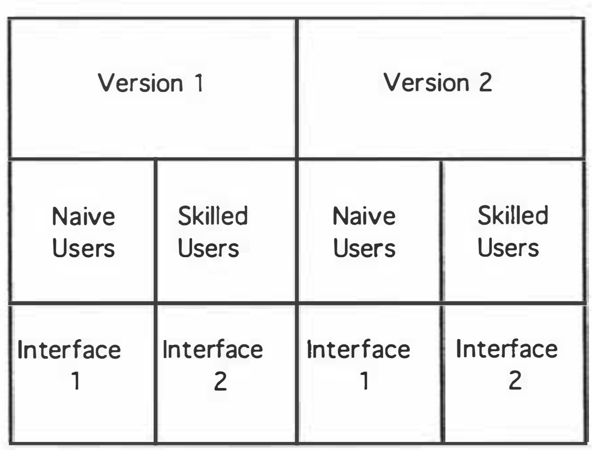


Figure 5.12 Two interface presentation styles

Finally, in the worst case, a decision may be made to tailor systems so that the functions required are presented in a way that suits both the domain and computing background (Figure 5.13). This may arise if a knowledge base is going to be used for both teaching and problem solving purposes (the APPLE system could have been used by both students and orchardists, for instance). Again, this has the advantage that with more computing experience, people can move from the naive to the skilled category.

Version 1		Version 2	
Naive Users	Skilled Users	Naive Users	Skilled Users
Interface 1	Interface 2	Interface 3	Interface 4

Figure 5.13 Four interface presentation styles

Discussion

The principal interactions among the three components of the conceptual model will be summarised here. For each model of communication, the knowledge engineer has to develop the associated inference and task layers of the model of expertise. The model of expertise and the model of communication are interdependent and changes in one have to be reflected in changes to the other. This can happen when allocation of tasks in the model of communication reveals the need for further knowledge elicitation. When the problem area is clarified both the model of expertise and the model of communication have to be altered. The task layer of the model of expertise should enable all the activities specified in the model of communication to be carried out. A check to ensure that this is the case should be made.

How are the models of expertise and user requirements linked? Firstly, once the explanation needs of the users have been identified, the domain layer can be examined to see whether it contains the required knowledge. If not, the relevant information can subsequently be incorporated into the domain layer.

Secondly, when modelling the problem solving aspect of the knowledge base, it becomes clear to the knowledge engineer whether a model of the user for problem solving purposes is wanted. This can then be included in the set of user models developed. Thirdly, the user task model influences what additional functions users will be offered. The model of expertise may have to be extended to cater for these. Finally, once the presentation style has been determined, domain knowledge may have to be collected in the appropriate form. For example, not only text descriptions but pictures may be needed.

Finally, the connections between the model of communication and user requirements must be considered. The model of communication cannot be started without taking account of the user task model. The knowledge engineer must review this to decide whether additional functions have to be provided for users. Moreover, the allocation of responsibilities (either to the user or the computer) also depends to some extent on the user task model. The final knowledge-based system should not act in a way that is quite foreign to the users. Once the model of communication is completed, it is possible to decide, in conjunction with the model of user knowledge, what explanation facilities should be offered. Consequently, this part of the user requirements cannot be finalised until quite late in the *Detailed analysis* phase.

The specification of the knowledge base also has to be verified. There are two ways to do this at this stage when there is no computerised system available. Firstly, a check should be made against the description of the application in the organisation model to ensure that no requirement has been overlooked. Secondly, the completeness and the consistency of the conceptual model must be checked. The dependencies between the three components assist this process. For instance, the model of communication and its corresponding task layer can be inspected to see that they incorporate the same functions. There is also some material, such as the model of a user for problem solving purposes, which appears in both the domain layer and the user requirements section. Obviously, whenever this happens the information should be identical.

5.3 Conclusion

Initially, a review of the life cycle literature indicated that KADS, a very popular methodology, could be used to determine interaction needs. Further experience proved this not to be the case since detailed guidelines for the knowledge engineer were lacking. Consequently, FOCUS was developed to supply the necessary framework. It recognises that the intended users of an expert system in an organisation may not all want the same problem-solving capabilities. The major user groups are identified and the functional requirements of each group specified. Communication issues can then be considered for each group. At the same time the analysis of the organisation's needs and elicitation of knowledge are not neglected.

By the end of the analysis stage the knowledge engineer should have built the conceptual model with its model of expertise, model(s) of communication and user requirements. At the heart of the conceptual model is the domain layer of the model of expertise which includes all the information needed for problem solving. Otherwise each version of the system will have its own model of communication, inference and task layer. This means that when more than one version of the system is specified, it is impossible to just select an appropriate template from the KADS library of interpretation models. Some adaptation must take place. Difficulties may be resolved by small-scale prototyping.

Not all the components of the conceptual model can necessarily be completed before the design stage begins as the domain layer may still be extended to fill in small gaps that become apparent. Otherwise, the most important decisions should all have been taken. This should avoid the problem of having to backtrack and re-design a program if some new and important domain or user requirement suddenly comes to light that requires major re-structuring. FOCUS is viewed as a systematic way of ensuring that critical decisions are taken at the right time. As so many models have to be produced in parallel and checked against each other, it would be difficult to miss a major requirement. Extensive backtracking should not occur. Moreover, during the design phase the conceptual model can be used as the basis for developing both the knowledge base and the interface(s).

Chapter 6

The FOCUS Framework Applied: Initial Analysis

In this chapter, the initial analysis for an expert system to assist with student enrolment is described. The term "initial analysis" refers to the first four stages of FOCUS, that is *Problem specification, Preliminary analysis, User analysis* and *Functional specification*. All relevant models are described.

6.1 Problem specification

During the *Problem specification* phase, the knowledge engineer should discuss the problem that has arisen with members of the organisation to determine whether the knowledge-based approach is a suitable way to proceed. The output from this stage, the organisation model, should detail the problems and objectives of management, the sources of expertise that should be consulted and possible users of the system.

The opportunity arose to carry out the analysis for an expert system to assist students enrolling for a course of study at Massey University. It has been argued that the FOCUS approach of determining user requirements during the analysis phase of a knowledge-based life cycle ensures that all relevant human-computer issues are taken into account. Applying it to the enrolment application allowed this thesis to be put to the test.

The Head of Student Affairs at Massey University has for a long time been interested in automating the enrolment process. His discussions with staff in the Computer Services division indicated that knowledge-based systems technology might be needed to solve this problem. Other researchers have tended to confirm that planning programmes of study for academic degrees is a typical Artificial Intelligence problem (Golumbic *et al.*, 1986; Chew, 1987; Sharma, 1992; Weekaroon *et al.*, 1992).

6.1.1 The organisation model

The Head of Student Affairs believed that as far as the university administration was concerned there were various problem areas with enrolment. Firstly, a large number of enrolments both internal and extramural had to be processed by the start of the academic year. Applications exceeded 20,000 per annum. Both academic and administrative staff were heavily involved in this process. Secondly, there were always some students who decided to make major changes to their enrolment. A student may wish to move, for example, from a Business Studies to a Science degree. It could often take time to work out a suitable programme of study for such a student, whilst also having to decide how much credit they could be given for papers previously passed. Thirdly, every year a large number of students wanted to alter their choice of papers. This meant that they had to obtain several staff signatures on their form and the whole process was time-consuming. If they knew more about the papers for which they enrolled, the number of changes could be reduced. Fourthly, he was concerned about the quality of advice given to students. Many members of staff were involved in the enrolment process and some of them were not providing students with appropriate advice. He observed that at universities abroad students enrolled by computer and wondered whether this would be feasible at Massey University.

The initial description of the problem clearly identified the objectives of management. These can be summarised as the speedier and more efficient handling of enrolment applications, the provision of quality advice to students, and the reduction in the number of paper changes made once a course had been initially approved. The principal beneficiaries of a computerised system would be the staff who handled the large number of extramural enrolments and those students who studied on campus. It was not envisaged at this stage that extramural students would enrol via a terminal as they would not all have access to the relevant networking facilities for several years to come.

Various sources of information (people and documents) about enrolment were identified. The university issues many documents which contain information about university procedures: the University Calendar, faculty and department handbooks, a booklet for prospective students entitled "Flying Start", as well as guidelines for filling in the enrolment form. With regard to expertise, this is dispersed throughout the campus. In the university administration there is the Head of Student Affairs and his staff, the staff of the Liaison Office who talk to intending students, the deans of each faculty, the heads of departments, and some lecturers within these departments. Lecturers can, but do not necessarily, have a wide range of experience in this matter. The last source of expertise is the students themselves. They have to be acquainted with the regulations for their faculty and they have immediate knowledge of the papers that they have taken. Perhaps they are the only source of expertise for information about the lecturers on the course and the workload (assuming that no changes have been made to a paper).

For the preliminary investigation, it was decided to consult the University Calendar, "Flying Start", the faculty handbooks and the Enrolment Pack (Massey University, 1992a; Massey University, 1992b). The University Calendar contains schedules for all degrees as well as a description of every paper that can be taken. "Flying Start" is particularly useful for an overview of the enrolment process and contains a glossary of important terms whilst faculty handbooks present students with fuller details of courses offered. Finally, the Enrolment Pack gives students advice about how to complete the accompanying enrolment form. In addition, the author had been involved with the enrolment process for a number of years and was conversant with the demands of the Science, Social Science, Business Studies and Technology faculties. This experience meant that some but certainly not all the knowledge acquisition process could be reduced.

6.2 Preliminary analysis

In the *Preliminary analysis* phase, the knowledge engineer should ascertain whether one or more possible knowledge-based systems solutions to the specified problem can be identified. Knowledge about both the organisation and the domain must be elicited and recorded. Based on these findings, a provisional decision about whether to continue can be taken. If the project is to proceed, the model of expertise should be commenced and a picture built up of the users' situation within the organisation.

For the enrolment case study, documents were studied and follow up interviews for clarification held with the Head of Student Affairs. As a result of these activities it was possible to extend the organisation model whilst starting the model of expertise and the description of user issues.

6.2.1 The extended organisation model

Massey University was originally established as Massey Agricultural College in 1927, gaining full university status in 1964. At the time that the knowledge acquisition process took place, the university had 44 departments which were divided for administrative purposes into eight faculties and two schools: Agricultural and Horticultural Sciences, Business Studies, Education, Humanities, Science, Social Sciences, Technology, Veterinary Science, the School of Mathematical and Information Sciences and the School of Aviation. Undergraduates can enrol for degrees, diplomas and certificates offered by the different faculties. A wide choice of papers is available for these students with over 1000 papers offered to the undergraduates. In 1992, the total enrolment was 24,457 comprising 15,763 extramural students and eight and half thousand internal candidates.

Faculties are central to the functioning of the university with their ability to award degrees and diplomas. Each is headed by a dean and comprises several departments each administered by a professor. Associate professors, senior lecturers and lecturers make up the permanent academic staff of a department. This academic structure is shown in Figure 6.1. Degrees and diplomas are awarded subject to students fulfilling the course requirements specified in the faculty regulations. There are some minor differences between the faculty regulations, such as the number of points that can be obtained from papers offered by other faculties. More significantly, some faculties also demand that a practical requirement be met, for example Agricultural and Horticultural Sciences. In some cases a degree can be awarded after three years whilst others involve four or five years of study.

The course of study for each student according to the University Calendar has to have the approval of academic committee. In practice this is delegated to the dean of the faculty in which a student is enrolling. Checks are made by a dean or his/her representative to ensure that a degree or diploma course meets the faculty requirements. Likewise any later changes to the permitted course of study have to be approved. Heads of departments (usually but not inevitably professors) have the responsibility of checking that the papers relating to the student's main study area form a coherent whole. Other members of the lecturing staff within the department may also perform this task. When an upper limit is placed on the number of students who can enrol on a paper, the controller may have to choose the students on the basis of certain criteria.

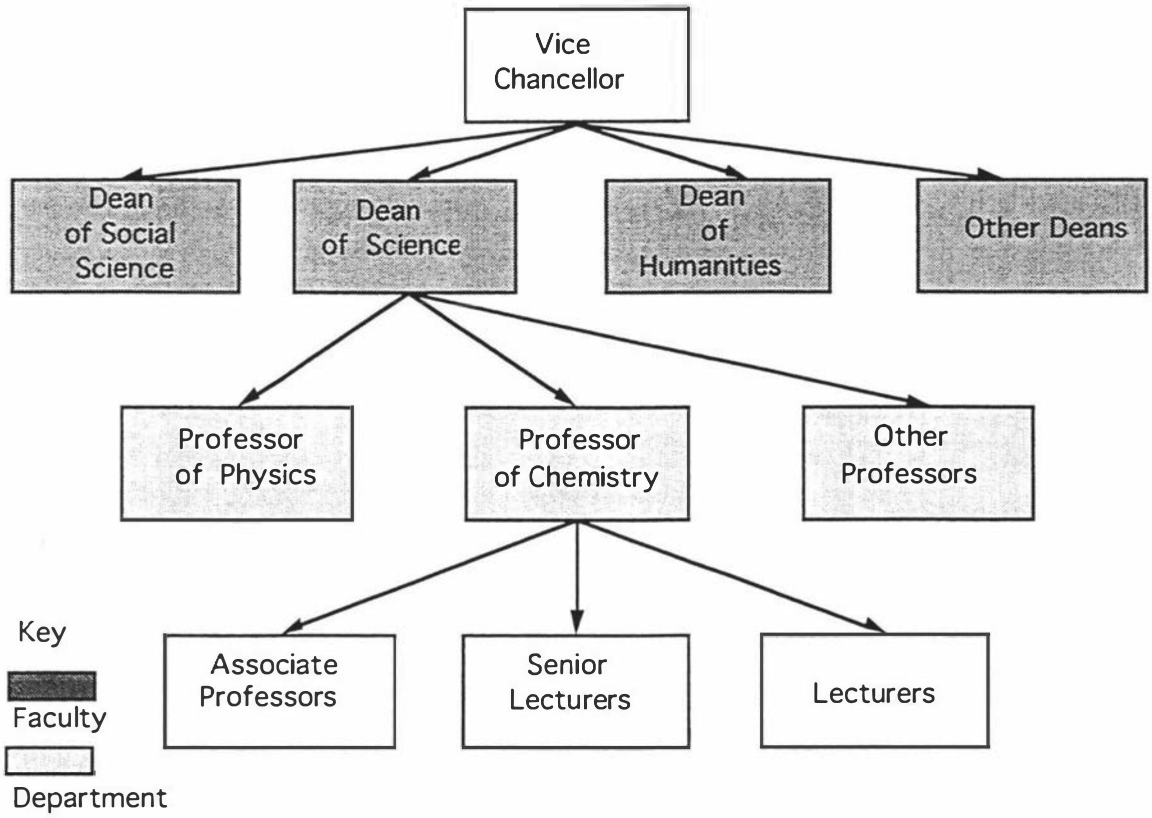


Figure 6.1 Academic structure of Massey University

Enrolment Procedures

After reviewing the enrolment procedures (see Appendix D for further details), it became apparent that current procedures would limit the usefulness of a fully automated knowledge-based system.

1. Final course approval by a dean would still be necessary until such time as procedures at the university were changed.

2. If academic staff wished to control entry to restricted papers in line with their own criteria then this would be difficult to automate.

3. Students can complete their degree under any set of faculty regulations that pertain since the year of their initial enrolment. There would always be some students who enrolled many years ago that could not be accommodated at all by the enrolment system. Older regulations often refer to papers no longer running and equating them to papers now offered would be an additional

problem. This task can usually only be accomplished by members of staff with many years experience.

4. Some students are exempted from taking first year papers or even, in some cases, the complete first year course of study. This could only be built into an enrolment system if students' bursary results were entered into the student database. Currently, this is not the case.

5. Students changing from one course to another (either within the university or from another institution to Massey) are allowed cross credits for papers that they have already taken. Whilst some of this information can be computerised, the range of possibilities is so great that many students would still have to discuss this with the head of the relevant department. The University Calendar specifically states that students who propose to change from one programme of study to another should write to the Registrar to determine which of the subjects they have passed may be credited to the new programme before enrolment. Students can also enrol in concurrent programmes of study such as a degree and a diploma. It is possible that some papers can be cross credited. Again it is preferable to discuss this with staff. Dealing with these matters can be quite a time-consuming process.

6. There is a great deal of flexibility within the system and requirements may be waived if a head of department thinks that a student has a good case. Indeed, some regulations specify that for entry to a particular paper either a specific prerequisite paper or the head of department's approval is needed. Sometimes an examination clash is permitted where a student has no obvious alternative course of action.

Any system built to assist administrative staff involved with extramural enrolments would be difficult to implement if academic staff were still allowed to make such important decisions as the waiving of pre-requisites. Furthermore if problems arose and a course of study could not be approved, no immediate advice could be given to students. These drawbacks ruled out a system for staff but still left students as a possible target group. This considerably narrowed the scope of the project and meant that no further involvement with the extramural section was necessary.

Alternative courses of action

Initially, it was thought that two groups of users could benefit from a knowledge-based approach but the staff alternative was ruled out on the grounds of the limited usefulness of such a solution. Nonetheless, having identified student problems and needs, some solutions could be considered. Firstly, there was scope for some computerised checking (that students had taken the necessary prerequisite papers, for example). This option seemed a partial solution only and was discounted. Alternatively, a system could be built to assist students on campus cope with enrolling for an undergraduate degree (the largest single category of internal students). This would meet the organisational goals of offering quality advice to students and could reduce the number of paper changes made after course approval. It was also a problem that required AI techniques since for any student there would be a large space of acceptable courses. Moreover advice (expertise) would have to be included within the system. What students wanted in the way of advice was still an unknown and would have to be elicited in the next stage, *User Analysis* (section 6.3).

A complete enrolment system would have to include all possible degree regulations. The analysis involved in this would be very time-consuming. A step by step approach seemed preferable. It was decided (in view of the author's background) to limit the scope of such a system to Computer Science students within the Science faculty. The needs of these students would be ascertained during the user analysis phase. A pilot study would be useless, though, if the system developed could not be extended. Consequently, it was decided to survey a large group of students from several faculties. Before drawing up a questionnaire it was essential to obtain more information about the users' situation within the organisation.

6.2.2 User issues

Whilst the university has its own problems with enrolment such as dealing quickly with a large number of applications, students too have their own particular concerns. The knowledge engineer knew, because of experience with the enrolment process, that internal students attempt unsuccessfully to do the following:

- enrol in a paper with limited numbers;

- enrol in a paper when they have not passed the necessary pre-requisite(s);
- enrol in a restricted paper;
- enrol in a paper when they have not enrolled for a necessary corequisite;
- enrol in papers whose examinations are at the same time;
- enrol in papers whose lecture times clash;
- enrol in too many papers.

In addition, their course of study may fail to meet the requirements of their degree. The university administration recognises the need to help students when planning a course of study. The various sources of assistance available are described below.

Academic staff

Academic staff are present in large numbers during the course approval process and counsel students as necessary. Students also make appointments to see them at other times. Deans often have to deal with the problems that arise when a student wishes to change from one faculty to another. Some students discuss options in a department with a professor or other senior academics. Students may also consult lecturers about the papers that they teach.

Liaison Office

Information about the work of the Liaison Office was obtained through discussions with the Liaison Officer. The author also accompanied two members of staff on a visit to a secondary school. It should be noted that this source of expertise had not been given its due priority in the organisation model and this deficiency had to be remedied.

The activities of the Liaison Office are numerous. Their staff visit schools all round the country (especially those in the vicinity of Massey University) twice a year. During June and July seventh formers also get the opportunity to spend a day at the university. Seventh formers intending to enrol at Massey University are interviewed by staff of the Liaison Office during a return visit to the school. Before their interview, the seventh formers are advised to plan their course later in detail, using the material in their Enrolment Pack and, if necessary, the University Calendar.

Individual counselling is a very important activity of the Liaison Office. In order to provide some continuity, the staff of the Liaison Office are also available to first year students at Massey University to help them resolve course planning problems.

Documents

Various documents are produced for students. The Enrolment Pack which is sent to all those intending students contains an enrolment form, an enrolment handbook, tuition fees booklet, lecture and examination timetable, and an appropriate faculty handbook. Detailed advice is provided to students to help them fill in the sections of the enrolment form. The faculty handbook gives course related advice and, if this is insufficient, department handbooks can also be consulted. Students in any doubt about regulations can always look them up in the University Calendar.

6.2.3 Model of expertise

As a result of the analysis that has taken place, the model of expertise can be started. Whilst the organisation and user models are being built up the domain layer can also be developed. Firstly, the lexicon can be started (Figure 6.2).

Definitions

Course
The grouping of papers for which you're enrolled. Most first year courses involve seven to eight papers. Sometimes 'course' is taken to mean a paper rather than a grouping of papers.

Major
The subject in which you specialise. A double major is a specialisation in two subjects.

Paper
The basic unit of teaching at Massey University. Can be half-year or full year. A required number of papers is needed for specific degrees.

Prerequisite
A paper which must be passed before taking another (more advanced) paper.

Corequisite
A paper which must be studied at the same time as another (or others), unless the corequisite has already been passed.

Restriction
A limit placed on the combination of papers within a degree eg. you can do one paper or the other but not both.

Figure 6.2 Important definitions in the lexicon

In this case, one of the university's own documents "Flying Start" was consulted (Massey University, 1992a; Massey University, 1993b). Its glossary contains definitions of relevant terminology such as paper, course, major, core paper, restriction, prerequisite, corequisite, and electives.

As a result of the knowledge acquisition process, the concepts, attributes and the relationships of the domain were elicited (Figure 6.3).

Concepts

Student
Paper
Student-history
Timetable
Faculty
Faculty regulations
Degree

Attributes

Of Student
    Student number, family name, given name, degree, major
Of Paper
    Paper name, paper number, paper points, paper exam date
Of Student-history
    Student number, paper number, paper grade, year
Of Paper-prerequisites
    Paper number, paper prerequisite number
Of Paper corequisites
    paper number, paper corequisite number
Of Paper-restriction
    paper number, paper restriction number

Relationships

A student enrols in one or more qualifications
A students submits an initial course of study
A course of study can be altered
A course of study is composed of one or more papers
A student enrols in many papers
A student may have passed one or more papers
The award of a degree is governed by a set of regulations
Each degree is awarded by one faculty
A faculty may award several undergraduate degrees
Some undergraduate degrees focus around a major
A major (eg. Computer Science) may be studied in more than one faculty
A major is composed of a number of papers
A paper may have one or more prerequisites
A paper may have one or more corequisites
A paper may have one or more restrictions
A paper has a name, number, examination date and points value associated with it
A paper may have one or more timetable slots associated with it
A timetable slot has a time of day, room, paper number and purpose (lecture, laboratory, etc)

Figure 6.3 Concepts, attributes and relationships

6.3 User analysis

Whilst the organisation's perspective is dominant in the first two stages of FOCUS, the emphasis moves to user concerns during *User analysis*. Information should be gathered to find out whether it is feasible, considering the skills and attitudes of the user, to proceed with the project. User requirements, domain background and familiarity with computers should all be obtained by the knowledge engineer. Further information about the domain may be elicited if the intended users of the system have some understanding of the problem area. This knowledge should be added to the model of expertise.

Once the scope of a possible system had been identified in the enrolment case study, the users' perspective was obtained by a questionnaire. The objectives of this study were to find out whether students thought there were problems with enrolment, to evaluate current sources of assistance, to find out about the general level of computing skills and to see whether students wanted to use a computerised enrolment system. Furthermore, since users proved to be another source of expertise, it was also possible to extend the domain layer of the model of expertise.

The user study at this stage involved the collection of data by a questionnaire (see Appendix E). To ensure that all the respondents interpreted the items in the questionnaire correctly a pilot study was carried out with 30 students. Comments on its comprehensibility were also obtained. Only a few small changes had to be made as a result of this process. Whilst the main focus of this study was on Computer Science students within the Science faculty, some overall context was needed. General problems at Massey University had to be identified as well as the more specific needs of the BSc students. The sampling process was set up to ensure that a comparison could be made between students from faculties whose regulations were fairly flexible (Science, Social Science, Business Studies) and students from faculties where the course of study was more prescribed (Technology and Agricultural and Horticultural Sciences). Care also had to be taken to include a reasonable proportion of females as well as males. Finally, input was also needed from those with little experience of computers. The number of relevant factors made surveying difficult. It was decided in these circumstances to use purposive sampling, that is where specific people are targeted (Sekaran, 1984). Purposive sampling is a form of non-probability sampling which allows the researcher to ascertain a wide variety of views. This approach cannot ensure representativeness but often provides surprisingly efficient predictions (De Vaus, 1991). Students on particular first, second and third year papers were asked to fill in the questionnaire. Subject areas (Botany, Geography, etc) were picked that would enable the capture of the relevant information and classes chosen arbitrarily from the possible alternatives.

The 31 item questionnaire elicited information about the age, sex and course of study of the students. Both open-ended and closed questions were included. The closed questions were of two types. In some cases Yes/No responses were required. Others involved rating the variables of interest on a 7 point Likert-type scale so that means and medians could be calculated. Students were asked to state their opinions on a wide range of issues in the open-ended questions. Since students did not have to give either their name or identification number, complete anonymity was ensured. This allowed students to discuss university procedures without any fear of the consequences. The data collected is shown in Appendices F and H. Numerical responses and comments are both included. Completed questionnaires were ordered within paper number. For example, the first response processed from the students taking the third year paper Human-Computer Interaction is numbered 59327 # 1.

6.3.1 Further user issues

Students supplied details about their sex and age as well as their course of study. In total, there were 421 respondents of whom 165 (39%) were female and 256 (61%) male. In the 5 faculties of particular interest the ratio of females to males was 42% to 58% and this was almost but not quite the case with the survey where 38% of the respondents were female and 62% male. Four hundred and fourteen of the respondents were prepared to give their age (albeit approximately, for example "40 plus"). The youngest student was 16 years of age whilst the oldest stated age was 47. The bar chart in Figure 6.4 shows the age profile of students quite clearly. About three quarters (299) of the 414 respondents who stated their age were under 22, but 52 (an eighth of the total) were older than 25.

The faculty in which a student enrols is very important. Students receive faculty related information from the Registry and interact principally with the staff of that faculty. A breakdown of the number of responses by faculty is shown in Figure 6.5. As intended, the largest single group was composed of Science (Sc) students with reasonable numbers from the other four faculties of interest: Agricultural and Horticultural Sciences, Business Studies, Social Science and Technology (abbreviated to AH, BS, SS and Tech respectively). In addition, 19 first year Veterinary Science (Vet) students filled in the questionnaire as well as small numbers from the Humanities (Hum) and Education (Ed) faculties. The "Other" category refers to students who would completing their degree (in subjects like Medicine and Architecture) at another university.

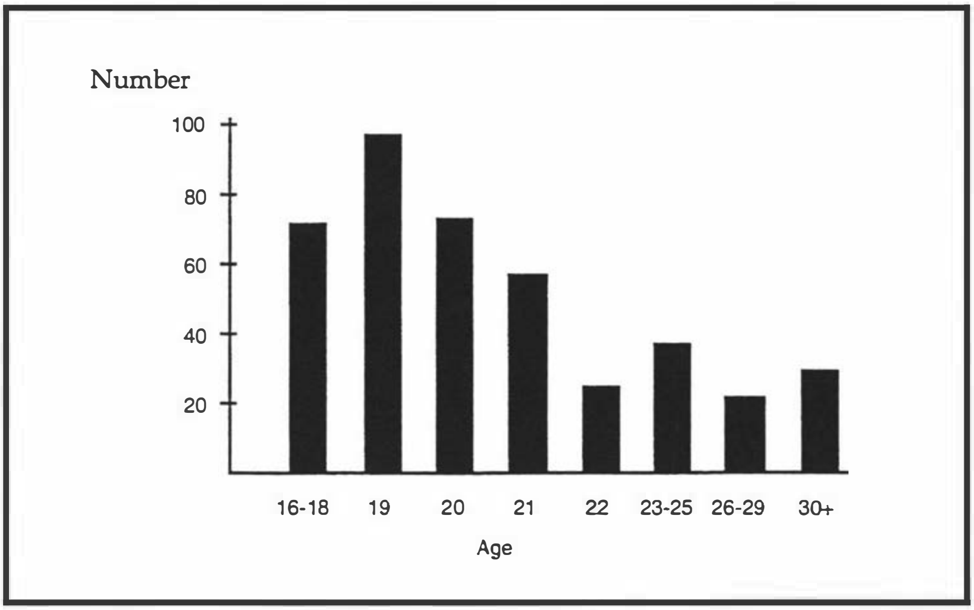


Figure 6.4 Analysis of students by age

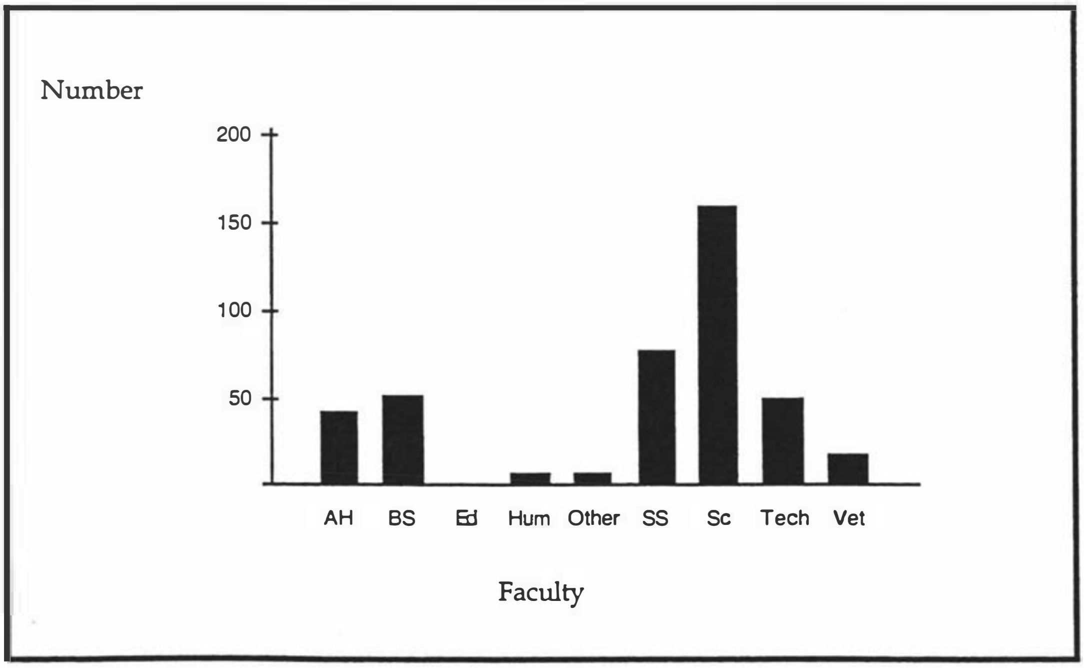


Figure 6.5 Analysis of students by faculty

A wide range of subjects was studied by the students who responded (see Appendix G). For the purposes of this study they can be divided into two groups - those who studied computing and those who did not. Computing is defined as any of the these options: Information Systems, Computer Science, Computer Systems Engineering or Information Technology. The population of respondents could then be divided into 185 computing students and 236 others. Of the 185 computing students, 49 were female and 136 male. This is a ratio of about 1:3 and is quite different from the composition of the other group which is virtually fifty-fifty (116 females and 120 males). For the faculty in which we are particularly interested (Science), there were 72 computing majors and 88 others. It was seen as essential to check during the analysis of the responses whether the non-computing students felt at ease with computing technology.

Problems with the enrolment process

From the interview with the Head of Student Affairs and the author's own experience, the enrolment process seemed beset with problems but it was important to see whether students confirmed this. Consequently, they were asked to state whether or not they had any of a specified set of problems (questions 17 to 21). There was also an open-ended question that asked them if they had any other problems with the enrolment process. Overall, the answers indicated that there was considerable concern over enrolment.

Quite large numbers of students appeared to have problems. Table 1 shows the question asked, the number who said "Yes" and this number as a proportion of the total respondents.

Did you have any of the following problems when enrolling this year?	Yes response	
	Total	%
17. Resolving lecture and examination clashes?	130	31%
18. Knowing how many papers to take from outside your faculty?	91	22%
19. Planning a course that you can complete in three years?	109	26%
20. Knowing which papers you can take next year?	121	29%
21. Planning a course when you have failed pre-requisite paper or obtained a restricted pass?	56	13%

Table 6.1 Problem areas

Timetabling a course, planning a course that can be completed in three years and knowing which papers can be taken next year were all seen as important with over a quarter of students reporting difficulties in these areas. The other two issues were of lesser concern with only a small percentage hampered in their planning by failing a paper. Since faculty regulations are quite different, it was expected that there would be a difference between the number of positive respondents from the various faculties for the first four of these questions. Failing a paper was not seen as something that would be related to a faculty. It was assumed that the average level of intelligence for each group of students was similar. To check whether there was a relationship between the faculty in which a student was enrolled and the responses, the proportion of students in each faculty who answered" Yes" to questions 17-21 was examined (Table 6.2).

Question number	AH	Tech	BS	Sc	SS
17	12%	10%	50%	35%	38%
18	14%	6%	27%	24%	28%
19	14%	2%	29%	41%	24%
20	23%	8%	38%	33%	36%
21	14%	16%	17%	15%	9%

Table 6.2 Analysis of problems by faculty

Since in two of the faculties the course is more prescribed (Technology and Agricultural and Horticultural Sciences), it was expected that these students would face fewer timetabling problems, that is fewer lecture and examination clashes. This proved to be the case, with less than one eighth of the students from Agriculture Horticulture and Technology responding "Yes". It was quite different with students from the Business Studies Faculty, half of whom reported that they had a problem. Students in the Science and Social Science

Faculties fell somewhere in between but over a third of these students indicated that they had difficulties with the timetable.

It was expected that a similar trend would be seen with responses to the question "Do you have problems knowing how many papers you can take from outside your faculty?" Indeed only 6% of Technology students had this problem with a somewhat larger percentage of Agricultural and Horticultural Sciences students (14%). Both of these figures were different from the results for the Science, Social Science and Business Studies students where about a quarter of the respondents were not aware of faculty regulations concerning how many other papers they could take.

Whether or not a course could be completed in three years was of no relevance to Technology students who have to study for four years. Only 2% were concerned with this issue. It was also of minor concern to Agricultural and Horticultural Sciences students (only 14%) responded "Yes" but was more of a worry for the remainder. About a quarter of the Social Science and Business Studies students had this problem but the figure for the Science Faculty was much higher at 41%. Finally, as predicted, the proportion of students in the three more flexible faculties who did not know which papers could be taken the following year was similar. Roughly a third said that they faced this predicament. In contrast, less than a tenth of Technology students had this problem and fewer than a quarter of the Agricultural and Horticultural Sciences students.

As anticipated there was no relationship between the sex of respondents and the problems faced during enrolment. Whether or not a student was a computing major was also unimportant. The main discriminant was the faculty to which a student belonged.

Many students took the opportunity to write in the appropriate section about problems that they faced during the enrolment process. One hundred and forty-seven complaints were made. These are classified for the purposes of discussion and analysis under the following headings: mistakes, difficulties planning a course, problems with course approval and other criticisms of the process (see Appendix H2).

Mistakes

The enrolment system handles a large number of applications and mistakes are bound to occur. The most common complaint involved the loss of the course approval form. Many students arrived to obtain approval for their course of study and found that there was no form to be signed. Two students were still very angry about the problems this caused them months after the event. The implications for a student can be serious if their name is not added to the necessary class rolls. For instance, students can be denied access to relevant facilities such as computing laboratories. This also happens to students whose course of study is approved but whose form is subsequently lost. Other mistakes affected students on an individual basis such as the student who was asked several times for information that had already been supplied.

Whilst much information is made available to students, several mistakes in the documents published by the university were reported. This is hardly surprising given that there are so many sources of information for students to reference; some errors are almost bound to creep in. Students complained that they were not able to get up-to-date information about what papers were being taught in the academic year for the following reasons:

> *faculty handbooks in enrolment pack were outdated by enrolment time*
> (59327 #7)
>
> *information was incorrect eg. papers listed when not actually offered*
> (75305 #13)
>
> *calendar and departmental information were in conflict*
> (59327 #13)

For other students, errors in the timetable made it difficult for them to even plan their course. Mistakes in documents can obviously have a wide impact as they affect all those who consult them.

Course planning

Even with all the information provided by the university, some students felt that they were not given enough information about papers. This made it difficult for them to find out about prerequisites and corequisites, obtain information about double majors, plan for the future, obtain career related information, choose a major or find out about the content and workload of papers. Many students were also frustrated by timetable or examination clashes since they could not necessarily take the papers that would suit their needs. This also occurred if a student on the internal roll was not allowed to take a paper taught only in extramural mode.

Students choose their papers prior to the course approval process. They may later change their programme of study if they wish. Alternatively, they may be refused entry to a paper and have to select another one. Both these situations caused problems. One commented that it was hard to change papers "after deciding you didn't want to do one" (57102 #47) whilst another mentioned the problem of quickly deciding on another paper when permission to take a paper was refused at course approval (75305 #2). One student implied that permission for a change was difficult to obtain when the new paper was not in the faculty schedule. In certain cases, paper approval is refused because students have done so well in their bursary examinations that they are exempted from pre-requisite or core papers. When students are aware of this in advance there is no problem but sometimes they arrive at course approval and find out at the very last minute that they have been granted exemptions and need to find other papers in which to enrol. This does not allow them much time to consider their options. Several first year Veterinary Science students complained about this although they were not the only ones affected. The following account was typical of what happened to this group of students:

> *I was exempt two first year vet papers. I was told this only on the day when I arrived at Massey for enrolment, even though my form with the relevant details had been sent weeks before-hand, I quite unexpectedly had to suddenly find two different papers to take and would have appreciated it if I had been notified before enrolment day.*
>
> (57102 #7)

Some students decide to change direction. Having undertaken a course of study that leads to one qualification they decide that they would rather enrol for another degree. Both students already enrolled at Massey and those moving from other educational institutions might change their degree. Permission has to be obtained to cross credit relevant papers that have already been taken to the new qualification. The cross crediting process causes some students serious difficulties as the process can be very time-consuming.

Course approval

Students often disliked the course approval procedure itself as it involved spending a considerable length of time in queues. Many were annoyed that they spent a long time in line when all that was needed was a signature. For one student the course approval process took a considerable time: "Takes too long to get course approval - a whole day." Another had to return on the second day to correct the mistakes made on the first. For some, the problem was compounded by not knowing where to go for course approval or whom to see.

Other criticisms of the enrolment process

Many other criticisms were also made that could not be so easily classified. Some only affected the individual concerned whilst others were more general such as complaints that it was difficult to work out the fees correctly or change from the extramural to the internal roll. Unfortunately, many students were less than complimentary about Massey procedures. Some students felt that the enrolment form was cluttered and difficult to interpret. One student (60103 #23) even said "You almost need a degree to fill out enrolment papers." Another was unsure about exactly what documents to send with the enrolment form. The large amount of information made available to students was not always welcomed. One student objected that reading and understanding the volumes of information relating to procedures for enrolment was largely an unproductive waste of time. For another it was difficult to co-ordinate all the different handbooks. Some students commented that when they tried to get answers to their questions, staff were less than helpful. One embittered student (75305 # 17) said, "No matter what question you ask, the lecturers or advisers never give you the right type of answer." Another stated that he was made to feel stupid when he tried to sort matters out (20205 # 9).

To sum up, one hundred and forty-eight complaints were made by the students (see Table 6.3). A third of these problems related to planning a course of study whilst almost a quarter were concerned with course approval procedures. Indeed, the single most common complaint (made by 29 respondents) was having to queue for signatures on the course approval form. Mistakes formed a somewhat smaller proportion of the total (16%). Finally, another quarter of the complaints were about some aspect or another of the enrolment process that seemed less than satisfactory. Overall, enrolment seems to be a confusing process. Students clearly demonstrated a wide range of concerns. If one theme did emerge it was the difficulty of finding relevant information. Even the queuing that occurs is partly a reflection of the need that students have to talk to staff to obtain some counselling about their enrolment application.

Evaluation of sources of assistance

Since information would have to be presented to students by a computerised system, it was important to know what sources of information were used by students and how they rated them. If there was an overwhelming preference for information given orally this would have important implications. The sources selected for evaluation were: the Enrolment Pack, faculty handbooks, University Calendar, department handbooks, deans, professors, lecturers, fellow students and the Liaison Officer. Students were asked to rate these on a scale of one to seven where 1 represented "no help" and 7 "very helpful". Only sources of assistance actually used were to be rated. In a follow up question, students were asked to specify any other sources of information. Twenty-six students answered this question, mentioning that assistance had been received from other members of Massey staff such as secretaries and tutors as well as from their own relatives.

	Total	% of problems	% of all students
Course planning	52	35%	12%
Criticisms	38	26%	9%
Course approval	34	23%	8%
Mistakes	24	16%	6%

Table 6.3 Problems reported by students

If the sources of information are ranked by the numbers of those who used them, then the Enrolment Pack tops the list (408 students) with deans (223) at the bottom (Table 6.4). An analysis of the list shows that there are essentially three groups: those sources rated by more than three quarters of the students (the Enrolment Pack, faculty handbook and students), those used by between two thirds and three quarters (University Calendar, lecturers and department handbook) and those used by between a half and two thirds (Liaison Office, dean and professor).

Another way of looking at the data is to rank the sources of information in order of their evaluation by students (the student's responses were, it should be remembered, based on a scale from 1 to 7). Table 6.5 shows that three of the sources had a mean of 5.2 (Enrolment Pack, faculty handbook and Students). The relevant faculty handbook is, as mentioned previously, included with the Enrolment Pack so one would expect their results to be similar.

Source	Number	Percent
Enrolment Pack	408	97%
Faculty Handbook	371	88%
Students	359	85%
University Calendar	307	73%
Lecturers	305	72%
Departmental Handbook	277	66%
Staff of Liaison Office	245	58%
Professor	228	54%
Dean	223	53%

Table 6.4 Ranking by order of usage

Source	Mean	Median	Number
Faculty Handbook	5.2	6	371
Enrolment Pack	5.2	5	408
Students	5.2	5	359
Lecturers	4.9	5	305
Department Handbook	4.9	5	277
Professor	4.4	5	228
Dean	4.3	5	223
Staff of Liaison Office	4.3	4	245
University Calendar	4.1	4	307

Table 6.5 Student assessment of sources of information

Once again the sources can be partitioned into three different groups: those with means above 5, those with means between 4.5 and 5 and those with means below 4.5. With one interesting exception, each source of information is placed in the same category (top, middle or bottom group) whether either the number of responses or the means are considered. Those sources of information that are highly rated have been used by over 85% of the respondents. On the other hand those less frequently used sources rated rather more poorly. It could be argued that students know which are the most helpful sources and use them. There is one notable exception to the previously described rule, the University Calendar. This was rated by almost three quarters of the students but has the lowest mean, 4.1. The University Calendar is a reference document and the Bible for the university's regulations. As such it is often consulted. Nonetheless, its particularly poor showing indicates that its content is not easily accessible to its readers. For comparison, the bar charts for the top and bottom ranked options are shown in Figures 6.6 and 6.7.

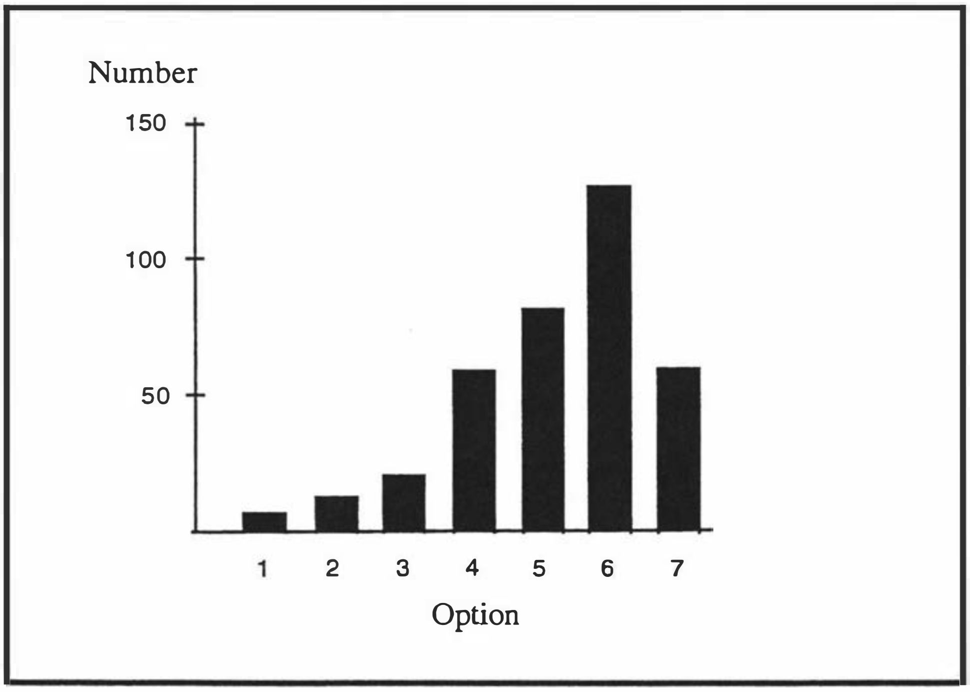


Figure 6.6 Helpfulness of Faculty Handbook

An analysis of the results by faculty (Table 6.6) shows that both the Science and Social Science students rated the sources of assistance quite highly. If the medians are considered (as these are not affected by outliers), there was not one source of assistance that was rated below five by the Social Science students. The results for the Science students were very similar. The least satisfied students appear to be those from the Business Studies and Technology faculties. This cannot be explained here except to note that, since each faculty operates in a different way, such discrepancies are to be expected.

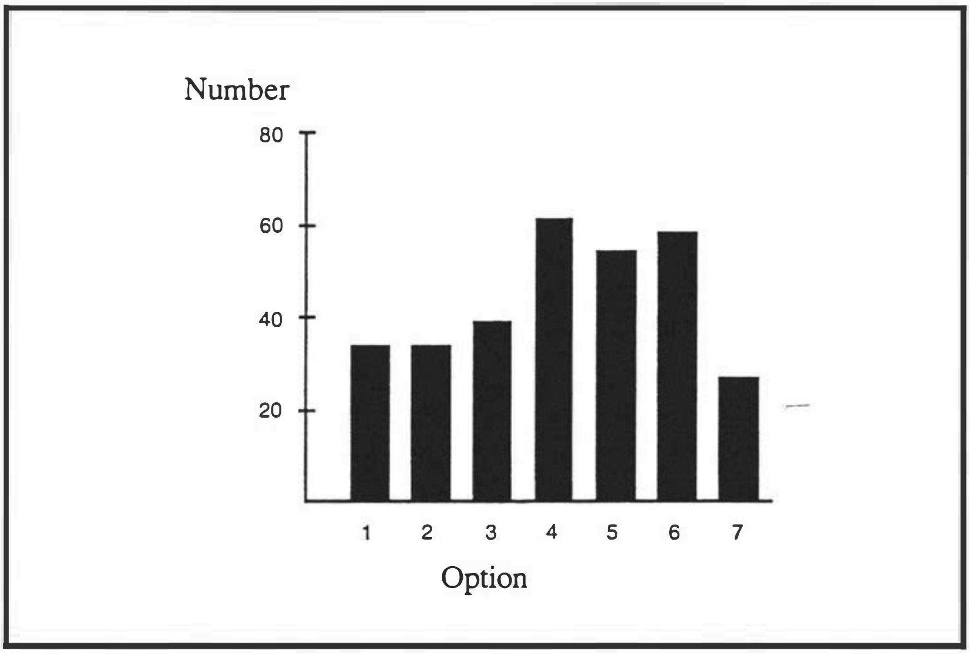


Figure 6.7 Helpfulness of University Calendar

Computing background

Several questions were asked about the computing background of the students. Their experience with both computer platforms (Apple Macintosh or other PC) and software was ascertained. Students were asked to state whether they had used a word processing package, a spreadsheet, a database or even programmed. This experience was seen as relevant not just as a measure of computer usage but because of the skills that such experience would develop. Using a word processor helps with the acquisition of keyboard skills, database usage relates to the ability to query, spreadsheets provide familiarity with a graphical interface whilst those who program have to know something about computer technology. The proportion of respondents with the relevant experience was calculated. Not only the overall percentages but those for particular groups of interest were determined. These groups were chosen to reveal any major differences between male and female students or between the computing (Comp) or non-computing (Non-comp) majors. The results of male and female non-computing students were also checked to ensure that no important differentiating factor was overlooked. As the pilot study was geared specifically to the needs of Science students their results were also analysed.

	AH	Tech	BS	Sc	SS	Overall
Enrolment Pack						
Mean	5.4	5.0	4.9	5.1	5.4	5.2
Median	6	5	5	5	5	5
University Calendar						
Mean	3.2	3.3	4.0	4.4	4.4	4.2
Median	3	4	4	5	5	4
Department Handbook						
Mean	4.6	4.1	4.8	5.0	5.0	4.9
Median	4	5	5	5	5	5
Faculty Handbook						
Mean	5.6	5.3	4.6	5.3	5.3	5.2
Median	6	6	5	6	6	6
Liaison Officer						
Mean	4.6	3.5	4.0	4.4	4.4	4.3
Median	4.5	4	5	4	5	4
Dean						
Mean	3.7	3.4	3.3	4.9	4.3	4.3
Median	4	4	3	5.5	5	5
Professor						
Mean	4.2	3.7	3.3	4.8	4.6	4.4
Median	5	4	3.5	5	5	5
Lecturer						
Mean	4.7	3.9	4.9	5.0	5.0	4.9
Median	5	4	5	5	5	5
Students						
Mean	5.6	5.1	5.2	5.2	5.3	5.2
Median	6	5	5	5	5.5	5

Table 6.6 Student assessment of sources of assistance analysed by faculty

The results in Table 6.7 show a very computer literate population. Eighty-two per cent (344) of students had used a Macintosh. This is the main platform for teaching introductory computing on both majoring and service courses at Massey University. A somewhat smaller number (311 students or 74%) had some experience with another PC. Overall, only 25 (11%) of the non-majoring students had never used a computer. Given this high level of usage, further analysis revealed little. Even when a group scored quite low in one area, it was compensated for elsewhere. For example, whilst only 58% of the female non-computing students had used a PC, 75% had used an Apple Macintosh.

Usage	Overall	Science	Comp	Non-Comp	Female	Male	Female Non-Comp	Male Non-Comp
Macintosh	82%	84%	94%	72%	80%	83%	75%	69%
Other PC	74%	68%	87%	64%	67%	78%	58%	69%

Table 6.7 Analysis of computer usage

With regard to software experience, Table 6.8 shows that most students had used word processing(W-P) and spreadsheet (Spread) packages. The figures were somewhat lower for database (DB) and programming experience (Prog). Again, there were few important differences. Whilst the proportion of females who programmed was not very high in comparison with their male counterparts (particularly in the non-computing category), many had spreadsheet and word processing experience. A large proportion of non-computing majors (81%) had used a word processing package.

Experience	Overall	Science	Comp	Non-Comp	Female	Male	Female Non-Comp	Male Non-Comp
W-P	87%	83%	96%	81%	84%	90%	81%	81%
DB	66%	57%	82%	54%	61%	70%	53%	54%
Spread	82%	76%	96%	71%	76%	86%	68%	74%
Prog	60%	68%	92%	34%	47%	68%	28%	40%

Table 6.8 Analysis of software experience

Students were asked whether they preferred text or window interfaces (see Table 6.9 for the results). Half of the students chose a windows interface whilst a third either specifically stated that they had no preference or indicated this by leaving the questionnaire blank. Less than a tenth chose the text alternative. Some students did not wish to choose between these alternatives and noted that they wanted both (6%). There were some important differences between the computing and non-computing students. Only 19% of the computing students failed to make an explicit choice whilst the corresponding figure for the non-computing students was 47%. This is hardly surprising as many students in the latter group noted on the questionnaire that they did not know the difference between window and text interfaces. Both groups preferred the windows option but this won less support from the non-computing students, 45% as against 58% of the computing students. The figures for the selection of text were also different but the discrepancy was not quite so large with 15% of computing majors but only 4% of the others choosing this. An interesting comparison can also be made between males and females since 41% of the females compared to 30% of men showed no preference. They were also less supportive of the windows option than their male counterparts.

Interface Preference	Overall	Science	Comp	Non-Comp	Female	Male	Female Non-Comp	Male Non-Comp
Text	9%	5%	15%	4%	7%	11%	6%	2.5%
Windows	50%	54%	58%	45%	45%	54%	40%	49%
Both	6%	7%	8%	5%	7%	5%	7%	2.5%
No Pref	34%	34%	19%	47%	41%	30%	47%	46%

Table 6.9 Analysis of interface preference

When only the responses of the two thirds who made a choice were analysed, though, there was overwhelming support for the windows option. Overall, 77% of this group chose windows only. If those who chose both is also added on then this figure rises to 86%. As a glance at Appendix H1 shows, a large number of reasons were given for preferring window-based systems. Overwhelmingly students viewed them as user friendly: easier to use, easier to get started, easier to swap applications, easier to do things not attempted before. This ease of use related to the mouse driven interface with many students observing that they did not need to memorise commands. Consequently, window based systems were seen as less error prone. Pull-down menus were also mentioned as a useful feature since they enabled users to see all the choices available and to generate ideas. Overall, students perceived window-based systems as an environment which allowed them, with only a little knowledge of computers, to get started and achieve something useful. Some students did, however, qualify their answers; their preference was for window-based systems as long as short cuts were provided and pull down menus did not always have to be used.

Whilst two students found that text based systems were user friendly, a large majority mentioned speed as the principal advantage. A couple of students mentioned that text was faster once the commands had been learned. Another student noted that typing was much quicker than clicking. Window systems in contrast were seen as too slow. Text was also favoured because of the greater control that it gives users. One student felt quite strongly about this, "I find window based systems insulting to my abilities, they are childish" (59112 #12). This view was put forward, albeit in a more sophisticated fashion, by the student who commented that window based systems isolate the user from the full potential of the system (59112 #70). Others favoured text because they had used it for years or thought such systems were more useful in the real world.

When asked whether they would use a computerised enrolment system, just over half of the 409 respondents said that they would definitely make use of such a facility (Table 6.10). Only 4% responded with a "no" whilst the remainder replied "possibly." There was some difference between those students majoring in computing and the others. Fifty-eight per cent of the computing majors and 45% of the non-majors replied that they would use an advisory enrolment system. Presumably because of their lack of confidence with computers half of the non-majors replied "possibly."

Response	Overall	Science	Comp	Non-Comp	Female	Male
No	4%	4%	3%	5%	4%	4%
Possibly	45%	43%	39%	50%	45%	45%
Yes	51%	53%	58%	45%	51%	50%

Table 6.10 Analysis of computerised enrolment system

Computerised assistance

Students were asked to evaluate the usefulness of several potential features of a computerised enrolment system. As students were asked to rate features that they had not seen implemented but thought would be helpful, it is not surprising that all suggestions received high ratings (Table 6.11). The suggestion that received most support was that the proposed system should provide information about each paper (mean of 6.3 and median of 7). All the others received about the same level of support except for the proposal that explanations of terminology be provided.

	Mean	Median
Explanations of terminology	5.2	5
Information about papers	6.3	7
Planning the current year's course	5.6	6
Planning for the future	5.6	6
Cost of paper	5.7	6
Examples of courses	5.4	6
Career information	5.7	6

Table 6.11 Evaluation of proposed features

In addition, 70 suggestions were made about facilities that a computerised enrolment system could support (Appendix H3). These fell into two categories: course related (63) and computer related (7). To deal with this latter group first, students suggested a variety of features that they thought would be useful including menus, help and print facilities. Some students also wanted remote access to the system. One computer science major even proffered advice about the kind of system that should be provided, " Very important to have an easy to navigate system eg. Hypercard designed well" (59327#20).

The other requests can be divided into several categories (Table 6.12). Many students (21) wanted more information about papers. A description of the paper (from the perspective of students), relevant prerequisites and corequisites, textbook information, the number likely to be on the course and the pass rate were all requested. A related issue was the name of the staff lecturing on a paper. Nine students asked for this. This obviously confirms the earlier finding that the most wanted feature in a computerised system was the provision of information about papers. Eleven students wanted to see the timetable for their selected course of study. A planning component was mentioned by eight students. Some wanted to see what would happen in the event of a course change whilst others asked for assistance in building a course that gave them the required number of points. For two of these students more long term assistance was wanted so that they could plan their studies for the three year period. These were the main requests made but other issues were mentioned by small numbers. There were four students who wanted the system to give the names of staff with whom they could discuss enrolment issues, four who asked for career advice and four who thought that a computerised system could also deal with the financial side of enrolment such as the calculation of fees and student allowances.

	Total	**% of requests**	**% of students**
Paper information	30	43%	7%
Timetable	11	16%	3%
Planning	8	11%	2%
Computer issues	7	10%	2%
Career advice	4	5.5%	1%
Staff names	4	5.5%	1%
Finance	4	5.5%	1%
Facilities	2	3%	5%

Table 6.12 Analysis of student suggestions

Summary

It seems likely that many students are confused and bewildered by the whole process of enrolment. The total number of complaints about the enrolment process made by the 421 respondents was 654. This was an average of 1.5 complaints per students, revealing considerable disquiet about the enrolment process. Even though various sources of assistance were quite highly rated this did not prevent student from requesting computerised facilities in large numbers. Of the 421 respondents only 16 stated that they would not use a computerised system.

There seemed to be no problem with providing a computerised enrolment system even though such a system would only be used infrequently. In total, 344 students had used an Apple Macintosh and 311 another PC. Indeed for students who were not taking computing as a major subject, 70% had used the Macintosh and 63% another PC. Overwhelmingly, students wanted an advisory system to provide them with information about papers so that they could make effective choices. Assistance with planning would also be useful as would career related advice.

6.3.2 Model of expertise

The model of expertise was extended with the addition of the concepts "lecturer" and "paper information" (Figure 6.8). The required attributes and relations were also identified.

Attributes

of lecturer

  lecturer name

of paper-information

  paper content, paper assignments, paper enrolment, paper lecturer (s)

Relations

paper information has a content, assignments, enrolment number and lecturer(s)

students require paper information

Figure 6.8 Extended domain layer

6.4 Functional specification

The *Functional specification* phase involves the completion of the organisation model. If a decision is made to proceed with the knowledge-based system, the knowledge engineer should specify the functionality of the knowledge-base and state which user groups will be supported. The criteria for interface evaluation should also be detailed. A high-level task model showing the problem-solving capabilities of the system must also be built. This allows the selection of any appropriate templates from the KADS library of interpretation models. The selected template(s) provides the basis for extending the model of expertise, that is developing the inference and task levels.

Once the user analysis for the enrolment problem had been completed and the models refined, it was possible to complete the organisation model by specifying the functions of the proposed system and its impact on the organisation.

6.4.1 Final organisation model

A system can be developed to enable undergraduate internal students build, in conjunction with the system, a course of study that fulfils the required regulations. The student database will provide student history details and a regulations database the university information. Since the system carries out a great deal of checking, a full explanation of any results will be given and the user will not have to request this information. Given that some students may not understand the university's terminology, clarification can optionally be obtained. As requested by students, paper information will also be provided. An approved course of study can be printed out at the end of a session and attached by students to the enrolment form, possibly acting as a confirmation that the course of study has been checked. The system will also print out, if required, the student's timetable, paper outlines and a summary of progress to date with a degree. Although the system would be developed for Science students, the framework provided should be easy to extend to students in other faculties. It was anticipated that a system of this kind would help reduce the number of paper changes made by students and ease the process of checking that regulations had been satisfied.

Organisational constraints mean that the system cannot cater for all problems faced by students such as those who change degree or move from another university. These would have to discuss their situation with appropriate staff. Two other associated modules could be developed in the future - one for long term planning and another to help students decide which degree to take. A small scale system of this type was implemented under the author's direction by a student from Wanganui Polytechnic. Such possible additions would have to be borne in mind when developing the system as different interface presentation styles may be needed.

As there is only one group of users for this highly constrained application, it is unnecessary to develop more than one interface. Since course enrolment is currently an annual ritual, the system will only be used infrequently by students. It is not possible in these circumstances to train users and the knowledge base and its interface will have to be easy to use. Although students have widely differing computing skills, the results of the study indicate that a large proportion have some experience with computers so this should not be an unrealistic goal.

Criteria for evaluation

The evaluation of a system involves checking both its functionality and the delivery of the required features. With regard to the former, this will require observation of students using the system and follow up interviews. Such an audit will find out whether students can complete their enrolment by computer. In the survey, students stated that they were particularly interested in receiving information about papers. During an evaluation, students can be invited to give their opinion of a feature providing the relevant details. Finally, students should be asked whether any important functions are missing. This evaluation should be done at the design stage using a prototype; it is too late once the system is up and running.

It is of crucial importance that the system is easy to use. Many checks can be made (Ravden and Johnson, 1989) but the following are the most important in this case.

1. The interface must be consistent, that is the method of entering and displaying information should ensure that the same type of information should be the in the same area on the screen.

2. The system should fit in with the way the user would carry out the task, that is the user's model of the task matches the task domain:

- terminology familiar to the user should be selected;
- any unfamiliar terminology should be explained;
- the organisation and structure of system should fit the user's perception of the task;
- the sequence of activities should be what the user would expect;
- it should be clear what stage the system has reached;
- it should be clear what the user needs to do to complete a task.

3. The user should feel that he/she has the appropriate degree of control:

- the system should be easy to navigate;
- appropriate feedback should be given;
- users should be able to undo and redo paper selections;
- users should have some control over the order in which they request information.

4. The interface metaphor selected should be appropriate; the suitability of the forms-based approach should be ascertained.

Overall, a benchmark could be set that 80% of the users should be able to run the system with no assistance.

6.4.2 Task model

Once the functions of the system had been identified, the task model was drawn up (see Figure 6.9). This shows at the top level the real world task that has to be carried out. This can be further decomposed into the major activities that have to be carried out to achieve the goal. Since much analysis remains to be done, the task model is only skeletal at this point and is further refined in the next stage. A course is proposed, checked and (if necessary) revised. The development of the task model is made easier by the identification of the appropriate generic task and selection of the requisite interpretation model from the KADS library (Hickman *et al.*, 1989). Essentially, this is a construction system which can act in an advisory mode. In the KADS classification we are looking specifically at the generic task "configuration" where the artefact to be produced is the logical entity, "approved course of study". This problem has the usual characteristics of configuration problems; a large space of possible alternatives has to be cut down to meet highly specified constraints (the university's) and requirements (the needs of the student).

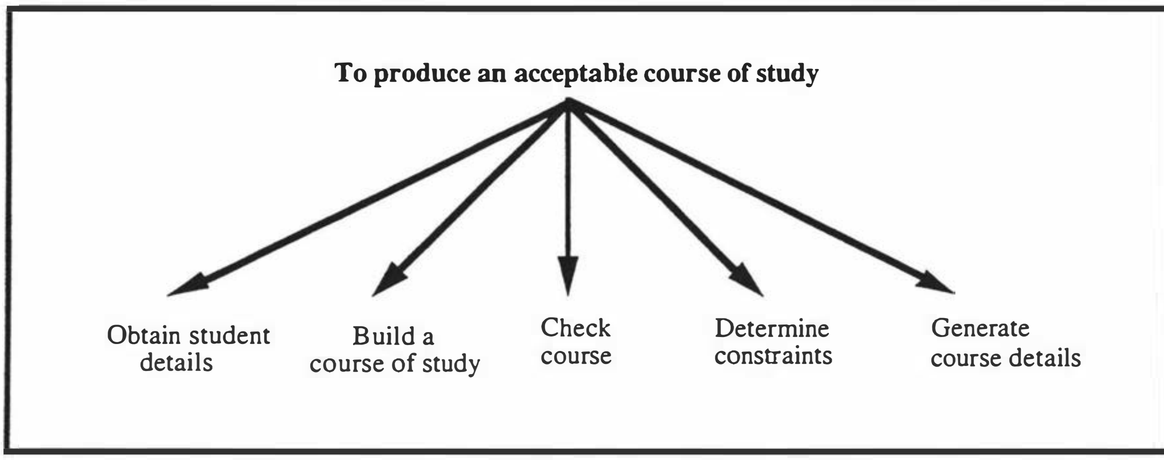


Figure 6.9 Task model

6.5 Conclusion

The initial analysis of the advisory enrolment system was completed using the FOCUS approach. There was little iteration between the four stages. A major source of information was overlooked during *Problem specification* but this was easy to rectify during the *Preliminary analysis* phase. All the information needed to specify the knowledge base was collected and described in the main outputs of the initial analysis, the organisation model and user issues. These, in line with the FOCUS approach, were progressively extended to allow for the iterative nature of knowledge base development. The first four stages of FOCUS permitted the knowledge engineer to concentrate on meeting organisational and user demands. The problem solving task of such a system would be to construct a course that satisfied the university's requirements. This would be done in conjunction with the student who could be provided with advice when necessary. By the nature of the application, students might use such a system only once. The expert system would have to be easy to use and provide explanations at many points.

Chapter 7

The FOCUS Framework Applied: Detailed Analysis

The results of the *Detailed analysis* stage of the enrolment system are reported in this chapter. Further analysis of the domain and user needs is described. The various components of the conceptual model (the models of expertise, communication and user requirements) are defined. The chapter concludes with a review of FOCUS.

7.1 Detailed Analysis

During *Detailed analysis* the knowledge engineer should complete the elicitation of domain knowledge, determine which user models should be developed and produce the three components of the conceptual model, that is the model of expertise, the model of communication and user requirements. Whilst a complete stage has been dedicated to examining the circumstances of users, more detailed analysis of users may be needed to complete the conceptual model (for example to ensure that the user models contain all relevant information).

With the enrolment problem, further elicitation of domain knowledge, principally from documents but also from the Dean of the Science Faculty, was necessary before the models could be completed. Users were also interviewed to find out how they currently tackled enrolment at Massey University. This was done to avoid the cognitive dissonance that might occur if a knowledge-based system took a completely different approach from that familiar to students.

7.2 Further analysis of user needs

Once the scope of the enrolment problem had been determined, much time and effort went into clarifying user requirements. The course approval process that takes place in February provided the ideal opportunity for contacting newly arrived students and ascertaining their views on enrolment. It was also possible to find out what problems arose in practice when dealing with the requirements specification, students were interviewed to see how they set about filling in their enrolment form. This was done to ensure that any system that was developed met their cognitive needs.

7.2.1 Survey of new students

Whilst the views of existing students were obtained in the large-scale survey (section 6.3.1), no investigation had been undertaken into the needs of students just arriving at university. Consequently, a small scale study of first year Science students was carried out during the course approval process. Thirty-seven students filled in the same questionnaire as before (see Appendix I1 for the data). The first aim of this survey was to determine the computing background of new arrivals. The results of the previous study showed a remarkably computer literate population and it was necessary to see whether this held for newcomers to Massey University. Secondly, it was important to see whether their needs differed significantly from those of the more well-established students. Thirdly, these students had just been through the process of enrolling for the first time and could provide immediate feedback about their experiences. The results of the three most relevant sections to these students were analysed: computing background, evaluation of sources of assistance and required computing facilities.

Thirty-seven students (15 female and 22 male) completed the questionnaire. Only 30% of them had used a Macintosh but 70% had experience with another PC. The most frequently used piece of software was a word processing package (73%). Fifty-one percent of the students had used a spreadsheet, 49% a programming language and 46% a database. Nearly all these figures are quite different from those in the large scale study (see Tables 6.7 and 6.8) because the students had not yet had the opportunity of taking a first year paper in computing. The two figures that are closest to those obtained previously are for PC usage (70% versus 74%) and word processing experience (73% versus 87%). It can be concluded, therefore, that new students are not arriving at university completely ignorant of computing technology and could be expected to use a computerised system. Over a half of them showed no preference for either of the interface alternatives but 43% chose windows and 5% text. Of those who did express a preference, 89% selected windows, a figure very close to that in the 1992 study (86%).

The results for question 23 (Would you use a computerised system?) were remarkably similar to those from the major survey (Table 7.1). In both cases just over half the students stated that they would definitely use a computerised system.

	Existing students	New students
No	4%	8%
Possibly	45%	41%
Yes	51%	51%

Table 7.1 Analysis of intended usage of a computerised system

With regard to the features a computerised system could offer, the results were again virtually identical to those for the existing students (Table 7.2). What was most wanted was information about papers and what was least sought were explanations of terminology. This seemed a little surprising in view of students' inexperience with university procedures. Note that in the comparisons shown in Table 7.2 the figures were derived, in each case, from over 90% of the respondents.

It is also interesting to analyse the results regarding the usage of the different sources of assistance (Tables 7.3 and 7.4). The comparison shows that the first four sources are identical for each group. This is not completely surprising since all students receive an enrolment pack with a faculty handbook. What is noteworthy is that even at this stage the new arrivals relied on other students as a means of information gathering. The other differences between the groups are easy to explain. The new students are much more likely to have met staff from the Liaison Office whereas the reverse is true for students within the system who have the opportunity to talk to lecturers. The situation is not very different when the evaluation of sources is considered. There are only two medians that are not identical: that for the faculty handbook (down from 6 to 5) and that for the staff of the Liaison Office (up from 4 to 5). The latter can again be explained by the assistance that the new students have received in the previous few months from Liaison Office staff. The former, presumably, is a reflection of the fact that students choosing a course from scratch do not find a faculty handbook quite so helpful as students within the existing system. Other students seemed to be the preferred mode of assistance, with a mean of 5.8.

	Existing students		New students	
	Mean	Median	Mean	Median
Explanations of terminology	5.2	5	5	5
Information about papers	6.3	7	6.2	7
Planning the current year's course	5.6	6	5.7	6
Planning for the future	5.6	6	5.7	6
Cost of paper	5.7	6	5.6	6
Examples of courses	5.4	6	5.5	6
Career Information	5.7	6	5.9	6

Table 7.2 Comparison of enrolment system preferences for existing and new students

	Existing students		New students	
Enrolment Pack	408	97%	35	95%
Faculty handbook	371	88%	34	92%
Students	359	85%	31	84%
University Calendar	307	73%	27	73%
Lecturers	305	72%	18	49%
Departmental handbook	277	66%	24	65%
Staff off Liaison Office	245	58%	27	73%
Professor	228	54%	19	51%
Dean	223	53%	17	46%

Table 7.3 Comparison of usage of information sources for existing and new students

7.2.2 Student problems

Since members of staff take part in the course approval process, it was easy to observe the problems faced by Computer Science students during enrolment in 1993. These included:

	Existing students		New students	
	Mean	Median	Mean	Median
Faculty handbook	5.2	6	5.1	5
Enrolment pack	5.2	5	5.3	6
Students	5.2	5	5.8	6
Lecturers	4.9	5	4.8	5
Department handbook	4.9	5	5.0	5
Professor	4.4	5	5.3	5
Dean	4.3	5	4.9	5
Staff of Liaison Office	4.3	4	5.0	5
University calendar	4.2	4	4.1	4

Table 7.4 Comparison of assessment of information sources for existing and new students

- students who wanted to know where to collect the course approval forms;
- students whose course approval form was lost and had to write their course on a piece of paper;
- students who wanted to know where else to go once they had been given approval for computing papers;
- a student who asked a member of staff to work out his timetable;
- a student who had just decided to take a BSc in Computer Science but had no idea what else to study (and had to pick six papers during the two day course approval period);
- students with cross credits trying to work out an appropriate course of study;
- students wanting to take an introductory paper in computing enrolling in the wrong paper;
- students trying to enrol in too many papers in an attempt to complete their degree in just one more year;

- a student who (as a result of advice given the previous year) had not got an essential pre-requisite;
- a veterinary science student with 5 paper exemptions who had to choose 5 other papers.

7.2.3 Task analysis

It was decided to talk to some Computer Science students informally to see how they went about constructing a course. This was to avoid problems that could arise if the computerised system was not consistent with the user's model of enrolment. New students would not have such a model it is true but the system should still help them to form a consistent view of the process. Students were asked how they set about filling in the enrolment form. They were also asked what factors (the subject, the lecture time, faculty regulations, the examination timetable, etc) were important when they were selecting papers.

Students usually started their planning by reading the appropriate faculty handbook to see what papers were available. Further information about possible papers was obtained by talking to students, relatives and university staff. A provisional lecture timetable would be written down to check whether there were clashes. Many students also wrote out their examination timetable to see if there were clashes or to check that the timetable looked reasonable (for example, they did not have four examinations in two days). Some well-organised students also tried to plan for other years and make sure that they could take papers of interest in the future.

Students picked their majoring papers first and then went on to fill in their other papers. Since Computer Science students have a fixed course at first and second year, they only have to pick from alternatives at third year. At this level, students chose from the computing subjects on offer those that interested them most and that they felt capable of doing. As for selecting other papers, some students were also enrolled in a second major and picked the appropriate second and third year papers. Other papers were chosen for a variety of reasons such as personal interest, to fulfil the faculty regulations or because they supported the major. A timetable clash was a problem unless students believed that one of the subjects was easy and missing lectures would cause few problems. Some students looked at the times of lectures whilst others tried to keep a day free from lectures, if possible. Examination clashes were a concern for most and all but one student tried to avoid them. Other factors that were taken into account included the staff who lectured on the paper, the percentage that could be obtained by internal assessment and the value of the final examination.

7.2.4 Student profile

An advisory system should provide students with information about typical courses of study. These could be produced by running a program against the university database to produce profiles for students taking the various majors. To check that such an approach was feasible, twenty first year Computer Science students were selected at random from the department's records and their choices recorded (see Appendix I2). Of the twenty students, all were enrolled in a first year Mathematics paper, thirteen in a Physics paper, ten in a Statistics paper and seven in Psychology. It was possible by looking at frequencies to come up with a typical first year course of study:

24101	Physics 1(A)
24102	Physics 1(B)
59111	Programming Fundamentals
59112	Introductory Computer Science
60101	Introductory Calculus
60102	Algebra and Geometry
61100	Principles of Statistics
or	
75102	General and Applied Psychology

Such information can be used in various ways. Advice about subjects typically taken by Computing students such as Maths and Physics can be given. Furthermore, students who have planned most of their first year course but are still a paper short can be told that the usual option in this situation is a Psychology paper.

7.3 Domain analysis

The domain analysis essentially involved obtaining the relevant information about the requirements for graduation in the Science faculty. This is not a straightforward matter as there are both general Science faculty regulations as well as the specific requirements for each major. Once this knowledge had been elicited, the Dean of the Faculty of Science was interviewed to ensure that the knowledge engineer had correctly identified the problem areas and possible solutions.

University regulations

The knowledge-based system would have to check that a student's selection of papers met the relevant criteria. Checking occurs at two levels, to see whether a student is permitted to enrol in a selected paper and to ensure that the completed course of study meets both faculty and university requirements. Most of the relevant information that the knowledge-based system would use for checking that a student fulfils all the requirements can be obtained from the University Calendar and Timetable (Massey University, 1993a; Massey University, 1993c): the faculty regulations, the paper number, title, paper contents, point value, number of lectures, number of laboratories, timetable details, examination date and the prerequisites, corequisites and restricted papers. There is one exception to this; there is a note in the Calendar which states that the department of Computer Science has to be consulted for pre-1991 papers that are regarded as equivalent for prerequisite and restriction purposes.

Given all this data, it was possible to determine the principal requirements for all the majors in the Science faculty. In Computer Science, for example, students must take two papers at first year, three at second year and the equivalent of three at third year. A student is able to take nine more papers within the faculty and four outside. Choice is not completely unrestricted. Second year courses can usually only be taken when the appropriate prerequisite has been passed. The same principle applies at the third year. It can be seen that this system is very flexible and allows students a considerable amount of freedom to tailor a degree to suit themselves.

Science Faculty Handbook

Some of the advice that students need can be found in the Science Faculty Handbook (Massey University, 1993b). The introduction to the Science Faculty Handbook tells students that flexibility and freedom of choice are hallmarks of Massey University's science degrees. Advice is also given about planning a degree (see Appendix I3). To make enrolment a little easier for students some sample BSc degrees are included. The one for Computer Science is as shown in

Figure 7.1. It is interesting to see how close the first year of this course is to the sample course generated from the student records. Seven of the courses are identical; the only exception is 23101 which is replaced by 75102. The advice given in the handbook seems to be heeded.

Finally, for each of the areas that can be studied, there is a brief description of what the discipline involves and what career opportunities are available to those who qualify in the area. Suggested courses are also specified although these are often vague.

	First year		Second year		Third year
Majoring requirements	59111		59211		59311
	59112		59212		59312
			59213		59313
Choice	23101		24240		24342
	24101		34205		60203
	24102		60208		60204
	60101		61240		
	60102				
	61100				
Points	112	+	106	+	84 = 302

Figure 7.1 Sample course for Computer Science

Departmental Publications

Whilst much of the faculty material, such as sample courses, can usefully be incorporated into an advisory module of the enrolment system, there is little detail about papers. For this, individual department publications have to be consulted. In the Computer Science Department, the paper controller provides an outline of what taking a paper involves. The outline for 59324 is shown in Appendix I4.

Dean of the Faculty of Science

In an interview, the Dean of the Faculty of Science discussed the enrolment process and clarified faculty related issues (for a complete account of the interview see Appendix I5). He agreed that there were problems with enrolment. Despite all the time and effort put in by staff before and during the course approval process, he and his secretary spend much of their time in the first month of term one sorting out problems with students who wanted to change papers. Moreover, many students who enrol for a Science qualification have changed from other degrees (for example BVetSci) and need much assistance to choose a major and build up a course of study.

It became obvious during the interview that if the pilot system was to be extended to other students within the Science Faculty, the Dean of Science would have to be interviewed at much greater length to elicit his expertise in counselling students. He has always tried to assist students to build a coherent course of study and, when necessary, suggests papers relevant to the major that a student has not considered. Over the years he has built up a picture of what other papers can be usefully studied in conjunction with majoring ones. Moreover, he is able to provide sample courses for all the majors. Physics students, for example, typically only study Mathematics, Statistics and Computing papers in association with their major. Information of this kind is invaluable if an advisory system is to offer the same quality advice available from the Dean.

7.4 Conceptual model

It is now possible to describe the proposed functions of the knowledge-based system. Students enter their identification details and confirm the qualification for which they are enrolled. Then, students can either enter paper numbers themselves or select them from those proposed by the system. The papers presented by the knowledge base will depend on the prerequisites passed by the students and/or their areas of interest. Any paper for which a student has not passed a prerequisite or has already taken an equivalent will automatically be ruled out at this stage. The complete course of study will then be checked and if there are problems these will be brought to the attention of the students. Those students whose course of study does not meet the regulations will have to make alterations to it, again with some assistance from the system if required. The detailed check of the course of study ensures that lecture and timetable clashes as well as problems with corequisites are brought to light. Any violations of faculty regulations will also be identified. Moreover, students who want to complete their degree in that year can also check that the course will meet the faculty requirements if all papers are passed.

This is an advisory knowledge-based system that offers assistance in various ways. If a student does not know what papers to enter, suggestions can be made as appropriate about the major, other faculty papers and interest subjects. Information about the contents of these papers can be obtained. Advice can be related to a student's interests or to degree stereotypes. For instance, a Science student can be told that Psychology and Critical Thinking are popular choices at first year. Once a course has been checked, the constraints identified allow the system to propose solutions that do not violate the restrictions. For instance, if a student is not allowed to take any more ex-faculty papers, that option will not be pursued and only faculty papers suggested.

7.4.1 Model of expertise

The domain layer of the model of expertise can now to be completed. It already contains the lexicon, concepts and the principal relationships between them. The definitions can be used as support knowledge whenever students request clarification of terminology. Normalised relations derived from the concepts, attributes and relationships already identified can be included in the domain layer. They can easily be translated at the design stage into a form suitable for computerisation. To show that all necessary information is included in the relations a brief example is worked through. Take the case of David Clarke who wishes to enrol in 59311, 59312 and 59313, *inter alia*. The Student relation holds details such as his name and identification number whilst information about the qualification for which he is enrolled is available from the Student Qualification relation. The Student-history relation shows that he has passed both first and second year papers. When he tries to enrol in the paper 59311 checks have to be made against the Paper-prerequisite, Paper-corequisite and Paper-restriction relations. In this case both prerequisites and restrictions apply. David Clarke has passed the prerequisite paper 59211 and has not taken the restricted paper 58301 so would be allowed to proceed with the enrolment. The lecture times for 59311 are obtained from Timetable whilst the period when the paper runs, the examination date and point value come from Paper. Finally, when David has entered all his computing papers, a check can be made to see that he meets the requirements for the Computer Science major in the Science faculty. Paper-major is consulted at this point. It can easily be seen that all the computing papers he has taken or wishes to take are in the regulations for the Computer Science major.

Student

Attributes: student number, student family name, student given name

Student_Number	Student_ FName	Student_ GName
88546644	Clarke	David
89783991	Johnson	Mary
90652133	Stewart	Jason

Student-qualification

Attributes: student number, qualification, major

Student_Number	Qualification	Major
88546644	BSc	Computer Science
88546644	BSc	Physics
89783991	BSc	Computer Science
90652133	BSc	Computer Science

Timetable

Attributes: paper number, room, time, day, type of activity (lecture, tutorial laboratory session)

Paper_Number	Room	Time	Day	Type
59311	SSLB1	10am	Monday	Lecture
59311	SSLB1	11am	Monday	Lecture
59313	SSLB4	9am	Monday	Lecture
59313	SSLB2	2pm	Tuesday	Lecture
59313	SSLB3	2pm	Thursday	Tut
59326	SSLB5	3pm	Monday	Lecture
59326	SSLB6	4pm	Friday	Lecture
59327	SSLB3	5pm	Tuesday	Lecture
59327	SSLB3	8am	Wednesday	Lecture

Student-history
Attributes: student number, year course taken, paper number, grade

Student_Number	Year	Paper	Grade
88546644	1993	59211	A
88546644	1993	59212	A-
88546644	1993	59213	A
88546644	1993	34103	B+
88546644	1993	24210	A-
88546644	1993	24202	B
88546644	1993	24201	B-
88546644	1992	59112	B
88546644	1992	59111	B+
88546644	1992	24101	B
88546644	1992	24102	B+
88546644	1992	75102	A
88546644	1992	60102	B-
88546644	1992	60104	B

Paper
Attributes: paper number, name (abbreviated), points value, examination date, session (morning or afternoon) and when paper runs during year - all year (AY), first half (FH) or second half (SH)

Paper_Number	Paper_Name	Points	Exam_Date	Session	Runs
59110	Intro. to Scientific Comp.	14	4/11/93	am	AY
59111	Programming Fund.	14	5/11/93	pm	AY
59112	Intro. to Comp. Science	14	4/11/93	am	AY
59113	Intro. to Computing	14	5/11/93	pm	AY
59211	Algor. and Data Struct.	14	4/11/93	pm	AY
59212	Data Management	14	29/10/93	pm	AY
59213	Computer Systems	14	2/11/93	am	AY
59311	Programming Languages	16	28/10/93	am	AY
59312	Software Eng. Techniques	16	2/11/93	pm	AY
59313	Comp. Arch. and Oper. Sys.	16	8/11/93	am	AY
59324	Knowledge-based Systems	8	6/11/93	am	FH
59325	Computer Networks	8	19/10/93	pm	FH
59326	Database Systems	8	21/10/93	am	SH
59327	HCI	8	1/11/93	am	SH

Paper-major
Attributes: paper number, qualification and major

Paper_Number	Qualification	Major
59111	BSc	Computer Science
59112	BSc	Computer Science
59211	BSc	Computer Science
59212	BSc	Computer Science
59213	BSc	Computer Science
59311	BSc	Computer Science
59312	BSc	Computer Science
59313	BSc	Computer Science
59324	BSc	Computer Science
59325	BSc	Computer Science
59326	BSc	Computer Science
59327	BSc	Computer Science

Paper-restriction
Attributes: paper number and number of restricted paper (r_paper)

Paper_Number	R_Paper
59111	58103
59111	59113
59211	58201
59311	58301

Paper-corequisite
Attributes: paper number and number of corequisite paper (c_paper)

Paper_Number	C_paper
60203	60102
60203	24102
60303	60204

Paper-prerequisite
Attributes: paper number and number of prerequisite paper(p_paper)

Paper_Number	P_Paper
59211	59111
59211	59112
59311	59211
59324	59212

A check has to be made, when developing the domain layer, that any relevant user models for problem solving identified in the user requirements have been included. There was no problem with the long term user model, that is the student information kept from session to session. The Student, Student-qualification and Student-history relations had already been specified. This was not the case, however, with the short term student model (that is information useful for one session only). This includes subjects of interest (for when students need advice) and whether a student expects to complete a degree in a particular year. The following relations can hold this information:

Student-completion
Attributes: student number, completion response

Student-interest
Attributes: student number, subject

The domain layer can also contain other kinds of information; it is not restricted to relations. Any rules in a system that are needed for problem solving or explanation purposes can be specified. The regulations for completing a degree in the science faculty regulations can be summarised as follows:

study for a minimum of 3 years;

obtain 300 points in total;

obtain 84 points minimum from first year papers;

obtain 140 points maximum from first year papers;

obtain 160 points minimum from second and third year papers;

obtain 112 points maximum from second year papers;

obtain 56 points maximum from papers taught by other faculties;

obtain 48 points from third year majoring papers;

meet requirements of major;

meet necessary prerequisites, corequisites and restrictions.

Other more general requirements for a course are that no examination clashes are permitted and only in unusual cases can a course of study exceed 120 points.

Some information is also required for advising students. Sample courses, paper outlines and statistics for previous years may all be made available. This does not need to be structured into relations, however, as it is not needed for problem solving but for illustrative purposes only. There is no requirement, therefore, for a paper-lecturer relation since the information about who lectures on a course would be available on the paper outline. Any advice that is given to students has to take account of both the short term and long term student models as well as the faculty regulations and any constraints identified after checking a course of study. These restrictions will be course specific but will be based on the rules identified above.

Inference level

The inference layer describes how information in the domain layer can be used. It is developed in parallel with the task model to ensure that no important function is omitted. Inputs and outputs are depicted by rectangles and actions by ovals. Using this notation, a generic template for a construction task has been developed (Figure 7.2).

The construction task involves the building of an artefact by a person in the light of system constraints. The set of constraints is represented by the system model. The appropriate constraints are selected to match the requirements of the individual's case description. Note that there are several instances of "case description". For each occurrence, the contents are identical. The duplication is needed to retain the clarity and simplicity of the diagram. Where the contents are different, a number is appended (see parameters 1 and 2). The inference structure diagram can be given a textual equivalent (Figures 7.3 and 7.4).

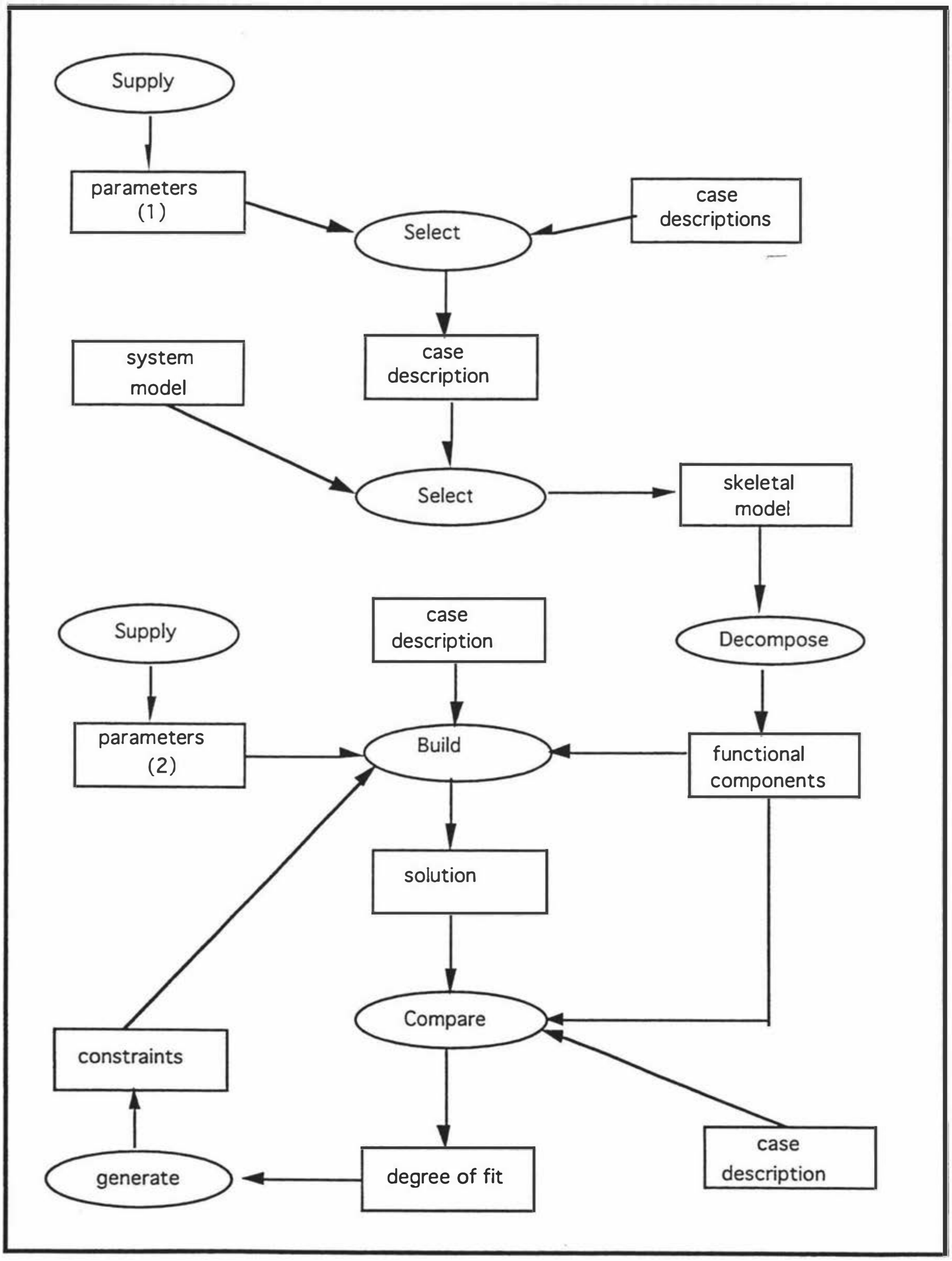


Figure 7.2 Inference layer template

Supply parameters (obtains values from students)
primitive inference action: supply
output
    parameters 1: student name and student identification number

Select case description
primitive inference action: select
input
    parameters 1: as before
    case descriptions: student, student-qualification and student-history relations
output
    case description: information about a student from the above relations

Select appropriate skeletal model
primitive inference action: select
input
    case description: as before
    system model : all university information
output
    skeletal model: relevant faculty information

Decompose skeletal model into functional components
primitive inference action: decompose (changes a structure into a set of instances)
input
    skeletal model: as before
output
    functional components: faculty regulations, paper information, course stereotypes

Supply parameters (obtains values from students)
primitive inference action: supply
output
    parameters 2: paper number, degree completion response, subjects of interest

Figure 7.3 Textual description of enrolment inference layer (1)

Build solution in accordance with paper regulations

primitive inference action: build (assembles a set of tested instances into a structure)

input

parameters 2: as before

case description: as before

functional components: as before

constraints: limitations on course (output by the "generate" primitive inference action. These do not apply first time through.)

output

solution: completed course of study, explanations about why papers not acceptable

Compare course

primitive inference action: compare (checks regulations are met.)

input

solution: as before

case description: as before

functional components: as before

output

degree of fit: the difference between the course and the regulations,

Generate the constraints

primitive inference action: generate (generates the constraints on the course of study)

input

degree of fit: as before

output

constraints: as before

Figure 7.4 Textual description of enrolment inference layer (2)

It is helpful at this point to re-draw the inference structure diagram to show the specific inputs and outputs (Figure 7.5).

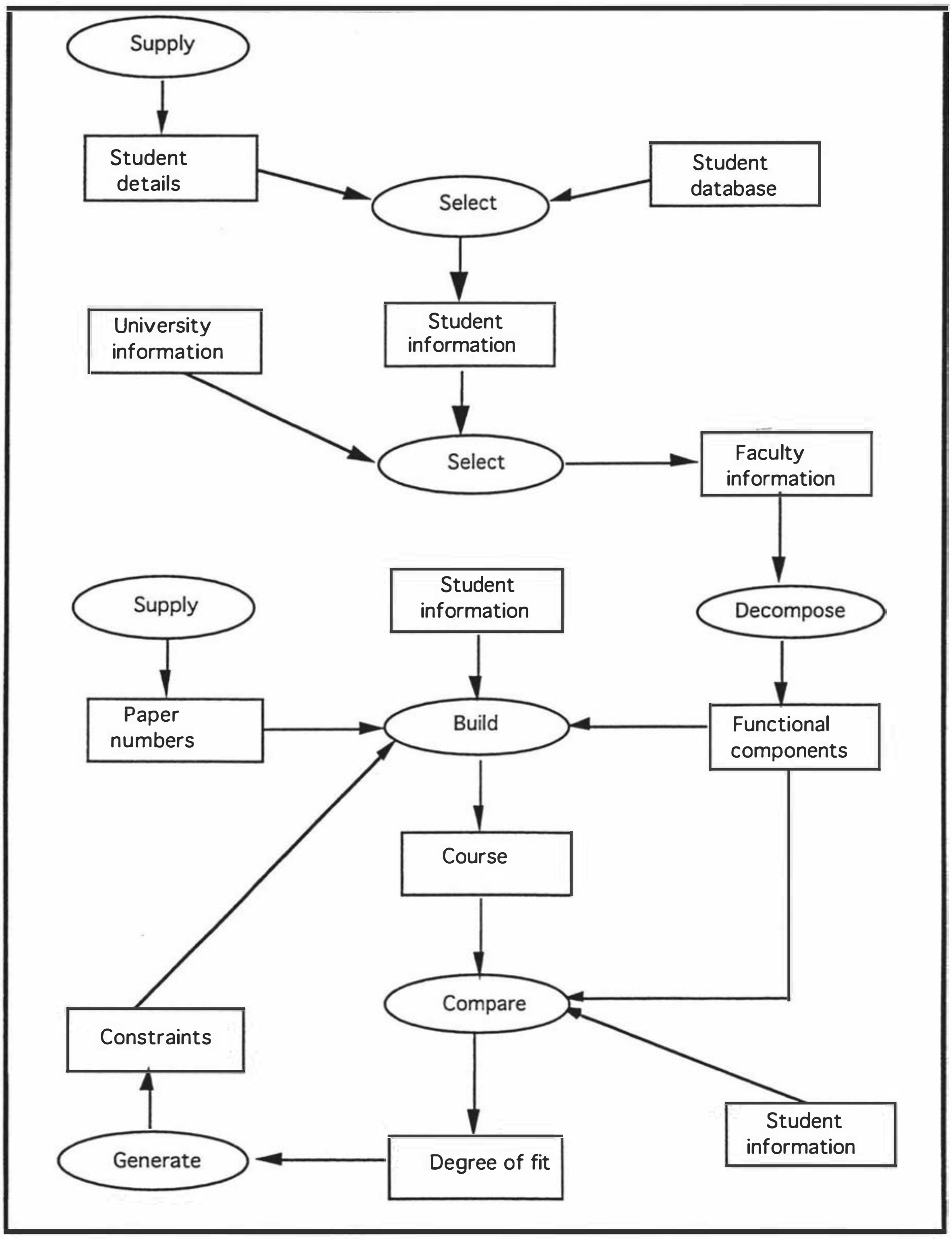


Figure 7.5 Inference structure diagram for student enrolment

The system model refers to the university's requirements, the skeletal model, that set of university requirements that pertain for a particular faculty and the case descriptions to the student information held in the university database.

From the details a student enters, it is possible to select the appropriate student history. As this includes the qualification for which a student is enrolled it allows for the appropriate faculty information to be accessed. Not only paper details but regulations and stereotypic courses (used to advise students) can be retrieved. A course of study is built up either by the student entering paper numbers or by choosing from papers selected by the system. Once a course of study is completed, the system will check that the faculty regulations have not been violated. If they have, the constraints are identified, for example, "You cannot take 75101 as you have already taken four ex-faculty papers". Subsequently, a student has to return to building up a course or can merely confirm the course with the offending paper(s) removed.

The above specification caters for the situation where a student is given no assistance with the selection of paper numbers. If an advisory level is to be incorporated, any assistance given to students to help them select papers must take account of relevant faculty information and any identified constraints. This can be shown in the following inference structure diagram (Figure 7.6) which is a decomposition of "Supply paper numbers".

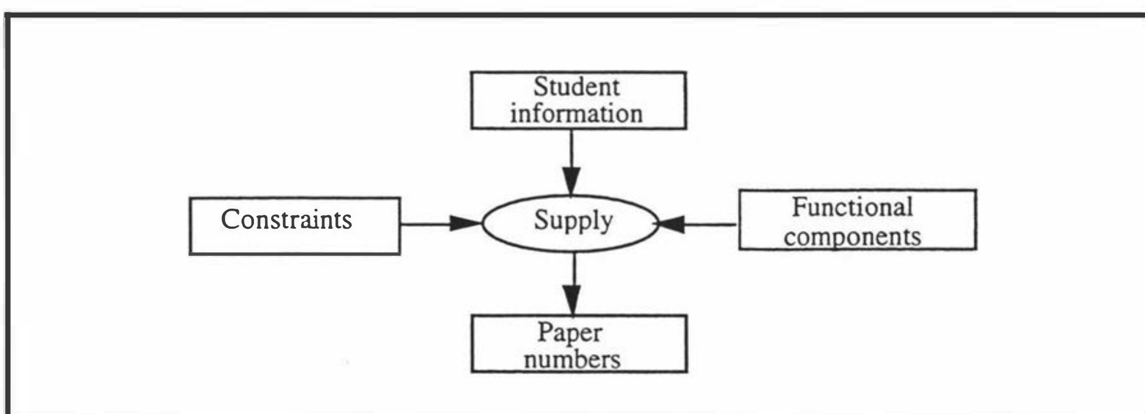


Figure 7.6 Decomposition of Supply Paper Numbers

The task structure

The task layer is a specification of the problem solving capabilities of the system which can show sequence and iteration. This is an important input to the design stage and can be used to help with the design of the knowledge-based system. The task layer below follows the instantiated version of the inference structure.

The task layer is specified as follows:

To produce an acceptable course of study do

obtain student name and identification numbers

select student, student-qualification and student-history relations in accordance with details supplied

select faculty information that matches student qualification

decompose faculty information into its components (regulations, paper information, course stereotypes)

repeat

obtain paper numbers

build course of study

compare course of study with student details and faculty information to determine problem area

generate constraints from degree of fit

until course of study acceptable

Compare course

A more detailed description of this process is required here for checking that a course of study meets the requirements of the faculty. The pseudocode for this process in the Science faculty is given below. Note that a thorough check is made if a student wishes to complete a degree in the current year.

For each paper

check corequisites met

For current course of study

if finish degree student has to meet the following requirements:

studying for at least 3 years

84 points minimum first year

160 points minimum at second and third year

no more than 56 points in subjects outside the faculty

at least 300 points in total

majoring requirements for subject area satisfied

at least 48 points at third year from majors

else

no more than 140 points at first year

maximum 112 points at second year

no more than 56 point from ex-faculty papers

majoring requirements satisfied

any general requirements for appropriate year satisfied

Obtaining advice

If a student requires help when selecting a paper number and requires information about course stereotypes or papers, the following algorithm could be formulated:

```
if information required about specific paper
        provide information
else if course information required
        show course stereotype
else if general paper assistance required
        if major
                show papers eligible to take
        else if faculty
                find out subject area of interest
                show papers eligible to take
        else if ex-faculty
                find out subject area of interest
                show papers eligible to take
        if student requests information about specific paper
                provide information
```

7.4.2 Communication model

A model of communication has to built for every user group. In this case there is only one group of users and hence one model of communication. The task model has to be extended first to show the important tasks and subtasks. It is necessary to move from the generalised task model of stage 4 (section 6.4.2) to a more detailed diagram which shows user actions such as confirming a course of study, providing student details and supplying paper numbers. A first attempt is shown in Figure 7.7.

This was subsequently revised to show more clearly how the system and users would interact. "Check qualification details" has been brought to the top level and "Build a course of study" revised. Each node can be annotated with a "U" for a user activity and an "S" for a system one. The top two levels of the model of communication are shown in Figure 7.8.

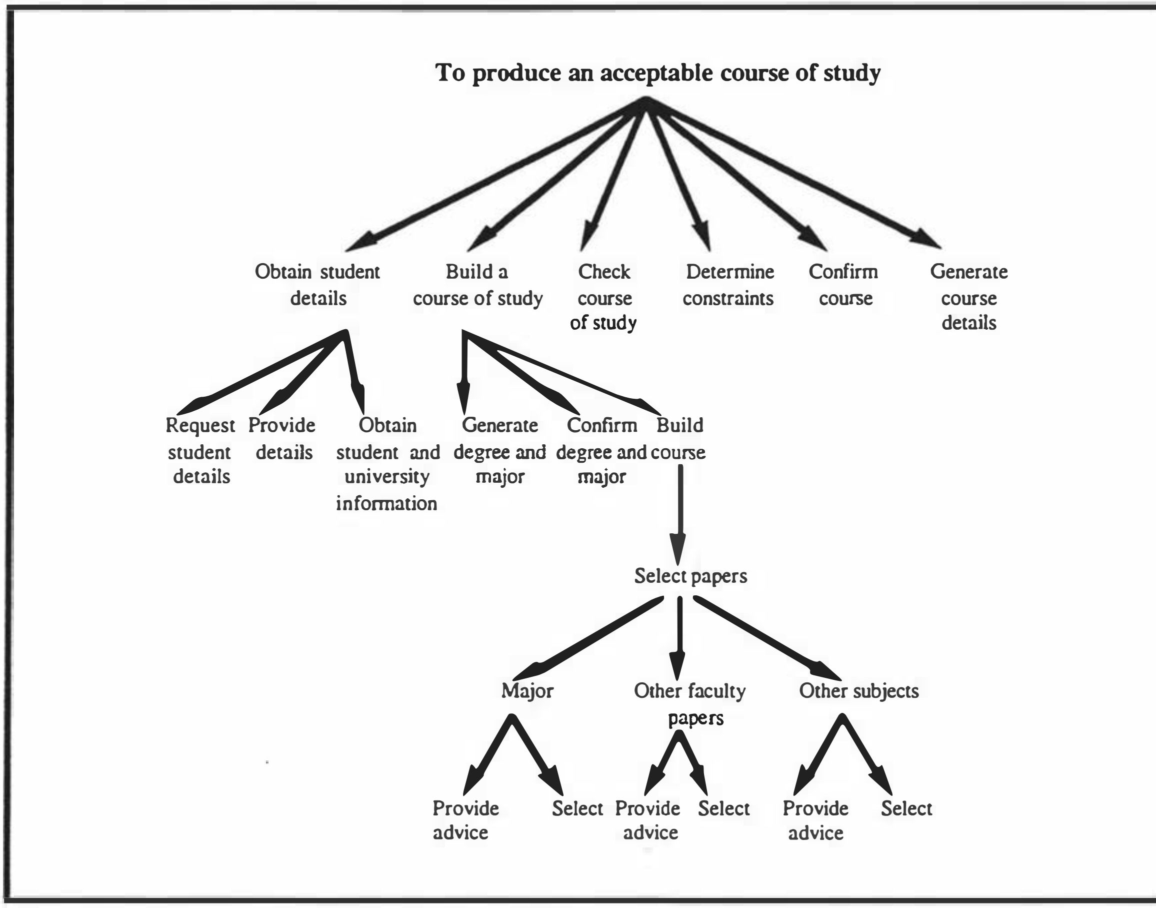
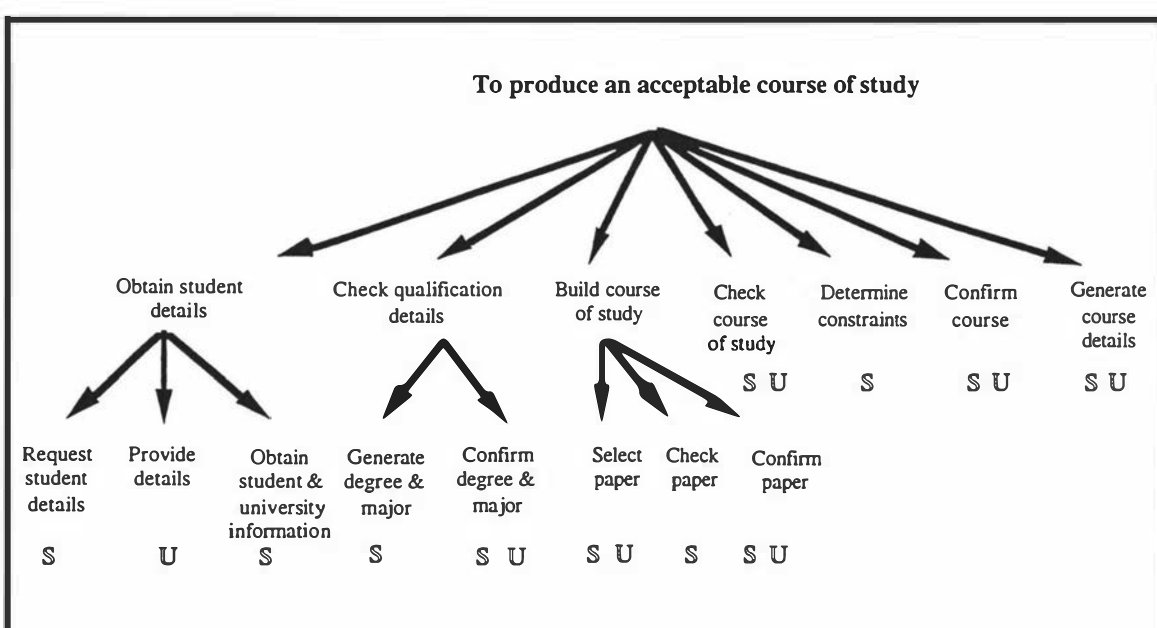


Figure 7.7 Developing the model of communication



Key
S = system
U = user

Figure 7.8 Model of communication

7.4.3 User requirements

From the information gathered from the questionnaires, it was possible to obtain the students' perspective on enrolment. The large and small scale surveys of student views indicated that they were willing to use an advisory enrolment system. Of particular importance was the availability of information about papers students were planning to take. To find out, though, how students went about the task of enrolment itself, interviews were held (section 7.1.3). Once this was done the user task model could be built.

The user task model is, in this case, a stereotypic one that describes how students typically deal with enrolment. For students the task involves finding out what papers are available, getting further information about them, making an initial choice, writing out a provisional lecture and examination timetable and resolving any problems that arise because of clashes. The course of study selected is then entered onto the enrolment form. Papers are chosen to meet the majoring requirements, to fulfil the general faculty regulations or for interest.

When developing the associated model of user knowledge, it was assumed that students did not have a detailed understanding of the enrolment process. This conclusion was based on the responses in the surveys. Firstly, students accessed many sources of information when enrolling which implied the need to obtain further information. Secondly they specifically requested assistance when planning their course of study. Thirdly, definitions of terminology whilst not a high priority for the students were still seen as a useful option. In these circumstances, the stereotypic model of user knowledge indicates that students know little of the domain.

Both long term and short term user models are needed for problem solving. The contents of the long term model have already been specified in the domain layer, that is the Student, Student-qualification and Student-history relations. The short term model holds information needed for the current session, that is the subjects that a student is interested in, whether the student intends to complete the degree in that year and any constraints which delimit their existing choices. The first two are supplied as parameters to the system and the latter generated when checking a course.

Explanations of three kinds have to be provided: definitions, support and task knowledge (see section 2.1.4). The model of user knowledge indicates that the typical student should be given access to definitions of terms such as prerequisite and corequisite. This is very straightforward as these definitions are already in the domain layer. On the basis of student needs elicited from the survey and interviews, the following information (that is, support knowledge) about a paper should be made available to students: paper number, paper name, paper points, course contents, course assignments, course text(s), exam date, value of final examination, lecturing staff, prerequisites and corequisites. Statistics for the last few years about the number of enrolments, the percentage of withdrawals and the distribution of grades could also be provided if permitted by the university. Such information could be printed out for the student to look at during the enrolment session.

Furthermore, explanations of decisions made by the knowledge-based system (task information) are vital if students are to make informed choices about their course of study. These then should be generated automatically. They would include explanations such as the following:

- you do not have the necessary prerequisite to take this paper;
- you have not enrolled in the corequisite paper;
- you cannot take this paper as you have already passed its equivalent;
- you have an examination clash;
- you have enrolled in too many papers outside your faculty;
- you have not enough points to complete your degree this year.

Finally, a simple model of user characteristics for interface purposes will suffice; the user is clearly defined and the task performed occasionally. The users are university students who most likely (but not inevitably) have experience with computers and possess keyboard skills. They appear to be at ease with direct-manipulation and text interfaces. In these circumstances, there seems no need for refinements such as speech recognition or natural language. Given that students may wish to access definitions and paper details to help them make decisions, any knowledge-based system developed should be mixed initiative.

As a result of the preceding user modelling it is now possible to make some major decisions about the interface. Since students are familiar with filling in forms (their user view), this seems the most suitable interaction style to employ. This should help students build up an appropriate user-system image and allows a high degree of cognitive consistency with the way students carry out their task. The intention would be, though, to eliminate all the activities that make filling in a form such a problem, that is obtaining information from several sources and cross-checking it. The tedious business of writing out provisional timetables could be done by the system, part of the screen could show a timetable that would be filled in as papers were selected. Students could then easily see whether they had a clash, a free day, etc. Complete paper details could be obtained through a series of advice screens. If sample courses were needed, these too could be provided.

7.5 Discussion

There is a problem in presenting the three models in sequence. Although the parallel nature of model development has been discussed, it appears as if the model of expertise was developed first, the model of communication second and the user requirements third. This was not the case in practice as certain decisions depended upon others. The task layer could not be completed until the model of communication was finished to ensure consistency between them. Furthermore, developing the models was an iterative process. Whilst much of the domain layer was finished at an early stage, the knowledge needed for explanation and advice was added later. The model of communication was based on a task model that had to be altered many times. Some of this revision was only to be expected; the task model was originally a high level model that could only be decomposed as more detail was ascertained. It also reflected the difficulties of ensuring that the principal user-system interactions were incorporated. Finally, the three components of the conceptual model were cross-checked against the other two when necessary to ensure consistency and that no important requirement had been overlooked. Once the user's requirements for explanation were determined a check was made to see that these could be met by the domain layer of the model of expertise.

The above discussion shows clearly the iterative nature of FOCUS. Whilst it may seem tedious producing several models, the overlaps and duplication allow the knowledge engineer to check the correctness of the conceptual model. The claim has also been made that organisational, user and expertise issues can be fully integrated using FOCUS. The initial analysis for the enrolment system illustrated how the functional specification and task model for the knowledge base could be produced once the organisation model, user issues and preliminary model of expertise were in place (Chapter 6). The task model has a vital role in unifying these three sets of requirements.

It has been maintained, furthermore, that FOCUS allows important human-computer issues to be dealt with. Functions requested by students and an interface tailored to meet their needs were incorporated into the conceptual model. It is instructive at this point to compare the FOCUS framework with the process followed by other developers of enrolment knowledge-based systems. Are the outcomes any different, from a human-computer interaction perspective, than those that already appear in the literature? In order to carry out the comparison, a brief account of four such systems follows.

The Intelligent Student Adviser, ISA (Weekaroon *et al.*, 1992) supplements the face to face advice process used with undergraduates and postgraduate Management students at Sangamon State University, Springfield. It aims to provide quality and timely advice to students about core and optional courses. At the time the paper was written, the student database was not linked to the knowledge base but on-line access was planned for the future. Unfortunately, there is little detail about what occurs during a consultative session with ISA.

The Academic Planning Environment, APE, (Golumbic *et al.*, 1986) is a knowledge-based system for advising and assisting students enrolling in a course of study. It generates an initial plan for a student consistent with the university regulations, departmental requirements and the student's history. This can subsequently be modified in view of the student's wishes. Since the requirements of each department differ considerably, the knowledge base is partitioned along departmental lines.

DPLANX (Sharma, 1992) is a constraint-based planning system to assist students enrol. The search space of possible solutions is narrowed down to a degree programme which takes account of university regulations, a student's previous results and student preferences. User interaction takes place via a user constraints language which allows for papers to be included and excluded.

EASY (Chew, 1987) is a decision support expert system for advising students and checking that a course meets regulations. Essentially, it generates recommended papers for a selected speciality and allows the students to alter such a course. It has a simple user interface but the author recommends altering this to a natural language interface if the prototype was further developed.

The developers of three of these systems (and possibly all four since insufficient detail is given about ISA) take the view that a course of study should be generated for a student to modify. This is quite unlike the approach taken in the specification for the Massey enrolment system where students enter papers and then have a course of study checked. It is probable that this is the result of a difference in environment. The flexible nature of many of the degrees at Massey makes the solutions implemented in APE, DPLANX and EASY impractical. The rationale for the approach these researchers take, however, is never fully presented.

Overall, little research was conducted into organisational concerns or user requirements. No form of task analysis or user modelling appears to have been carried out. The interface is mentioned in three of the papers (again the authors of ISA failed to provide information on this point). In one case a special language was developed for inputting constraints, in a second a text interface was used (although it was suggested that later a natural language interface front end should be built) and in the third case it was mentioned that a more user-friendly environment be added. These authors developed an interface suitable for their purposes with, in two instances, the intention of enhancing it later. The interface, it has been argued in this thesis, is not just something that can be added afterwards but should be considered during the analysis phase.

Obviously, the developers of the four systems described above had their own research agenda when building their systems. Nonetheless, the differences between these approaches and FOCUS are illuminating. FOCUS has a number of features that are not incorporated into other approaches:

- Functions are specified in response to a task analysis and elicitation of user requirements.

- Users are modelled so that requirements for explanation can be provided.

- The criteria for evaluating the system are specified in the conceptual model.

- Provision can be made for other versions of the system as well as alternative interfaces.

7.6 Conclusion

Considerable analysis of the domain and computing background of students enabled user requirements to be formulated in way that suited the particular environment of the organisation. The comprehensive modelling of the user allowed the main interface presentation style to be selected and suitable explanation features to be identified. Ways in which they can be implemented will be explored at the design stage. Despite the problems of developing the three components of the conceptual model in parallel, the time spent on constructing all the relevant models was not wasted. It would have been very difficult to handle so much information without structuring it in this way. As requirements emerged during knowledge elicitation they could easily be accommodated within this framework. The generality of the approach described in the inference and task layers should be noted. Whilst the case study considers the needs of Computer Science students in the Science faculty, specifically, there is no problem handling the requirements of any faculty. Other features could easily be incorporated such as the automatic generation of fees or the ability for the system to generate a course of study (using course stereotypes, for instance). FOCUS allowed the analyst to cope with a large amount of information and determine user needs whilst also catering for organisational and domain requirements.

Chapter 8

FOCUS and Interface Design

In this chapter an extended example, based once more on the enrolment system, shows how the outputs from FOCUS can assist with the high level design of the human-computer interface. Initially, control is built into the model of communication so that activities can be sequenced and the major input and output flows identified. The sequence of application screens is then produced based on a logical grouping of these inputs and outputs. Other facilities that users would need to run the knowledge-based system are also determined. Finally, the screens themselves are designed taking into account the interface metaphor that has been selected.

8.1 Interface design in the context of FOCUS

During the design stage, the interface characteristics of the future system have to be determined. Using FOCUS, many important decisions have already been taken during the analysis stage. The users' interface requirements have been specified and the criteria for evaluation selected. To see how decisions made during analysis impact on the design phase let us consider again the enrolment application.

The specification of students' interface requirements in the detailed analysis phase was comprehensive. It was suggested in the conceptual model that a form-fill interface be selected since students have to provide the knowledge-based system with personal details and paper numbers. This, it was argued, matches the way that students naturally think about the task. Students also have to be able, however, to obtain advice and explanations when they require them. To achieve this another interface presentation style needs to be combined with form-fill. Direct manipulation features (point and click) seem most suitable for giving students fast access to the appropriate functions. The model of user characteristics indicated that most students have experience with computers and should have little difficulty either typing in information or using direct manipulation.

The results of the task analysis also have important consequences for interface development. Discussions with students indicated that most of them wanted to see what their timetable would look like. One way that this can be handled is to have a large portrait screen that can be split into two parts. The top window only will be forms based and deal with the main communication flows needed for the application, that is entry of student details. The lower window will then be free to display the timetable and provide advice about papers and faculty regulations, etc. This split screen interface is shown below (Figure 8.1).

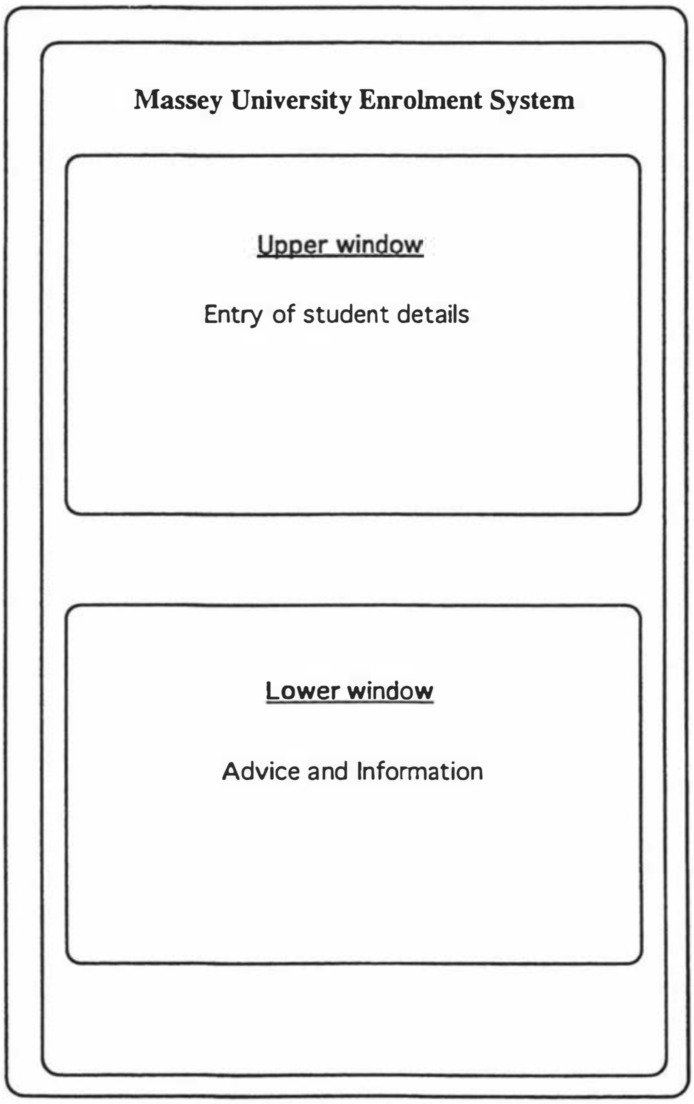


Figure 8.1 Split screen interface

Having established the overall framework, the criteria specified for interface evaluation can be used to guide detailed interface development. This should make it easier to build systems that people will find usable. Bearing in mind the criteria specified in the organisation model, the following decisions were made:

- to sequence activities in a way that the user would expect;
- to allow students to obtain definitions of terms they do not understand;
- to provide students with as much control as possible;
- to make it clear to students what options are available;
- to minimise memory load by providing Help at all points;
- to allow students to move easily from screen to screen;
- to provide users with feedback so that they can see the consequence of their actions;
- to give each screen a name that clearly indicates its function;
- to fully explain decisions about why a paper or a course of study cannot be taken.

8.2 Adding control to the communication model

One of the major components of the conceptual model, produced during detailed analysis, is the communication model. It already shows who carries out what tasks (the system (S) or the user (U)) but needs to be refined to show sequence, iteration and selection. When designing an interface, the order of events is very important for screen design. Sequence is shown in the extended communication model by numbering the activities from left to right. Following the practice of structure charts (Yourdon, 1979), iteration is indicated by an arc and selection by a diamond. Figure 8.2 shows the top level of the extended communication model with iteration around steps 3, 4 and 5, that is from "Build course of study" to "Determine constraints". The expanded tasks 1 and 2 are shown in Figure 8.3. The remainder are in Appendix J.

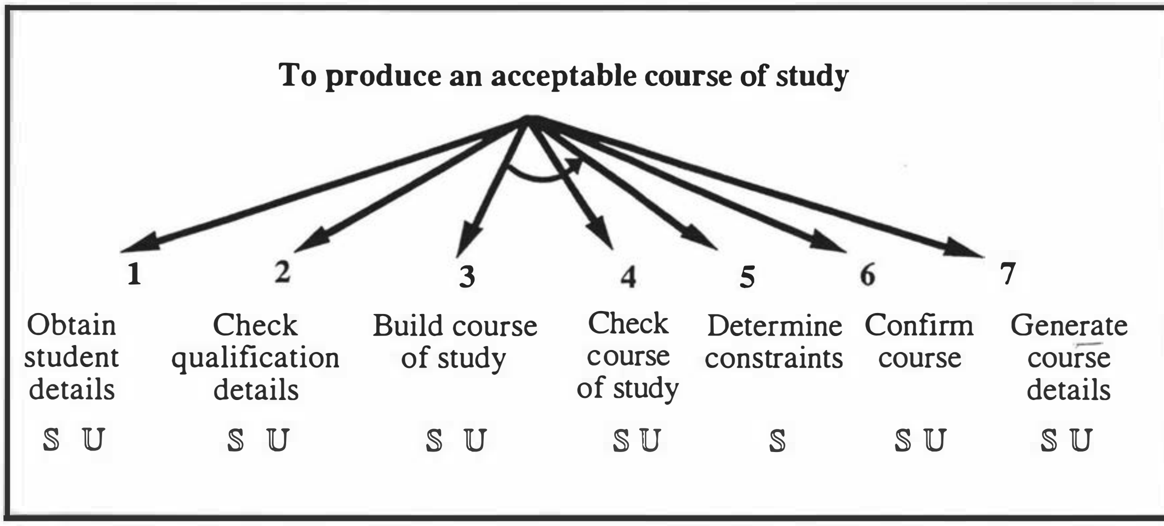


Figure 8.2 Top level of extended communication model

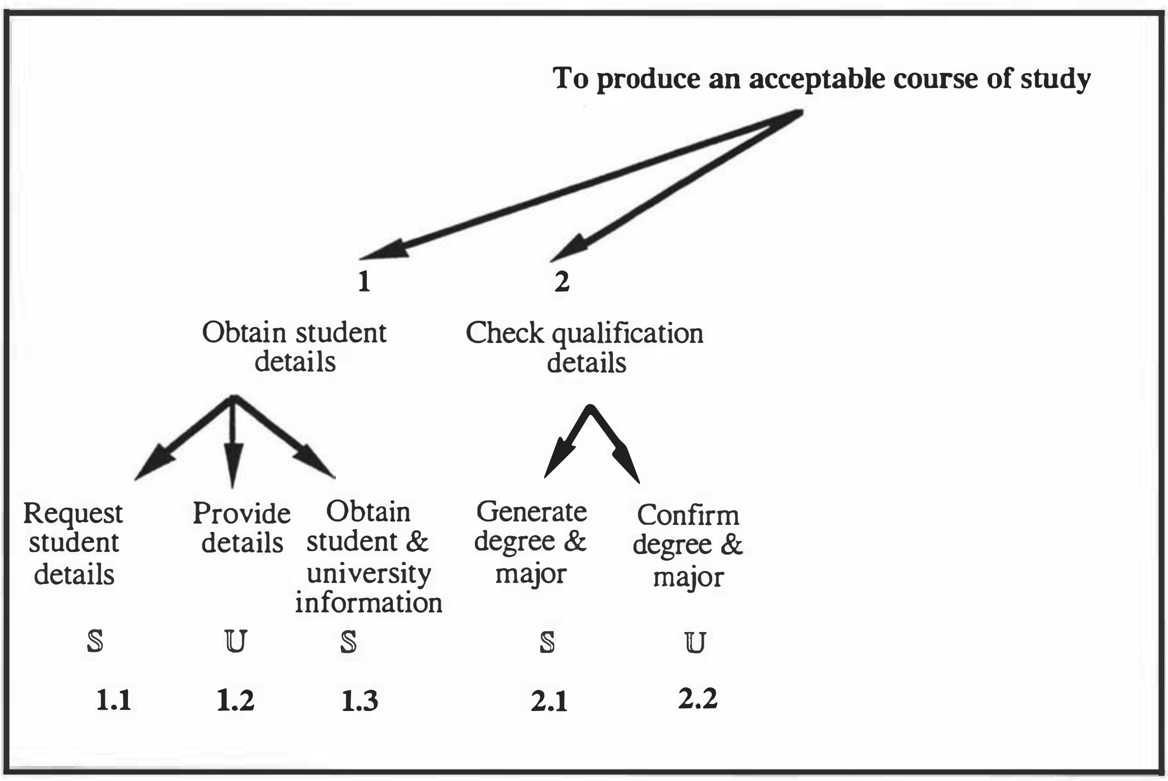
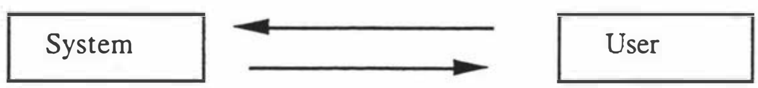


Figure 8.3 Decomposition of part of the extended communication model

8.3 Screen sequencing and organisation

Once the extended communication model is in place, the major application inputs and outputs should be identified. Working systematically through the extended communication model, the numbered flows shown in Figure 8.4 can be derived. First of all the system requests the student's details (1.1) and the student responds (1.2), etc. Using numbered flows makes it easy to match interaction pairs as well as order activities. A decision can then be made about which flows to bring together logically into one screen. These groupings are separated from each other by dashed lines in Figure 8.4.



1.1	Request student details	1.2	Provide details
2.1	Request qualification confirmation	2.2	Qualification confirmation response
- - -	- - -	- - -	- - -
3.1	Request paper details/ Offer advice	3.2.1	Supply paper number
3.2.2.3	Provide advice	3.2.2.2	Request advice
3.3.1	Check paper message	3.3.2	Check paper message response
3.4.1	Request paper confirmation	3.4.2	Paper confirmation response
3.5.1	Complete list request	3.5.2	Complete list response
- - -	- - -	- - -	- - -
4.1.1	Course checked message	4.1.2	Course checked response
4.2.1	Degree status message	4.2.2	Degree status response
6.1	Request course confirmation	6.2	Course confirmation response
- - -	- - -	- - -	- - -
7	Print out request 7.1.1 Degree to date 7.2.1 Course of study 7.3.1 Paper information	7	Print out response 7.1.2 Degree to date 7.2.2 Course of study 7.2.3 Paper information

Figure 8.4 System - user flows

The sequence of screens can be provisionally produced once this step has been completed (Figure 8.5). There is not a one to one correspondence between the top level activities in the model of communication and the screen hierarchy. The re-grouping of activities is based on user rather than knowledge-based requirements. For instance, "Check course of study", "Determine constraints" and "Confirm course" have all been amalgamated into a screen (number 4) that carries out at one point all activities associated with checking a course of study. A major addition is an introductory screen "Getting started" which tells students about the facilities of the advisory knowledge-based system. Because of the changes that have been made, only limited iteration occurs around screens 3 and 4.

Once the principal flows have been identified, it is a simple matter to see where mistakes may be made by students. Error conditions can only occur when a student name, student identification number or paper number is incorrectly entered. A message telling the student that a mistake has been made can be included in the appropriate window and assistance offered. For example, a student can be asked to re-type a paper number or told how to obtain further advice.

It should be remembered that there will be windows at two levels. At the top level, students can enter their details and select papers. The system checks that each paper can be taken and that the list of papers meets the requirements of the degree. From the lower window, a student can obtain timetable information or more detail about papers offered by Massey. The division between these windows is shown in Figure 8.5 by a dotted line. The links between the two levels are represented by the lines that cross the divide.

8.4 Other interface considerations

Before detailed screen design can begin, the arrangement of screens has to be checked to see what other options should be offered to users. Since the enrolment system is one that will only be used intermittently, it is important to generate appropriate help messages to inform a student what is happening. Help can be made available at all times. Other options that can be offered to students to give them more control include Undo, Print and Quit. Students have to have some way of changing a response such as the entry of a paper number. This calls for some kind of Undo facility. Students also have to indicate that they have completed what they want to do, so Quit is required. This option should always be available so that students can leave the system whenever they want. Finally, students will want a Print function so that they can get a print-out of their timetable, paper information and course approval form.

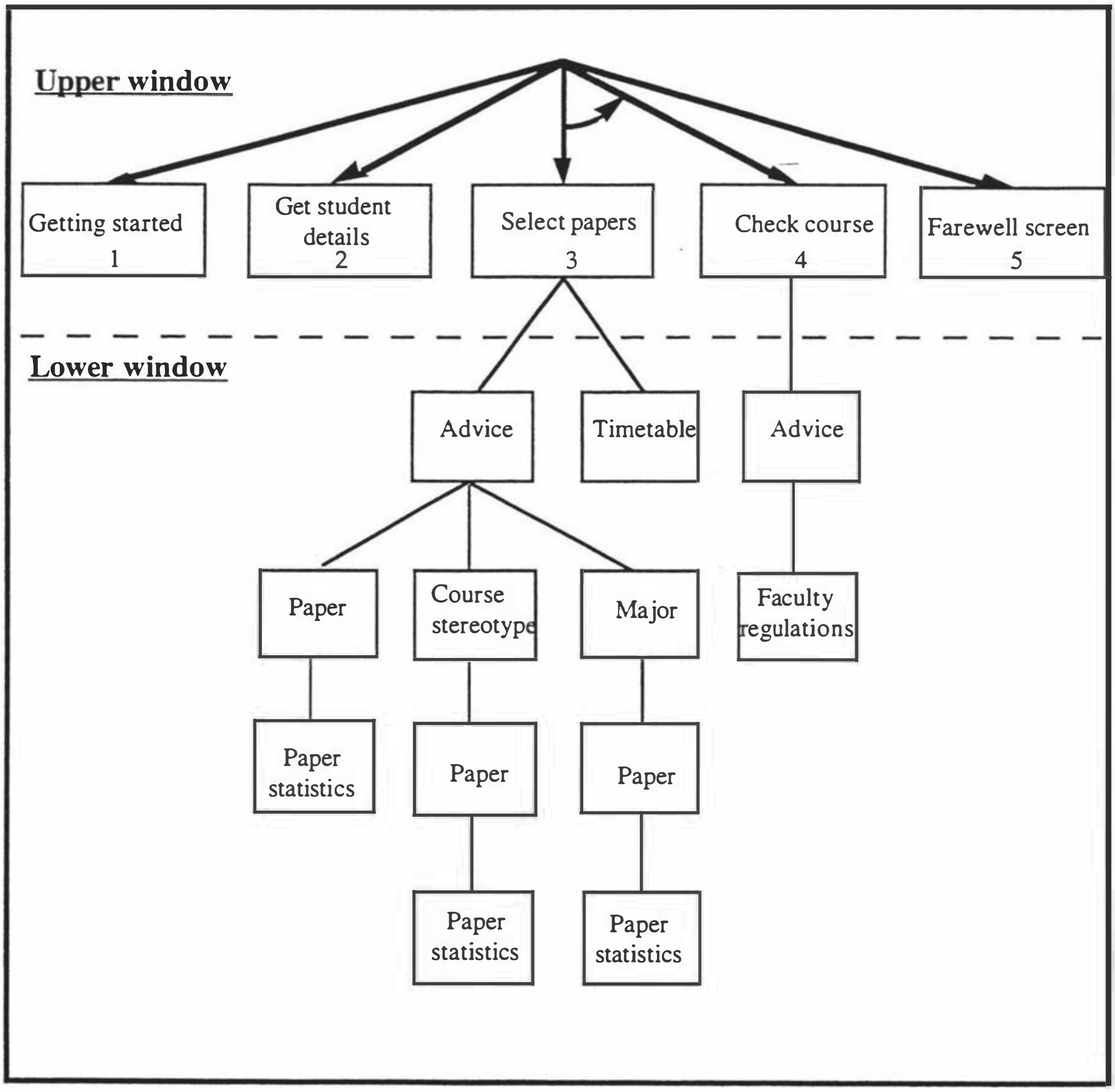


Figure 8.5 Relationship between the two windows

There are various ways these options (Help, Undo, Print and Quit) can be implemented, for example as alternatives in a pull-down menu (Figure 8.6) or buttons between the two screens (see Figure 8.8). Since the decision has already been taken to make it clear to students what operations they can choose at any particular time, these options are shown as buttons which are highlighted when available. The difference between the options can be clearly seen in Figure 8.8 where, of the four, only Help and Quit are operative at this point.

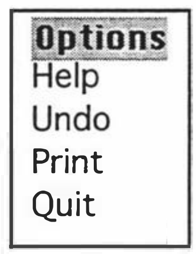


Figure 8.6 Options in a pull-down menu

A knowledge-based system may also offer explanations or advice. The enrolment expert system has to be able to provide advice about papers and courses as well as definitions of terminology. Two mechanisms are suggested for dealing with this. Students who do not know what papers can be included in a degree can press the '?' key whenever they are asked to enter a paper number. The required assistance is provided in the bottom window. Further details about a paper such as its contents and assignment schedule might be required. This can be handled by displaying paper numbers entered in the top window in colour, red, for example. Students can double click on the number and get full information about a paper in the bottom window. The use of colour (shown as bolded text in the layouts) has the advantage of drawing students' attention to the fact that they can get further details. The same approach can be used to provide clarification of terms such as prerequisites and corequisites. These terms that appear in the university regulations can hardly be avoided without giving students imprecise information. Selected terms can be shown in colour and brief definitions presented when students double click on them.

To assist the interface designer, a diagram can be drawn to show when options such as Help (H), Undo, Print, Quit (Q), Clarify (C) and '?' are available to users. A detailed task decomposition for each screen can be produced (based on the extended model of communication) and the options available to users whenever they are in control of the dialogue are indicated. Figure 8.7 shows screen 4 in the enrolment application which deals with checking a course of study, finding out whether a student intends to graduate and asking for confirmation of the enrolment. The options available to the user at each point are clearly marked. This process enables the knowledge engineer to ensure that all the associated links (help and clarification messages and connections with the lower screen) are identified.

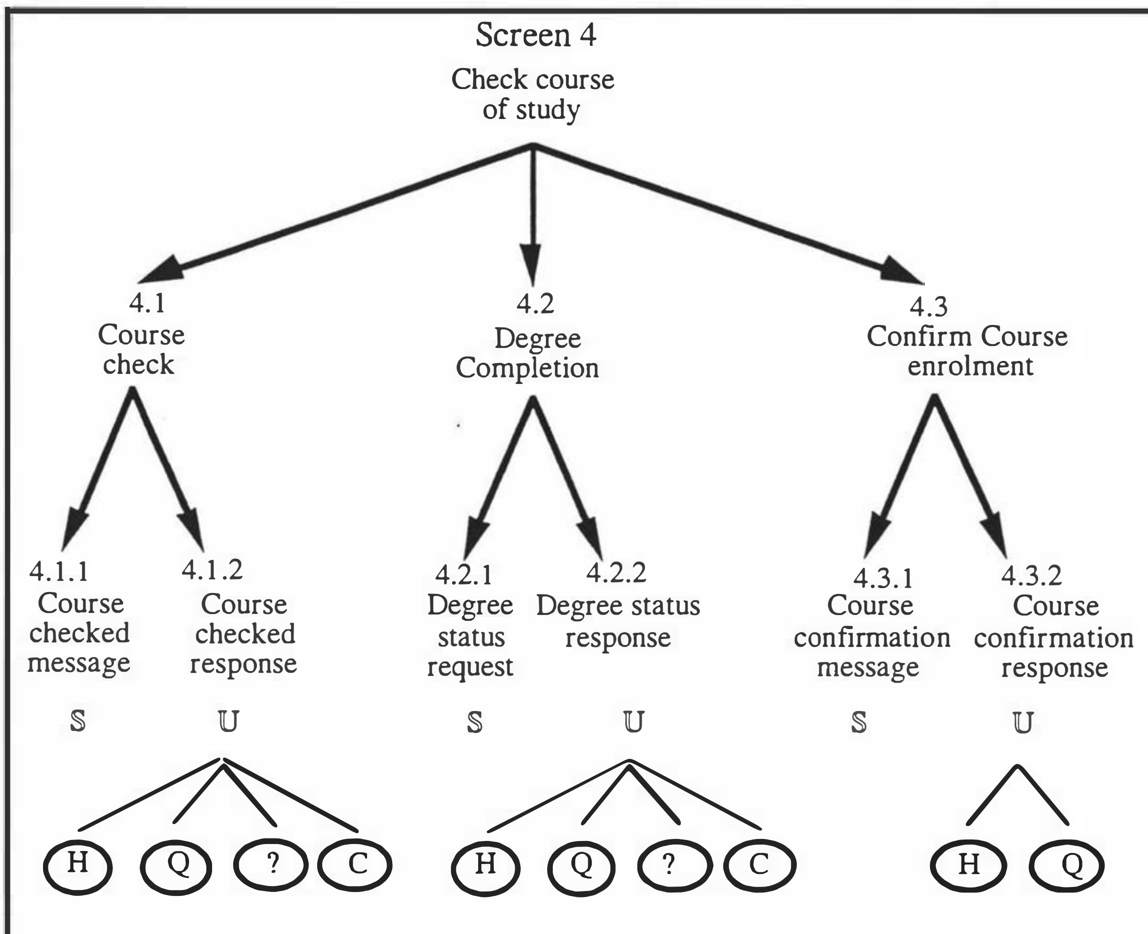


Figure 8.7 Extended communication model with options

There is one final decision to make before screen design can commence. Since a combination of form-fill and direct manipulation presentation styles was recommended, students have to handle data input in two different ways. The issue of input management is discussed in some detail by Bass and Coutaz (1991) who note that two types of cues are often needed: the mouse cursor and the text cursor. Such a course of action will have to be followed here. Whilst the implementation of this can be left until the next stage, a decision has to be made about how users move from one text field to another. It was decided that the return key should be used for this purpose. There is not such a problem with the mouse cursor which can be moved by users to the desired position in a particular window.

8.5 Interface design

Storyboards are used as the medium for interface design. Such screen layouts are very useful, as Johnson (1992) notes, for sketching the interface and showing how a screen changes as an input is received or a process is completed. They can be hand drawn or produced by either drawing or interface generation tools. However they are created, they are essentially a paper specification as there is no associated code. They can be drawn by the knowledge engineer or under her/his direction. In general, there are various advantages to using a specification like storyboards. It is usually quicker and easier to develop a paper specification than a runnable system. The storyboards showing the sequence of activities can be checked by the knowledge engineer to see that all user requirements have been considered and that the interface acts, as Shneiderman (1988) indicates it should, as the framework on which to hang details. These sketches can also be evaluated for usability by employing techniques such as cognitive walkthrough or heuristic evaluation. A prototype can subsequently be developed but the groundwork has been laid for a principled rather than an *ad hoc* approach to prototyping. More specifically, in this chapter, the series of storyboards is the end result of the application of FOCUS. It serves to demonstrate that the outputs from the FOCUS stage lead directly to interface design.

To illustrate how a student would use the enrolment system some sample storyboards are shown here. The screens that appear in the upper window are discussed first. The "Getting started" screen informs students that they can enter paper numbers, have their list of papers checked to ensure that no faculty regulations have been contravened and obtain advice (Figure 8.8). For those who need assistance in running the system, further details can be obtained by selecting the "Info" option. Otherwise students press "OK" to proceed. The four options placed between the two screens (Help, Undo, Print and Quit) are also shown.

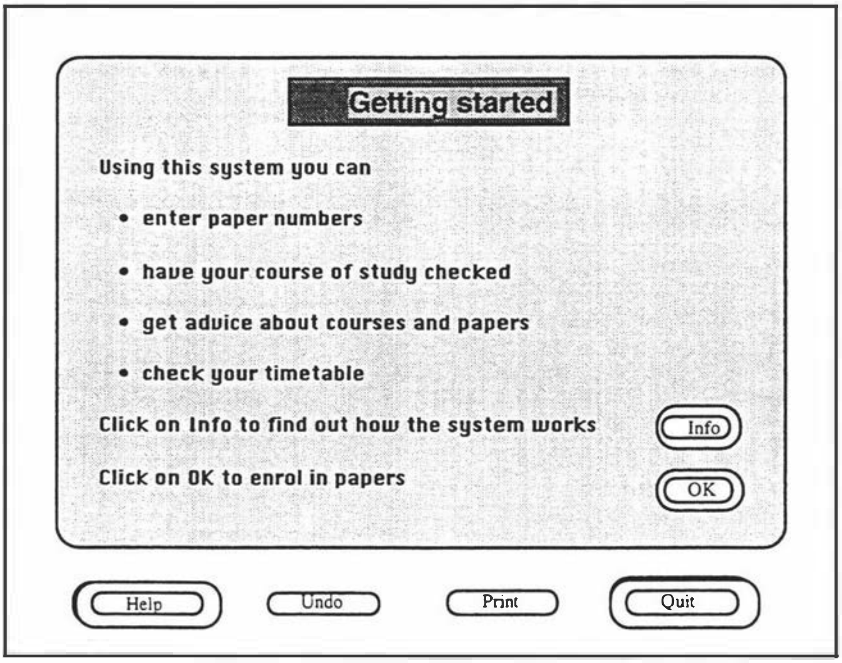


Figure 8.8 "Getting started" screen

Those students who press the "Info" option can scroll through more information about the knowledge base (see Appendix K for full screens). They can learn about the purpose of the two windows (Figure 8.9), the options that are available to them and how to obtain advice. When students feel ready to continue, they can press "OK".

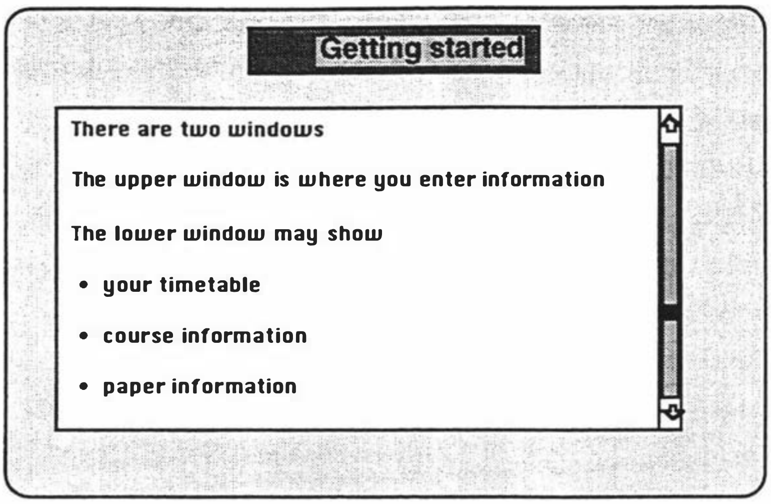


Figure 8.9 Information about the two windows

Students are asked to enter their family and given names as well as their student identification number (Figure 8.10). If the number entered is valid, a message appears displaying the major and the degree in which a student is enrolled. When these details are confirmed, a student can proceed to select papers. If a record cannot be found to match the details, a message is displayed to this effect. If a student makes a mistake typing in details, they can be re-entered, otherwise he or she will have to contact the enrolment clerk to see why there is a problem.

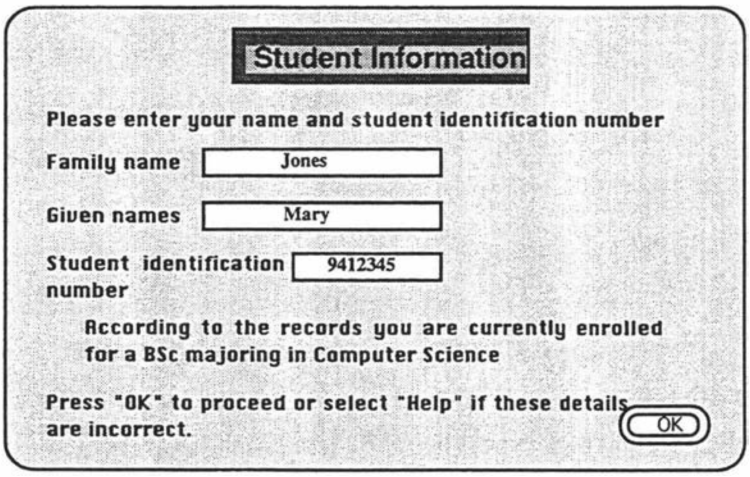


Figure 8.10 Student and degree details

A student can then type in a paper number, have it checked and added to the course of study. On entering the screen entitled "Paper Selection", a student sees four fields: one where a paper number or '?' can be entered, one for the course of study, one to show how many points a student has already obtained and one to show the points value of the course of study. The points value of the student's course of study is calculated incrementally. This basic template is shown in Figure 8.11.

Let us now take the example of first year student, Mary Jones, who wishes to take the paper, 59111. Since the student is new to the university, she has not yet passed any papers and so has no points. There are no prerequisites for 59111, however, so it can be added to the course list. As a check, though, that the paper is the one in which the student intended to enter, the paper title is displayed. Mary Jones can then confirm by pressing "Yes" that she wishes to enrol in the paper. Once this has ben done, the paper number is entered in the course of study list and the points value of the course updated. All this is shown in Figure 8.12. The student is then asked if she has completed her list or wishes to enter another paper. If she has more papers to enter then the slot for entering the paper number has to be cleared and other paper details removed.

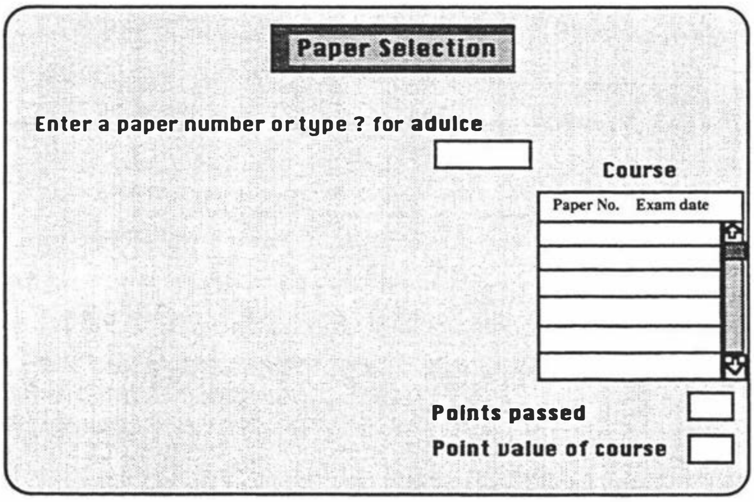


Figure 8.11 Basic template for "Paper Selection"

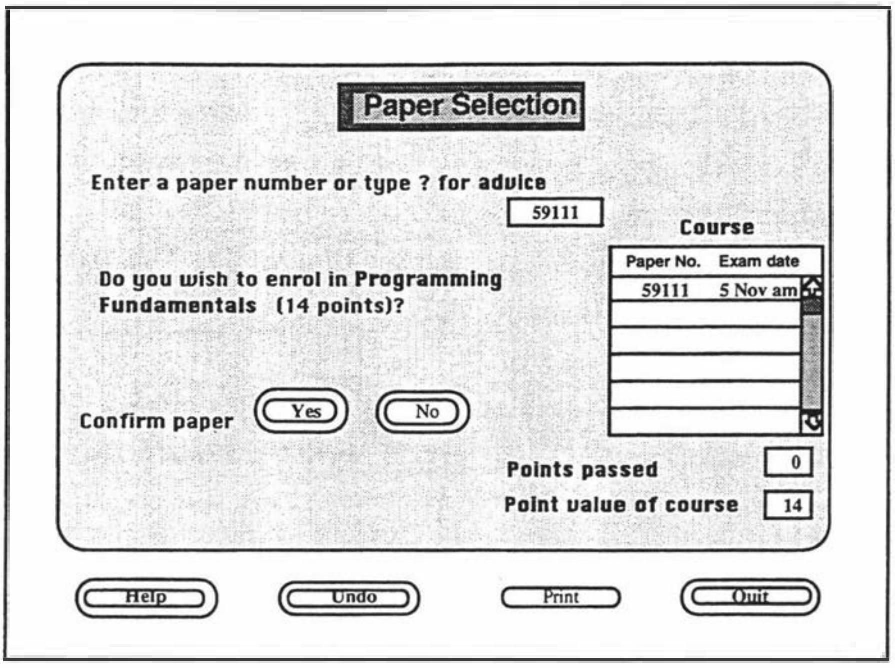


Figure 8.12 Student confirmation

Should Mary Jones subsequently decide not to take a paper, then "Undo" can be selected followed by double clicking on the paper number in the list. She is asked to confirm via a dialogue box that she wishes to remove that particular paper since it is possible to select the wrong number from the Course list (Figure 8.13). The decision is confirmed if "Yes" is pressed otherwise "No" leaves the *status quo*.

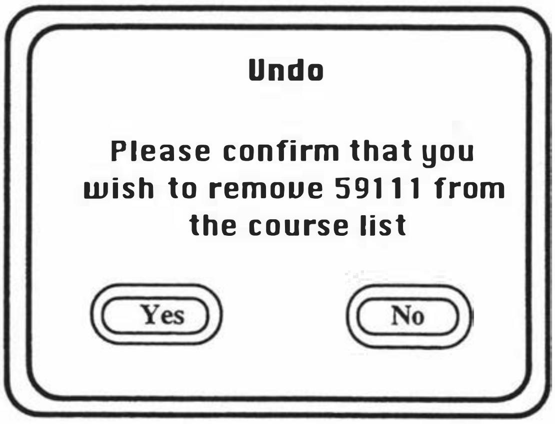


Figure 8.13 Undo paper selection dialogue box

The example given above deals with a straightforward case but what happens if there are problems? The system has first to verify the paper number. If it is invalid then an error message is displayed (Appendix K, Figure 15). If the paper number is valid then various checks are made: that the student has the correct prerequisites, that the paper is not restricted against one that the student has already passed, etc. See Appendix K, Figure 12 for an example of what happens when a student has not passed a prerequisite. If the student does not understand a term in the message such as the word "prerequisite" this can be clarified in a dialogue box when the student double clicks on the term itself (Figure 8.14).

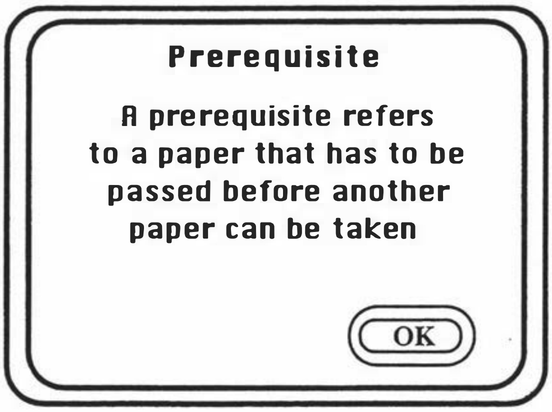


Figure 8.14 Definition of prerequisite

Context-dependent Help is available at all times. When a student first sees the "Paper Selection" screen and is unsure of how to proceed, an example can be given of the correct way to enter a paper number (Figure 8.15).

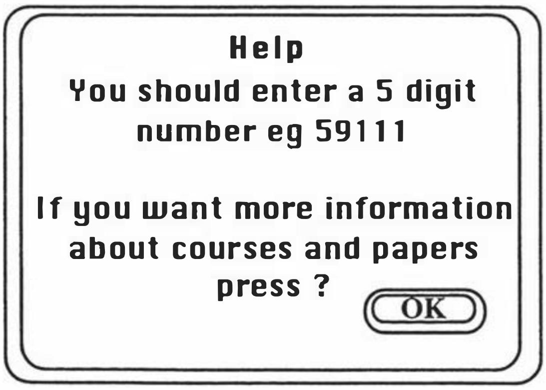


Figure 8.15 Help - enter paper number

Once students have selected their complete course of study, it has to be checked. Assuming that there is no problem, students are asked whether they intend to complete their degree by the end of the year. This system can work out whether students will have the required 300 points or more for graduation. Students who will meet the target are asked to confirm their enrolment (Figure 8.16).

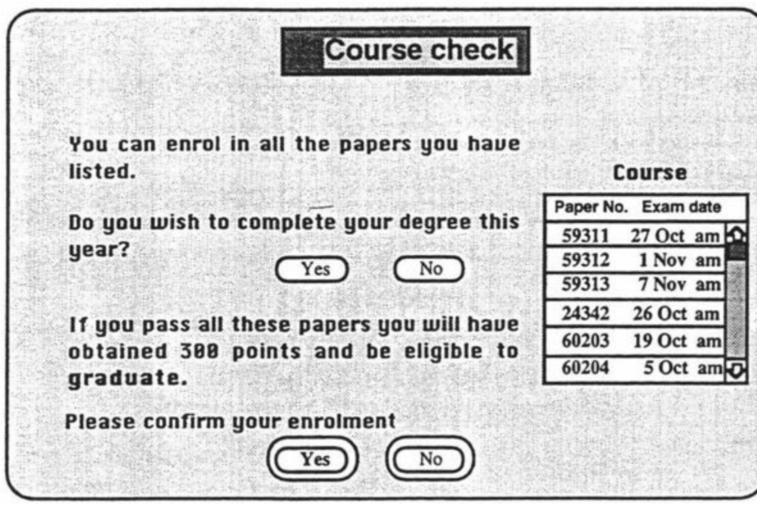


Figure 8.16 "Course check" screen

If they do not wish to confirm this, they can Quit the system (they may have been using the system in an exploratory way) or they can alter their course by returning to the "Paper Selection" screen. Again, Help is available at each step to explain how to continue.

There can be various problems with a course of study: a student who has enrolled in only one of two corequisites, a student taking too many points from outside the faculty schedule, a student enrolled in too many papers, a student failing to meet the requirements for the major and/or a student who does not have the correct number of points specified at the first, second, and third level. Whatever the problem, it is pointed out to the student who has to take some action (Figure 8.17). Advice can be given about Science faculty regulations if the student presses '?'. Students whose course does not meet their faculty's regulations have to return to the "Paper Selection" screen and change their course of study. A message is displayed to this effect. Note that once a course of study has been checked, any further advice is tailored to the student's circumstances. If he/she has too many extra-faculty points then any other paper that the system suggests will come from the faculty schedule.

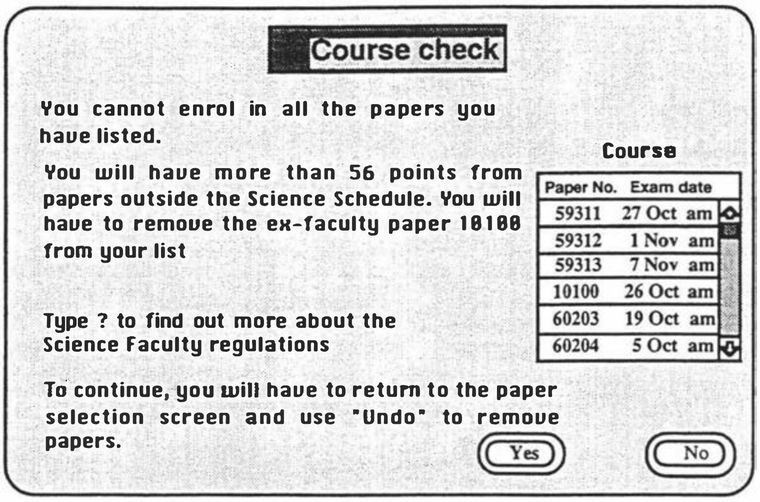


Figure 8.17 "Course check" problem

The "Farewell" screen thanks students for using the system and gives them the opportunity to print out any documents they require (Figure 8.18). These include a course approval form, the timetable and details of the papers a student has passed to date. Information is given about where to collect the print-out.

How are the top and bottom windows related? There are three ways that the upper screen is linked to the lower one: by double clicking on a paper number, by pressing '?' and by entering a paper number. All these cases will be considered.

Students who want to find out more about the material covered in a paper and how it is assessed, can double click on the paper number they have entered. Full details will then be displayed (see Appendix I4). This information can be printed out for students to read later at their leisure.

To obtain assistance with course planning, students should press '?'. The response depends upon the point a student has reached. If a student is entering paper numbers they can find out more about degree courses, sample majoring programmes or individual papers (Figure 8.19). If their course of study is being checked then they can see the regulations of the Faculty in which they are enrolled.

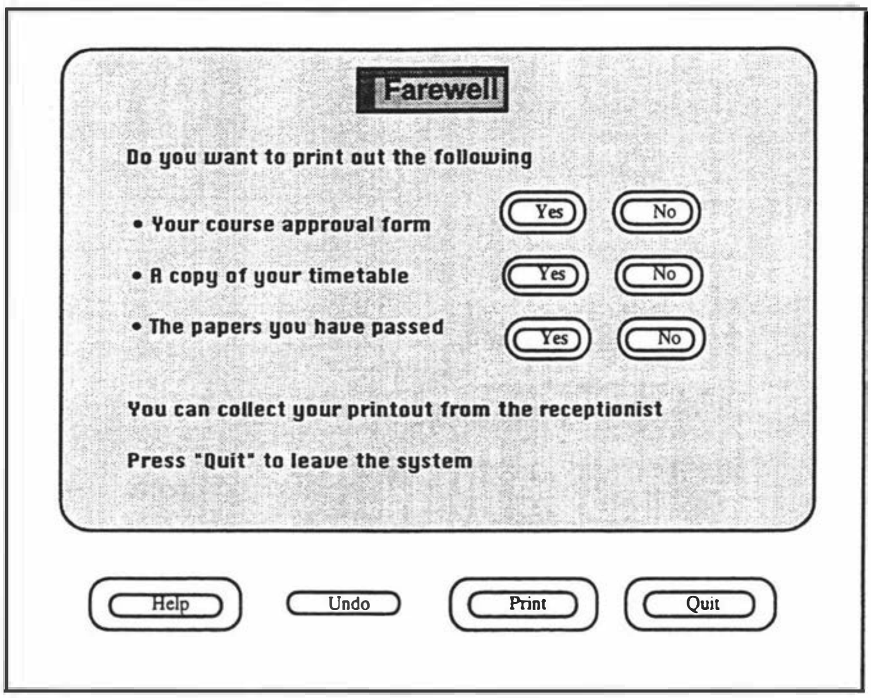


Figure 8.18 "Farewell" screen

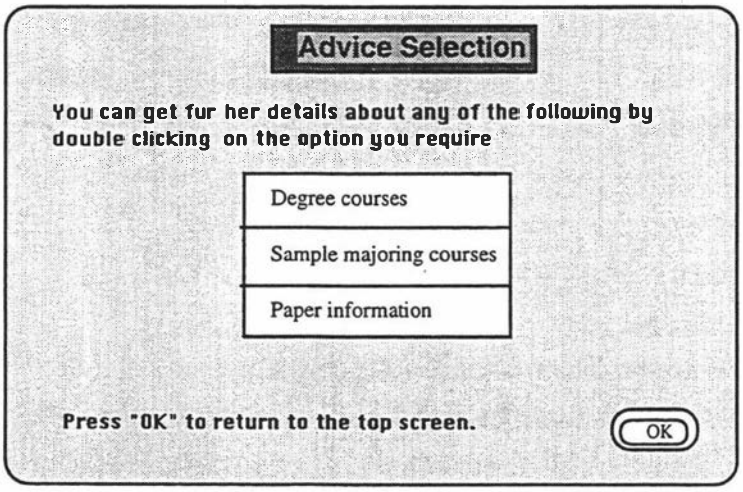


Figure 8.19 "Advice Selection" screen

If a student has pressed '?' from within the "Paper Selection" screen and chooses "Sample majoring courses", a further decision must be made. A student can find out about all the majoring papers (Figure 8.20) or request typical first, second and third year courses of study. The papers shown relate to the major a student is taking. Figure 8.20, for example, lists all the Computer Science papers. The student can find out more about any papers of interest by double clicking on the required number as previously described.

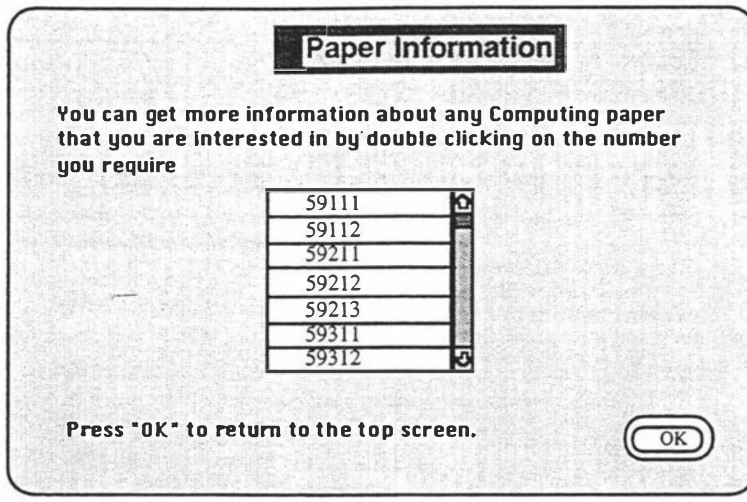


Figure 8.20 "Paper Information" screen

When a student enters a five digit paper number, if it is valid, a timetable will be displayed in the bottom screen (Figure 8.21). This shows the lecture and tutorial slots (optional classes are denoted by italics). This timetable will also show the slots for any other papers that are already in the course list. Students can then see whether they wish to attend lectures at the times indicated and whether they are happy with their overall timetable (they may wish to avoid lectures on Friday, for example). The timetable to date can be printed out if required since many of the features available from the upper window can also be accessed from the lower one. A student can request help, print out paper or timetable details, quit the system and ask for explanations of highlighted terms.

8.6 Conclusion

The purpose of this chapter has been to illustrate how FOCUS leads on naturally to interface design. A major goal of FOCUS is to provide in Hartson and Hix's (1989, p.7) words " an environment in which good interfaces can be constructed." Once the analysis stage is completed, attention is directed in the

Timetable 1994

	Monday	Tuesday	Wednesday	Thursday	Friday
8-9				59.111 SSLB2	59.112 SSLB1
9-10	*59.112 Tut* *SST2.44*		*59.112 Tut* *ScB1.09*		
10-11	*59.112 Tut* *ScB1.09*				*59.111 Tut* *ScB1.09*
11-12					
12-1	*59.111 Tut* *ScB1.09*				
1-2					
2-3			59.112 SSLB1		
3-4					
4-5					
5-6		59.111 SSLB1			

Figure 8.21 Timetable

design phase to important interface issues such as planning screen layouts for the application and deciding what other options (help messages, etc) to offer users. The model of communication and the description of users' explanation and interface requirements from the detailed analysis stage are the foundation on which high level screen design can be built. Adding control to the model of communication enables the knowledge engineer to identify groupings of activities that will each be handled by a screen. Finally, the screens themselves can be designed taking into account the decisions made during the analysis phase about the user's interface requirements. Storyboards are employed to provide a paper specification of the interface. This offers the knowledge engineer a quick way to mock up the interface with the flexibility to modify the design at an early stage.

Part 3

Review and Further Work

Chapter 9

Conclusions and further research

In this chapter, FOCUS is reviewed and evaluated in the light of the objectives specified in section 1.4. The major contributions that this thesis has made in the areas of knowledge-based systems and human-computer interaction are also identified. Finally, suggestions are made for future research.

9.1 Summary of the research

Human-computer interaction was identified as a significant issue in the development of knowledge-based systems. This belief was confirmed by the results from the survey of students who evaluated the APPLE system. Distinct preferences for how information should be input, qualified and output were shown. The literature was subsequently examined to see to what extent user issues were addressed within current knowledge-based systems life cycles. On the whole, they are considered during prototyping, the principal exception being the KADS system development methodology which incorporates a more structured approach. The KADS researchers argue that rapid prototyping leads to unnecessary backtracking and poor design (Hayward, 1986; Wielinga *et al.*, 1987). Besides developing a systematic approach to knowledge-based development, the KADS researchers advocate that user issues should be borne in mind during the analysis phase (Hickman *et al.*, 1989). Unfortunately, they fail to show in detail how this may be achieved. FOCUS was developed in order to rectify this omission.

To assess the process, FOCUS was applied to a university enrolment problem. A preliminary investigation revealed that there were various ways that a knowledge base could be used (by staff, intending or existing students, for instance). What seemed the most useful option, an advisory enrolment system for students, was selected for further development. The detailed organisation model described the problems that the knowledge base would solve, its intended users and the principal functions of such a system. Extensive elicitation of relevant domain knowledge then took place, the problem proving to be, from the knowledge-based perspective, one of construction rather than analysis. The appropriate domain, inference and task layers all had to be built to meet the goals of the enrolment application.

Comprehensive modelling of user needs also proved essential. User opinions were solicited through several surveys as well as interviews with Computer Science students. This revealed that users were very concerned with the enrolment process and had their own views about what an expert system should provide. Users' task and interface requirements were specified in view of the extensive analysis that took place.

Key decisions about human-computer interaction were based on the students' input. Users particularly wanted to obtain information about the papers in which they might enrol and provision was made for this. Interviews with students also indicated that when selecting papers they built up a picture of their timetable. Such a feature was incorporated into the specification of the knowledge-based system. To display paper and timetable information, a two-tiered interface was designed. At the top level, a course of study can be entered and checked whilst the window at the lower level provides the additional timetable and course information students use to make their decisions.

9.2 Review and evaluation of FOCUS

The main objective of this research has been to provide a means of bringing together HCI and knowledge-based systems concerns in a user-centred context. The FOCUS process ensures that users' task and computing requirements are both elicited; the resulting system should meet their functional and interface needs without ignoring organisational and domain issues.

The analysis phase is broken down into several stages: *Problem specification, Preliminary analysis, User analysis, Functional specification* and *Detailed analysis.* During these steps, three deliverables are produced: the organisation, task and conceptual models. In the first four stages, the knowledge engineer checks that the problem is suitable for knowledge-based development, that the application meets the organisation's objectives and that the potential users are consulted early enough for any system to be tailored to their needs.

Relevant models are developed as appropriate. The organisation model specifies the functionality of the proposed knowledge base. FOCUS is flexible enough to permit more than one version of the system to be developed if the

User analysis reveals that the intended users have substantially different goals. Furthermore, if users' computing backgrounds vary widely, the organisation model states whether each group should have its own interface presentation style. Even if it is decided to build more than one version of the system, a single task model describes, at a high level, all the problem solving activities required in the knowledge base.

During *Detailed analysis* the knowledge engineer can concentrate on eliciting domain knowledge and modelling the users. The conceptual model with its three inter-related components is completed. The model of expertise should include all the domain knowledge needed to meet organisational and user goals. For each version of the system, a model of communication is also developed. This shows which party, the user or the computer, carries out each activity. Again, the users have to be considered so that appropriate tasks are allocated to them. Important decisions about the models of expertise and communication that involve the user are made in view of the information held in the user task model.

The three components of the conceptual model are not independent of each other and are developed, to some extent, in parallel. This allows the knowledge engineer to cross-check that all relevant domain information is in the model of expertise and that the principal system objectives are met. Meanwhile, the groundwork has been laid for the design phase. The model of expertise can be converted into a suitable form for computerisation and the interface developed according to the requirements of both the application and the users.

The extent to which FOCUS meets the criteria that should be applied to a framework for use in the analysis phase of the knowledge-based systems development (that it be structured, focused, open and practicable) is now discussed in the context of the example used in this thesis.

A *structured* approach requires that the domain knowledge described in the models is in a form that can be decomposed into its elementary parts for design purposes. From this perspective FOCUS can be seen as a structured approach. The model of expertise itself comprises the domain, inference and task layers. The domain layer contains the information needed for the knowledge base: definitions, support knowledge, concepts and relations. The inference and task layers, on the other hand, relate to the required functionality. With regard to the enrolment application, the relations and regulations required for problem solving are described in the domain layer whilst the inference layer indicates that the main objectives of the system are to build and check a course of study. A detailed description of how these goals can be achieved is provided in the task layer.

In the context of knowledge-based development, a process is *focused* if the problem as well as the domain is studied in detail. FOCUS encourages the knowledge engineer not only to fully explore the domain but also to carry out extensive analysis of the problem as perceived by both the organisation and the potential users. FOCUS, therefore, clearly meets this particular criterion. In the enrolment case study, the ways that a knowledge base would be useful to the organisation were investigated during the initial stages of the analysis. User opinions, too, were extensively canvassed before deciding what functions to incorporate into the knowledge base.

A process should also be *open*, allowing other relevant tools and techniques to be used. Knowledge-based systems development does not take place in a vacuum. Project management techniques have to be employed if a system is to be developed on time and within budget. In FOCUS, much of the information collected is useful for establishing the economic and technical feasibility of a project. The analysis of organisational needs, for example, ensures that the project is one that is suited to knowledge-based development. Since all possible sources of information are identified, the scope of the project and its economic feasibility can also be determined. Milestones can be established to determine when the conceptual model, storyboards, etc should be completed.

Furthermore, FOCUS does not prescribe how the domain knowledge should be obtained A knowledge engineer can select any relevant knowledge gathering techniques whether manual or automated. It is also open, therefore, in this respect. For the enrolment application, the techniques employed included observation, interviewing, questionnaires and reviewing documentation.

Finally, the *practicability* of using FOCUS had to be determined. An illustrative case study demonstrated how a knowledge engineer could use this framework to specify an advisory enrolment system. All relevant models were developed (Chapters 6 and 7) and based on these it was possible to move from the analysis stage to the design of the interface (Chapter 8). The *User analysis* and *Detailed analysis* phases allowed a comprehensive picture to be built up of the users' application, explanation and interface needs. The resulting user models together with the model of communication were invaluable at the design stage for developing an interface to provide users with the desired functionality.

It may be observed that the FOCUS process is lengthy in comparison, for instance, to rapid prototyping. Analysis, however, cannot be short circuited without incurring other costs such as unnecessary backtracking. Tools are also available to shorten the time spent on analysis and design. Software, such as Shelley (Anjewierden *et al.*, 1993), is available to assist the knowledge engineer develop the model of expertise. Hybrid environments offer facilities to design screen layouts. For example, screen layouts can be built using the Smart Elements feature of Nexpert Object (Neuron Data, 1993).

9.3 Contribution of FOCUS

The principles which underpin FOCUS have been derived from a study of the HCI literature and previous research into knowledge-based development. In some respects, FOCUS is like methodologies such as KADS, POMESS and Basden's client-centred approach which have a similar philosophy. Nonetheless, FOCUS incorporates features that in combination make a unique contribution to the identification and specification of users' interaction requirements in a knowledge-based context. A discussion of the principles that underlie FOCUS and the way that they have been implemented follows.

To elicit organisational and user requirements

FOCUS is an approach like POMESS (Diaper, 1988), KADS (Hickman *et al.*, 1989) and Basden's client-centred methodology (Basden, 1989) that emphasises the importance of organisational and user issues. Only FOCUS, however, provides the knowledge engineer with a step by step guide to ascertaining the functional, interface and explanation requirements of users. The possible users of the system are identified during *Problem specification*, their existing situation determined during *Preliminary analysis* and their domain and computing background elicited in the *User analysis* phase. The definition of the knowledge base proposed during *Functional specification* has to meet user needs. A decision has also to be made about whether to build more than one version of the system. Finally, task analysis and user modelling take place in the *Detailed analysis* stage so that user interface requirements can be specified and explanation requirements determined.

To use the modelling approach in knowledge-based development

FOCUS, like KADS, employs a modelling approach to cope with the complexity of knowledge-based systems development (Wielinga *et al.*, 1991). The developers of KADS believe that ideally three streams of analysis are needed to produce the conceptual model: one to handle the requirements of the organisation, one to model the knowledge and a third to deal with user issues (Hickman *et al.*, 1989). This, they admit, is complex and difficult to document. FOCUS, on the other hand, has the one stream of analysis threading through the five stages. Since the outputs of each stage are all defined, it makes the analysis phase easy to document. What is potentially complicated is producing the conceptual model with its three parts: the model of expertise, the model of communication and user requirements. To simplify the process of developing the conceptual model, the role played by each model and the links between them are clearly stated (section 5.2.5). With a conceptual model of this kind, it is possible to verify that all the major requirements have been identified. The various elements of the conceptual model can be cross-checked against each other (the task layer versus the model of communication, for instance) as well as against the functional specification in the organisation model.

FOCUS also emphasises the importance of user modelling. This is often advocated for handling a diverse population of users (Busche, 1989; Cleal and Heaton, 1988; Rich, 1983; Sutcliffe, 1988). Within FOCUS a set of models is identified, from which the knowledge engineer can select as appropriate. This set includes the user task model, the models of user characteristics and knowledge, and the problem solving model of the user. Both stereotypical and individual models can be employed if required. This enables the knowledge engineer to cater for the difficulties that arise when users have varying domain and/or interface requirements. Not only can different versions of the knowledge base be built but the interface can also be tailored to particular needs.

To produce an abstract representation of the knowledge base

FOCUS leads directly to the design phase for both knowledge base and interface design. As with KADS, the model of expertise can be translated into any suitable form of knowledge representation that is required for implementing the expert system. This is not possible with POMESS and Basden's client-centred methodology which both use prototyping as the basis for design. Moreover, in FOCUS, the model of communication can be extended to show iteration, sequence and selection. Breuker and de Greef (1993) state that this can be done in KADS but it serves no particular purpose.) Once this has been completed, decisions can be taken about the order of screens for the application. The detailed screen layouts are designed taking into account the model of characteristics and the user task model. If any changes have to be made to the developed system, there is a full description of both knowledge and user requirements.

To allow for iterative development

FOCUS takes account of the iterative nature of knowledge-based development (Basden, 1989; Guida and Tasso, 1989; Turban, 1992). Models are revised until such time as solutions have been determined for all identified problems. The knowledge engineer, for example, does not produce only one version of the organisation model but revises it extensively throughout the early stages of analysis. This contrasts with the cycles of prototyping that usually occur.

To enable the use of principled prototyping

FOCUS lays the groundwork for principled prototyping to take place. In this thesis, the central role of analysis in knowledge-based systems development has been stressed. This avoids the problems associated with rapid prototyping: a superficial exploration of the domain (Wielinga *et al.*, 1987), backtracking which might involve costly and expensive re-writes (Hickman *et al.*, 1989) and poorly designed systems that do not meet the real needs of the application (Touche Ross, 1989). Once the storyboards have been developed, however, there should be no problem building a prototype for user or expert evaluation. At this point extensive organisation, domain and user analysis have already occurred.

To consider usability issues during the analysis phase

FOCUS permits usability issues to be considered during the analysis stage of knowledge-based systems development (Hix and Hartson, 1993). The criteria for interface evaluation are specified in the organisation model (during *Functional specification*) and are used as guidelines for interface development. The storyboards, a paper specification, can be evaluated early in the design process by using techniques such as cognitive walkthrough or heuristic evaluation to provide formative evaluation. It is may be easier to develop a paper specification than an executable prototype. Storyboards allow the knowledge engineer to quickly produce screens that can be checked at an early stage of the design process. The storyboards can be inspected to see that the sequence of activities is logical, that the criteria for evaluation have been met, that the characteristics of the users have been taken into account and that human-related goals have been met.

9.4 Future research

Other possible avenues for future research were identified during the course of this research.

- FOCUS should be applied to other domains. The author of this thesis is currently involved in a project for assisting dairy farmers to determine the optimum date for drying off their cows in autumn (Kemp *et al.*, 1994). There are two groups of users, farmers and New Zealand dairy board consulting officers. Their backgrounds and needs are quite dissimilar and they would require significantly different interfaces to the knowledge base. This application appears to be an appropriate one for testing FOCUS further.

- FOCUS should also be integrated into a project management life cycle with the tools that would be useful at each point specified. Given the reduced role of prototyping within FOCUS, the spiral approach to knowledge-based development would provide a suitable framework for this.

- It would be useful to extend the enrolment case study. Further development will provide additional feedback on the FOCUS process. The first step would be to build and evaluate the complete enrolment knowledge-based system. Parts of this have already been implemented under the author's supervision using Nexpert Object (Neuron Data, 1993). There are various advantages to using a hybrid environment like Nexpert Object. A choice of methods for developing the knowledge base is available; either the rule-based or object-oriented paradigms can be selected. Smart Elements, a tool for interface development, is also available from Neuron Data. Once the interface is designed the necessary scripts can be written to link the interface with the application.

Once a working system is complete the interface should be evaluated by potential users of the system. Formative evaluation allows users to suggest how the interface can be improved to meet their needs. Some or all of their proposals might be incorporated into the design. When the knowledge base is complete then summative evaluation, to assess the overall performance, can take place. It will be necessary to determine the criteria for summative evaluation by further investigation.

The claim has been made that FOCUS allows more than one version of the system to be developed. A further refinement of the enrolment system would be to implement a module that allowed intending students to select a degree. An interface for a system of this kind has been developed using Hypercard under the author's supervision but has not so far been linked with the knowledge base. Only minor additions to the model of expertise will be required to cater for this group of users.

- The outputs from FOCUS can also act as the starting off point for the interests of other HCI researchers. For example, the storyboards for the enrolment system can be used as an input for various methods of interface evaluation such as cognitive walk-through (Lewis *et al.*, 1990) and heuristic evaluation (Nielsen and Molich, 1990). Work on methods for automatically generating interface code from textual dialogue specification languages is currently taking place (Anderson and Apperley, 1991). Storyboards here can act as an input to test out such methods provided they can be translated into the appropriate dialogue specification language.

References

Allen, J. F. and Perrault, C. R. (1980). Analyzing Intention in Utterances. *Artificial Intelligence, 15*, 143-178.

Allison, G., Catteral, B., Dowd, M., Galer, M., Maguire, M. and Taylor, B. (1992). Human Factors Tools For Designers of Information Technology Products. In M. Galer, S. Harker and J. Ziegler (Eds.), *Methods and Tools in User-Centred Design for Information Technology*. Amsterdam: Elsevier Science.

Anderson, P. S. and Apperley, M. D. (1991). *Dialogue Activation: A Design Approach For Direct Manipulation Interfaces*. Information Science Report No. 91/2. Massey University, New Zealand.

Anjewierden, A., Wielemaker, J. and Toussaint, C. (1993). Shelley- Computer Aided Knowledge Engineering. In G. Schreiber, B. Wielinga and J. Breuker (Eds.), *KADS A Principled Approach to Knowledge-Based System Development* (pp. 247- 264). Cambridge: Academic Press.

Babbie, E. (1989). *The Practice of Social Research* (Fifth ed.). Belmont, California: Wadsworth Publishing Company.

Baroff, J., Simon, R., Gilman, F. and Schneiderman, B. (1988). Direct Manipulation Interfaces in Expert Systems. In J. Hendler and C. Lewis (Eds.), *Expert Systems : The User Interface* (pp. 99-125). Norwood, N. J.: Ablex.

Basden, A. (1989). A Client-Centred Methodology for Building Expert Systems. In A. Sutcliffe and L. Macaulay (Eds.), *People and Computers V: 5th Conference of the British Computer Society Human-Computer Interaction Specialist Group* (pp. 261-275). Cambridge: Cambridge University Press.

Bass, L. and Coutaz, J. (1991). *Developing Software for the User Interface*. Reading, Mass.: Addison-Wesley.

Benyon, D. (1992). The role of task analysis in systems design. *Interacting with Computers, 4*(1), 102-123.

Berry, D. and Hart, A. (1990). Towards effective interfaces. In D. Berry and A. Hart (Eds.), *Expert systems: human issues* (pp. 133-135). Cambridge, Mass.: MIT Press.

Bielawski, L. and Lewand, R. (1991). *Intelligent Systems Design*. New York: Wiley.

Bobrow, D. G., Mittal, S. and Stefik, M. (1986). Expert Systems: Perils and Promise. *Communications of the ACM, 29*(9), 880 -894.

Boehm, B. (1988). A spiral model of software development and enhancement. *IEEE Computer, 21*(5), 61-72.

Bonnet, A., Haton, J.-P. and Truong-Ngoc, J.-M. (1988). *Expert Systems*. Hemel Hempstead: Prentice Hall.

Booth, P. (1989). *An introduction to human -computer interaction*. Hillsdale N.J.: Lawrence Erlbaum Associates.

Brachman, R. J., Amarel, S., Engelman, C., Engelmore, R. S., Feigenbaum, E. A. and Wilkins, D. E. (1983). What are expert systems? In F. Hayes-Roth, D. A. Waterman and D. B. Lenat (Eds.), *Building Expert Systems* (pp. 31-58). New York: Addison-Wesley.

Breuker, J. and de Greef, P. (1993). Modelling System-User Cooperation in KADS. In G. Schreiber, B. Wielinga and J. Breuker (Eds.), *KADS A Principled Approach to Knowledge-Based System Development* (pp. 47-70). Cambridge: Academic Press.

Breuker, J., Wielinga, B., van Someren, M., de Hoog, R., Screiber, G., de Greef, P., Bredeweg, B., Wielemaker, J. B., Davoodi, M. and Hayward, S. (1987). *Model-Driven Knowledge Acquisition: Interpretation Models*. Esprit Project 1098, Deliverable task A1. University of Amsterdam.

Bright, C., Inman, A. and Stammers, R. (1989). Human factors in expert system design: can lessons in the promotion of methods be learned from commercial DP? *Interacting with Computers, 1*(2), 141 -158.

Brownston, L., Farrell, R., Kant, E. and Martin, N. (1985). *Programming Expert Systems in OPS5*. Reading, Mass.: Addison-Wesley.

Brule, J. F. and Blount, A. (1989). *Knowledge acquisition*. New York: McGraw-Hill.

Buchanan, B. G., Barstow, D., Bechtel, R., Bennett, J., Clancey, W., Kulikowski, C., Mitchell, T. and Waterman, D. (1983). Constructing an Expert System. In F. Hayes-Roth, D. A. Waterman and D. B. Lenat (Eds.), *Building Expert Systems* (pp. 127-167). New York: Addison-Wesley.

Buchanan, B. G. and Shortliffe, E. H. (1984a). The Context of the MYCIN Experiments. In B. G. Buchanan and E. H. Shortliffe (Eds.), *Rule-based Expert Systems* (pp. 3-19). Reading, Mass.: Addison-Wesley.

Buchanan, B. G. and Shortliffe, E. H. (1984b). Explanation as a Topic of AI Research. In B. G. Buchanan and E. H. Shortliffe (Eds.) *Rule-based Expert Systems* (pp. 331-337). Reading, Mass.: Addison-Wesley.

Busche, R. (1989). *User Modelling in the Development of Knowledge-based Systems*. Esprit 1098, A6.2. Neu technologien Entwicklungsgesellschaft.

Bylander, T. and Chandrasekaran, B. (1987). Generic Tasks for Knowledge-Based Reasoning: The 'Right' Level of Abstraction for Knowledge Acquisition. *International Journal of Man-Machine Studies, 26*, 231-244.

Card, S., Moran, T. P. and Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale N.J.: Lawrence Erlbaum Associates.

Carey, J. M. and Currey, J. D. (1989). The Prototyping Conundrum. *Datamation*(1 June), 29-33.

Carey, M. S., Stammer, R. B. and Astley, J. A. (1989). Human-computer interaction design; the potential and pitfalls of Hierarchical Task Analysis. In D. Diaper (Ed.), *Task Analysis for Human-Computer Interaction* (pp. 56-74). Chichester: Ellis Horwood.

Carey, T. T. (1991). A Usability Requirements Model For Procurement Life Cycles. In J. M. Carey (Ed.), *Human factors in information systems: an organisational perspective* (pp. 89-104). Norwood, N. J.: Ablex.

Carroll, J. M. and Mack, R. L. (1985). Metaphor, computing systems, and active learning. *International Journal of Man-Machine Studies, 22*, 39-57.

Cawsey, A. (1993). Planning interactive explanations. *International Journal of Man-Machine Studies, 38*, 169-199.

Chandrasekaran, B., Tanner, M. C. and Josephson, J. R. (1988). Explanation : the role of control strategies and deep models. In J. Hendler and C. Lewis (Eds.), *Expert Systems : The User Interface* (pp. 219-247). Norwood, N. J.: Ablex.

Chandrasekaran, B., Tanner, M. C. and Josephson, J. R. (1989). Explaining Control Strategies in Problem Solving. *IEEE Expert, 4*(1), 9-24.

Chew, R. (1987). *Matching Techniques to Tasks in Expert Systems.* Masterate thesis, University of Auckland.

Clancey, W. (1983a). The epistemology of a rule-based expert system - a framework for explanation. *Artificial Intelligence, 20*, 215-251.

Clancey, W. (1983b). GUIDON. *Journal of Computer-Based Instruction, 10*(1), 8-14.

Clancey, W. (1984). Methodology for building an intelligent tutoring system. In W. Kintsch, P. G. Polson and J. J. Miller (Eds.), *Methods and Tactics in Cognitive Science* (pp. 51-83). Hillsdale N.J.: Lawrence Erlbaum Associates.

Clancey, W. (1987). *Knowledge-Based Tutoring*. Cambridge, Mass.: Massachusetts Institute of Technology.

Clancey, W. and Letsinger, R. (1981). NEOMYCIN : reconfiguring a rule-based expert system for application to teaching. *Proceedings of Seventh International Joint Conference on Artificial Intelligence* (pp. 829 - 836), Vancouver.

Cleal, D. M. and Heaton, N. O. (1988). *Knowledge-based systems : implications for human-computer interfaces*. Chichester: Ellis Horwood.

Clowes, I., Cole, I., Arshad, F., Hopkins, C. and Hockley, A. (1985). User Modelling Techniques for Interactive System. In P. Johnson and S. Cook (Eds.), *People and Computers: Designing the Interface* (pp. 33-45). Cambridge: Cambridge University Press.

Cordingley, E. S. (1989). Knowledge elicitation techniques for knowledge-based systems. In D. Diaper (Ed.), *Knowledge elicitation principles, techniques and applications* (pp. 89-172). Chichester: Ellis Horwood.

Coulson, J., Carr, C. T., Hutchinson, L. and Eagle, D. (Eds.). (1976). *The Oxford Illustrated Dictionary* (2nd ed.). Oxford: Oxford University Press.

Coyne, R. D., Rosenman, M. A., Radford, A. D., Balachandran, M. and Gero, J. S. (1990). *Knowledge-Based Design Systems*. Reading, Mass.: Addison-Wesley.

Dance, F. E. X. (1970). The 'Concept' of Communication. *Journal of Communication, 20,* 201-210.

Davis, F. (1993). User acceptance of Information Technology: system characteristics, user perceptions and behavioral impacts. *International Journal of Man-Machine Studies, 38,* 475-487.

de Greef, P., Breuker, J. and de Jong, T. (1988). *Modality: an analysis of functions, user control and communication in knowledge based systems*. Esprit project P1098, Deliverable D6 of task A4. University of Amsterdam.

Desurvire, H. W., Kondzieka, J. M. and Atwood, M. E. (1992). What is Gained and Lost when using Evaluation Methods other than Empirical Testing. In A. Monk, D. Diaper and M. D. Harrison (Eds.), *People and Computers VII* (pp. 89-102). Cambridge: Cambridge University Press.

De Vaus, D. A. (1991). *Surveys in Social Research* (3rd ed.). London: Allen & Unwin.

Diaper, D. (1987). POMESS: A People Oriented Methodology for Expert System Specification. *Proceedings of First European Workshop on Knowledge-Acquisition For Knowledge-Based Systems*, Reading University.

Diaper, D. (1988). The Promise of POMESS: A People Oriented Methodology for Expert System Specification. *Proceedings of Human and organisational issues of expert systems*, Stratford -on-Avon.

Diaper, D. (1989). Task analysis for Knowledge-Descriptions (TAKD); the method and an example. In D. Diaper (Ed.), *Task analysis for human-computer interaction* (pp. 108-159). Chichester: Ellis Horwood.

Dieng, R. and Giboin, A. (1991). Expert Systems Explanations: The Gap between Research and Theory. P.M. Wognum (Ed.), *Proceedings of IJCAI workshop on explanation generation for knowledge-based systems*, Sydney, UT-KBS-91-42 (pp. 23-41), The Netherlands: University of Twente.

Dix, A., Finlay, J., Abowd, G. and Beale, R. (1993). *Human-Computer Interaction.* Hemel Hempstead: Prentice Hall.

Eason, K. and Harker, S. D. P. (1987). A User Centered Approach to the Design of a Knowledge-Based System. In H. J. Bullinger and B. Shackel (Eds.), *INTERACT'87* (pp. 341-346). Amsterdam: Elsevier Science.

Ellis, C. (1989). Explanation in intelligent systems. In C. Ellis (Ed.), *Expert Knowledge and Explanation* (pp. 108-126). Chichester: Ellis Horwood.

Erman, L., London, P. E. and Fickas, S. F. (1981). The design and an example of HEARSAY-III. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence: IJCAI-81*, 7 (pp. 409-415), Vancouver, Canada: IJCAI.

Eshelman, L. (1988). Mole : A Knowledge Acquisition Tool for Cover-and-Differentiate System. In S. Marcus (Ed.), *Automating Knowledge Acquisition for Expert Systems* (pp. 37-80). Boston: Kluwer Academic.

Gaines, B. and Linster, M. (1990). Development of Second Generation Knowledge Acquisition Systems. In B. Wielinga, J. Boose, B. Gaines, G. Schreiber and M. van Someren (Eds.), *Current Trends in Knowledge Acquisition* (pp. 143-160). Amsterdam: IOS Press.

Gaines, B. R. (1987). An Overview of Knowledge-Acquisition and Transfer. *International Journal of Man-Machine Studies*, *26*, 453-472.

Gaines, B. R. and Shaw, M. L. G. (1983). Dialog Engineering. In M. C. Sime and M. J. Coombs (Eds.), *Designing for Human Computer Communications* (pp. 23-53). London: Academic Press.

Gammack, J. G. (1987). Different Techniques and Different Aspects on Declarative Knowledge. In A. L. Kidd (Ed.), *Knowledge Acquisition for Expert Systems: A Practical Handbook* (pp. 137-164.). New York: Plenum Press.

Gaschnig, J. (1982). Prospector, an expert system for mineral exploration. In D. Michie (Ed.), *Introductory readings in expert systems* (pp. 47-64). New York: Gordon and Breach.

Georges, M. (1992). Knowledge Engineering Trends in Europe. In T. Wetter, K. D. Althoff, J. Boose, B. R. Gaines, M. Linster and F. Schmalhofer (Eds.), *Current Developments in Knowledge Acquisition -EKAW'92* (pp. 7-8). Berlin: Springer-Verlag.

Golumbic, M. C., Markovich, M., Tsur, S. and Schild, U. J. (1986). A Knowledge-based Expert System for Student Advising. *IEEE Transactions on Education*, *E-29*(2), 120-124.

Gonzalez, A. J. and Dankel, D. D. (1993). *The engineering of knowledge-based systems: theory and practice*. Englewood Cliffs, N.J.: Prentice Hall.

Gould, J. D. and Lewis, C. (1985). Designing for usability - key principles and what designers think. *Communications of the ACM*, *28*(3), 300-311.

Gregor, S. D. (1991). Explanations in an Expert System For Capital Gains Tax. P.M. Wognum (Ed.), *Proceedings of the IJCAI workshop on explanation generation for knowledge-based systems*, Sydney, UT-KBS-91-42 (pp. 45-56), The Netherlands: University of Twente.

Gregor, S. (1992). Explanations and the communication of knowledge from computer to human. *Proceedings of OZCHI92*, Gold Coast, Australia.

Grice, H. P. (1975). Logic and conversation. In P. Cole and J. Morgan (Eds.), *Syntax and semantics* (pp. 41-58). New York: Academic Press.

Grice, H. P. (1989). *Studies in the way of words.* Cambridge, Mass.: Harvard University Press.

Guida, G. and Tasso, C. (1989). Building Expert Systems: From Life Cycle to Development Methodology. In G. Guida and C. Tasso (Eds.), *Topics in expert system design: methodologies and tools* (pp. 3-24). Amsterdam: Elsevier Science.

Guillemette, R. A. (1991). The Usability Criterion For Designing Information Systems: A Conspectus. In J. M. Carey (Ed.), *Human factors in information systems: an organisational perspective* (pp. 65-87). Norwood, N. J.: Ablex.

Harmon, P. and Hall, C. (1993). *Intelligent Software Systems Development.* New York: Wiley.

Harmon, P. and King, D. (1985). *Expert Systems.* New York: Wiley.

Hart, A. (1992). *Knowledge Acquisition for Expert Systems* (3rd ed.). New York: McGraw-Hill.

Hartson, H. R. and Hix, D. (1989). Human-Computer Interface Development: Concepts and Systems for its Management. *ACM Computing Surveys, 21*(1), 5-92.

Hayes-Roth, F., Waterman, D. and Lenat, D. (1983). An Overview of Expert Systems. In F. Hayes-Roth, D. Waterman and D. Lenat (Eds.), *Building Expert Systems* (pp. 3-29). Reading, Mass.: Addison-Wesley.

Hayward, S. (1986). A structured development methodology for expert systems. *Proceedings of KBS'86* (pp. 195-203), London: Online.

Hendler, J. and Lewis, C. (1988). Introduction : Designing Interfaces for Expert Systems. In J. Hendler and C. Lewis (Eds.), *Expert Systems : The User Interface* (pp. 1-13). Norwood, N. J.: Ablex.

Hickman, F. R. (1989). *An overview of the KADS methodology.* Touche Ross, UK

Hickman, F. R., Killin, J., Land, L., Mulhall, T., Porter, D. and Taylor, R. M. (1989). *Analysis for Knowledge-Based Systems.* Chichester: Ellis Horwood.

Hix, D. and Hartson, H. R. (1993). *Developing user interfaces : ensuring usability through product & process.* New York: Wiley.

Howard, S. and Murray, D. M. (1987). A Taxonomy of Evaluation Techniques for HCI. In H. J. Bullinger and B. Shackel (Eds.), *Human Computer Interaction. Interact'87* (pp. 453-459). Amsterdam: North Holland.

Howey, K. R., Wilson, M. R. and Hannigan, S. (1989). Developing a User Requirements Specification for IKBS Design. In A. Sutcliffe and L. Macaulay (Eds.), *People and Computers V, Proceedings of the 5th Conference of the BCS Human-Computer Interaction Specialist Group* (pp. 277-289). Cambridge: Cambridge University Press.

Jackson, P. (1990). *Introduction to Expert Systems* (2nd ed.). Reading, Mass.: Addison-Wesley.

Johnson, H. and Johnson, P. (1992). Acceptable explanations : Theory before implementation. *Proceedings of AAAI Spring Symposium on Producing Cooperative Explanations.*

Johnson, P. (1989). Supporting system design by analyzing current task knowledge. In D. Diaper (Ed.), *Task Analysis for Human-Computer Interaction* (pp. 160-185). Chichester: Ellis Horwood.

Johnson, P. (1992). *Human computer interaction.* London: McGraw-Hill.

Johnson, P., Diaper, D. and Long, J. B. (1984). Tasks, skills and knowledge: task analysis for knowledge based descriptions. In B. Shackel (Ed.), *INTERACT'84 - Proceedings of the First IFIP Conference on Human-Computer Interaction* (pp. 23-27). Amsterdam: North-Holland.

Karat, J. (1988). Software evaluation methodologies. In M. Helander (Ed.), *Handbook of Human-Computer Interaction* (pp. 891-903). Amsterdam: Elsevier.

Karbach, W., Linster, M. and Voli, A. (1990). Models of Problem-Solving: One label - One idea? In B. Wielanga, J. Boose, B. Gaines, G. Schreiber and M. van Someren (Eds.), *Current Trends in Knowledge Acquisition* (pp. 173-189). Amsterdam: IOS Press.

Kass, R. and Finin, T. (1991). General User Modeling: A Facility to Support Intelligent Interaction. In J. W. Sullivan and S. W. Tyler (Eds.), *Intelligent User Interfaces* (pp. 111-128). New York: Addison-Wesley.

Kelly, C. and Colgan, L. (1992). User Modelling and User Interface Design. In A. Monk, D. Diaper and M. D. Harrison (Eds.), *People and Computers VII* (pp. 227-239). Cambridge: Cambridge University Press.

Kemp, E. A. (1990). Interface Issues in Expert Systems. E. A. Kemp (Ed.), *Proceedings of NZES 90* (pp. 145-158), Massey University, New Zealand.

Kemp, E. A. (1992a). Cognitive Ergonomics and the External Task. *Proceedings of 4th NZ Ergonomics Conference* (pp. 129-151), Massey University, New Zealand. Also in S. V. Burger and F. W. Darby (Eds.), *Human-Computer interaction in New Zealand* (pp. 50-70). New Zealand Ergonomics Society.

Kemp, E. A. (1992b). Communicating with a knowledge-based system. P. Brezillan (Ed.), *Proceedings of ECAI-92 Workshop W15 "Improving the Use of Knowledge-Based Systems with Explanation"*, Vienna, 92/91 (pp. 23-32), Paris: Institut Blaise Pascal.

Kemp, E. A. (1993). FOCUS: A User-Centered Approach to Expert Systems Development. In N. K. Kasabov (Ed.), *The First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems* (pp. 188-191). Los Alamos, California: IEEE Computer Society Press.

Kemp, E. A. and Kemp, R. H. (1991a). Explanation in knowledge-based systems: a user perspective. P. M. Wognum (Ed.), *Proceedings of IJCAI Workshop on explanation generation for knowledge-based systems* (pp. 57-80), Sydney, UT-KBS-91-42, The Netherlands: University of Twente.

Kemp, E. A. and Kemp, R. H. (1991b). The management of the life cycle in expert systems development. *International Journal of Information Resource Management*, *2*(1), 11-23.

Kemp, E. A., Todd, E. G., da Silva, A. and Gray, D. I. (1994). Knowledge acquisition applied to farmer decision making. *Proceedings of SPICIS 94* (pp. B7-B12), Singapore.

Kemp, R. H. and Boorman, A. (1987). Using Wimps to beat the expert system blues. *Proceedings of Second New Zealand Expert System Conference*, (pp. 69-75).

Kemp, R. H., Stewart, T. M. and Boorman, A. (1989). An expert system for diagnosis of pests, diseases and disorders in apple crops. *New Zealand Journal of Crop and Horticultural Science*, *17*, 89-96.

Kidd, A. L. (1987). Knowledge Acquisition: An Introductory Framework. In A. L. Kidd (Ed.), *Knowledge Acquisition for Expert Systems: A Practical Handbook* (pp. 1-16). New York: Plenum Press.

Kieras, D. E. and Polson, P. G. (1985). An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies*, *22*, 365-394.

Kirakowski, J. (1988). *Human-Computer Interaction*. London: Chartwell-Bratt.

Kirwan, B. and Ainsworth, L. K. (1992). Introduction. In B. Kirwan and L. K. Ainsworth (Eds.), *A Guide to Task Analysis* (pp. 1-11). London: Taylor & Francis.

Kitto, C. M. and Boose, J. H. (1989). Selecting knowledge acquisition tools and strategies based on application characteristics. *International Journal of Man-Machine Studies*, *31*, 149-160.

Klut, J. P. and Eloff, J. H. P. (1993). MethoDex: A Methodology for Expert Systems Development. J. Komorowski and Z. W. Ras (Eds.), *Proceedings of Methodologies for Intelligent Systems, 7th International Symposium,(ISMIS '93)* (pp. 106-115), Trondheim, Norway: Springer-Verlag.

Lehner, P. E. and Kralj, M. M. (1988). Cognitive Impacts of the User Interface. In J. A. Hendler and C. Lewis (Eds.), *Expert Systems: The User Interface* (pp. 307-318). Norwood, N.J.: Ablex.

Lewis, C., Polson, P., Wharton, C. and Rieman, J. (1990). Testing a Walkthrough methodology for Theory-based design of Walk-up-and-use-Interfaces. In J. C. Chew and J. Whiteside (Eds.), *Proceedings of CHI'90: Human Factors in Computing Systems* (pp. 235-242). New York: ACM press.

Lindsay, R. K., Buchanan, B. G., Feigenbaum, E. A. and Lederberg, R. (1980). *Applications of artificial intelligence for organic chemistry : the DENDRAL project*. New York: McGraw-Hill.

Lowgren, J. (1989). *Supporting Design and Management of Expert System User Interfaces*. Linkoping University.

Lucas, P. and Van Der Gaag, L. (1991). *Principles of Expert Systems*. Reading, Mass.: Addison-Wesley.

Marcus, S. (1988). SALT:A Knowledge Acquisition Tool for Propose-and-Revise Systems. In S. Marcus (Ed.), *Automating Knowledge Acquisition for Expert Systems* (pp. 81-123). Boston: Kluwer Academic.

Markopoulos, P., Pycock, J., Wilson, S. and Johnson, P. (1992). Adept - A task based design environment. *Proceedings of Hawaii International Conference on System Science* (pp. 587-596): IEEE Computer Society Press.

Martin, J. and Oxman, S. (1988). *Building expert systems : a tutorial.* Englewood Cliffs, N.J.: Prentice-Hall.

Massey University (1992a). *Flying Start*. Palmerston North, New Zealand: Massey University.

Massey University (1992b). *1992 Calendar*. Palmerston North, New Zealand: Massey University.

Massey University (1993a). *Academic Timetable*. Palmerston North, New Zealand: Massey University.

Massey University (1993b). *Flying Start*. Palmerston North, New Zealand: Massey University.

Massey University (1993c). *Science Degree Courses 1993*, Palmerston North, New Zealand: Massey University.

Massey University (1993d). *1993 Calendar*. Palmerston North, New Zealand: Massey University.

McDermott, J. (1982). R1: a rule-based configurer of computer systems. *Artificial Intelligence, 19*, 39-88.

McKeown, K. R., Wish, M. and Matthews, K. (1985). Tailoring Explanations for the User. *Proceedings of 9th International Joint Conference on Artificial Intelligence 18-23 August 1985* (pp. 794-798), Los Angeles, CA.

Mehandjiska, D. S. and Page, D. (1993). Object-Oriented Development of Expert Systems. In N. K. Kasabov (Ed.), *The First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems* (pp. 168-172). Los Alamos, California: IEEE Computer Society Press.

Mitta, D. A. (1989). A Modeling Approach For Measuring Expert System Usability. In G. Salvendy and M. J. Smith (Eds.), *Designing and Using Human-Computer Interfaces and Knowledge-Based Systems* (pp. 783-790). Amsterdam: Elsevier Science.

Moore, J. D. and Swartout, W. R. (1989). *A Reactive Approach to Explanation.* No. ISI/RS-89-239. Information Sciences Institute, University of Southern California.

Moore, J. D. and Swartout, W. R. (1990). Pointing: A Way Towards Explanation Dialogue. *Proceedings of the 8th National Conference on Artificial Intelligence (AAAI '90)* (pp. 457-465), Boston, Mass.

Moran, T. P. (1981). The command language grammar: a representation for the user-interface of interactive computer systems. *International Journal of Man-Machine Studies, 15*(1), 3-50.

Moran, T. P. (1983). Getting into a system, external-internal task mapping analysis. In A. Jamda (Ed.), *Proceedings of CHI 83 : Human Factors in Computing System* (pp. 45-49). New York: ACM Press.

Motta, E., Pitman, K. and West, M. (1988). Support for knowledge acquisition in the Knowledge Engineer's Assistant (KEATS). *Expert Systems, 5*(1), 6-28.

Musen, M. A. (1989). An editor for the conceptual models of interactive knowledge-acquisition tools. *International Journal of Man-Machine Studies, 31,* 673-698.

Musen, M. A., Fagan, L. M., Combs, D. M. and Shortliffe, E. H. (1987). Use of a Domain Model to Drive an Interactive Knowledge-Editing Tool. *International Journal of Man-Machine Studies, 26,* 105-121.

Neuron Data (1993). *Nexpert Object Version 3: User's Guide.* Palo Alto, CA: Neuron Data.

Nielsen, J. (1990). A meta-model for interacting with computers. *Interacting with Computers, 2*(2), 147-160.

Nielsen, J. and Molich, R. (1990). Heuristic Evaluation of User Interfaces. In J. C. Chew and J. Whiteside (Eds.), *Proceedings of CHI'90: Human Factors in Computing Systems* (pp. 249-256). New York: ACM Press.

Nijholt, A. (1991). Foreword. In P. Lucas and L. Van Der Gaag (Eds.), *Principles of Expert Systems* (pp. v-vii). Reading, Mass.: Addison-Wesley.

Norman, D. A. (1986). Cognitive Engineering. In D. A. Norman and S. W. Draper (Eds.), *User centered system design* (pp. 31-61). Hillsdale N.J.: Lawrence Erlbaum and Associates.

Norman, D. A. (1988). *The Psychology of Everyday Things.* New York: Basic Books.

O'Brien, L. (1993). The Billion Dollar Vision. *AI Expert, 8*(5), 5.

op de Hipt, S. (1988). *Human Factors in KBS.* Esprit project 1098, A6. Neu technologien Entwicklungsgesellschaft.

Paris, C., Wick, M. and Thompson, W. (1988). The line of reasoning versus the line of explanation. *Proceedings of Workshop on Explanation, AAAI-88* (pp. 4-7), Saint-Paul, Minnesota.

Paris, C. L. (1991). The role of the user's domain knowledge in generation. *Computational Intelligence*, *7*(2, May), 71-93.

Partridge, D. A. (1986). *Artificial intelligence: applications in the future of software engineering*. Chichester: Ellis Horwood.

Payne, E. C. and McArthur, R. C. (1990). *Developing Expert Systems*. New York: Wiley.

Porter, D. (1992). *Towards the CommonKADS method*. Esprit Project P5248, (KADS-II), T5.1.2. Touche Ross, London.

Potter, A. (1988). Direct Manipulation Interfaces. *AI Expert*, *3*(10), 28-35.

Preece, A. D. (1990). DISPLAN; Designing a Usable Medical Expert System. In D. Berry and A. Hart (Eds.), *Expert systems: human issues* (pp. 25-47). Cambridge, Mass.: MIT Press.

Preece, J. (Ed.). (1993). *A Guide to Usability*. Wokingham, England: Addison-Wesley.

Pressman, R. S. (1992). *Software Engineering A Practitioner's Approach*. Singapore: McGraw-Hill.

Quinlan, J. R. (1982). Fundamentals of the knowledge engineering problem. In D. Michie (Ed.), *Introductory Readings in Expert Systems* (pp. 33-46). New York: Gordon and Breach.

Ravden, S. J. and Johnson, G. I. (1989). *Evaluating usability of human-computer interfaces : a practical method.* Chichester: Ellis Horwood.

Rector, A. L., Newton, P. and Marsden, P. H. (1985). What kind of system does an expert need? In J. P and C. S (Eds.), *People and Computers: Designing the Interface* (pp. 239-247). Cambridge: Cambridge University Press.

Regoczei, S. (1992). Agent and meaning modelling as a comprehensive general interpretation of knowledge acquisition. *Proceedings of AAAI Spring Symposium*, Stanford, CA.

Reichman-Adar, R. (1984). Extended Person-Machine Interface. *Artificial Intelligence*, *22*, 157-218.

Rich, E. (1983). Users are individuals: individualizing user-models. *International Journal of Man-Machine Studies*, *18*, 199-214.

Roberts, H. J. (1990). Expert Systems Clubs: Design Methods. *The Computer Journal*, *33*(6), 556-561.

Rolston, D. W. (1988). *Principles of artificial intelligence and expert systems development*. New York: McGraw-Hill.

Rubin, T. (1988). *User Interface Design For Computer Systems*. Chichester: Ellis Horwood.

Sacerdoti, E. D. (1991). Managing Expert-System Development. *AI Expert*(5), 26-33.

Schreiber, G., Wielinga, B. and Breuker, J. (Eds.). (1993). *KADS A Principled Approach to Knowledge-Based System Development*. London: Academic Press.

Sekaran, U. (1984). *Research Methods For Managers*. New York: Wiley.

Sell, P. S. (1985). *Expert systems: a practical introduction*. Basingstoke: Macmillan.

Senn, J. A. (1989). *Analysis and Design of Information Systems*. New York: McGraw-Hill.

Shackel, B. (1984). The concept of usability. In J. L. Bennett, D. Case, J. Sandelin and M. Smith (Eds.), *Visual display terminals: usability issues and health concerns* (pp. 45-87). Englewood Cliffs, N.J.: Prentice Hall.

Shackel, B. (1986). Ergonomics in design for usability. In M. D. Harrison and A. F. Monk (Eds.), *People and computers: design for usability* (pp. 44-64). Cambridge: Cambridge University Press.

Sharma, D. (1992). DPLANX: A Constraint-Based Planning System. G. K. Gupta and C. D. Keen (Eds.), *Proceedings of Fifteenth Australian Computer Science Conference ACSC-15*, 14 (pp. 833-845), University of Tasmania.

Shneiderman, B. (1987). *Designing the User Interface: Strategies for Effective Human Computer Interaction*. Reading, Mass.: Addison Wesley.

Shneiderman, B. (1988). Series editor's preface. In J. Hendler and C. Lewis (Eds.), *Introduction : Designing Interfaces for Expert Systems* (pp. vii-x). Norwood, N. J.: Ablex.

Shneiderman, B. (1992). *Designing the User Interface: Strategies for Effective Human Computer Interaction* (2nd ed.). Reading, Mass.: Addison-Wesley.

Shortliffe, E. H. (1976). *Computer-based medical consultation: MYCIN*. New York.: American Elsevier.

Stelzner, M. and Williams, M. D. (1988). The Evolution of Interface Requirements for Expert Systems. In J. Hendler and C. Lewis (Eds.), *Expert Systems : The User Interface* (pp. 285-306). Norwood, N. J.: Ablex.

Sutcliffe, A. S. (1988). *Human-Computer Interface Design*. London: Macmillan Education Ltd.

Swartout, W. R. (1983). XPLAIN: a system for creating and explaining expert consulting programs. *Artificial Intelligence*, *21*, 285-325.

Swartout, W. R. and Smoliar, S. W. (1989). Explanation: A Source of Guidance for Knowledge Representation. In K. Morik (Ed.), *Knowledge Representation and Organisation in Machine Learning* (pp. 1-15). Berlin: Springer-Verlag.

Tanimoto, S. (1987). *The elements of artificial intelligence : an introduction using LISP*. Rockville, MD: Computer Science Press.

Tanner, M. and Keuneke, A. (1991). Explanations in Knowledge Systems : The Roles of the Task Structure and Domain Functional Models. *IEEE Expert*, *6*(3), 50-57.

Thimbleby, H. (1990). *User Interface Design*. New York: ACM Press.

Todd, E. G., Gray, D. I., Kemp, E. A., Lockhart, J. C. and Parker, W. J. (1993). Using the KADS methodology: Expert systems in farm management. N. Kasabov (Ed.), *Proceedings of The First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, (pp. 262-265), Dunedin, IEEE Computer Society Press.

Touche Ross (1989). *Integrating With The Mainstream Knowledge-Based Systems: The Way Ahead.*, Touche Ross, London.

Turban, E. (1992). *Expert Systems and Applied Artificial Intelligence*. New York: Macmillan.

Tuthill, G. S. (1990). *Knowledge Engineering*. Blue Ridge Summit: TAB BOOKS.

Waterman, D. (1986). *A Guide to Expert Systems*. Reading, Mass.: Addison-Wesley.

Weekaroon, A. I., Kwon, O. and Leatherwood, M. (1992). Intelligent Student Adviser: Development and Implications of Expert Systems Technology for Academic Advising. *Proceedings of 23rd Annual Meeting of the Decision Sciences Institute* (pp. 674-676), San Francisco, CA.

Weitzel, J. R. and Kerschberg, L. (1989). Developing Knowledge-Based Systems: Reorganizing the System Development Life Cycle. *Communications of the ACM*, *32*(4), 482-488.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. Los Altos, California: Morgan Kaufman.

Wetter, T. (1992). Preface. In T. Wetter, K. D. Althoff, J. Boose, B. R. Gaines, M. Linster and F. Schmalhofer (Eds.), *Current Developments in Knowledge Accquisition-EKAW'92* (pp. V-IX). Berlin: Springer-Verlag.

Whitefield, A. (1987). Models in human-computer interaction: a classification with special reference to their use in design. In H. J. Bullinger and B. Shackel (Eds.), *Human Computer Interaction. Interact'87* (pp. 57-63). Amsterdam: North Holland.

Wick, M. R. and Slagle, J. R. (1989). The Partitioned Support Network for Expert Systems Justification. *IEEE Trans on Systems, Man. and Cybernetics*, *19*(3), 528-535.

Wielinga, B., Bredeweg, B. and Breuker, J. (1987). Knowledge Acquisition for Expert Systems. In R. Nossum (Ed.), *Lecture Notes in Artificial Intelligence* (pp. 96-124).

Wielinga, B., Van de Velde, W., Schreiber, G. and Akkermans, H. (1993). Towards a Unification of Knowledge Modelling Approaches. In J.-M. David, J.-P. Krivine and R. Simmons (Eds.), *Second Generation Expert Systems* (pp. 299-335). Berlin: Springer-Verlag.

Wielinga, B. J., Breuker, J. A. and van Someren, M. W. (1986). *The KADS System Functional Description*. Esprit Project P1098, T1.1. University of Amsterdam.

Wielinga, B. J., Schreiber, A. T. and Breuker, J. A. (1991). *KADS: A Modelling Approach to Knowledge Engineering*. Esprit Project P5248 (KADS-II), University of Amsterdam.

Wognum, P. M. and Mars, N. J. I. (1991). Why is explanation still limited in practice? P.M. Wognum (Ed.), *Proceedings of the IJCAI workshop on explanation generation for knowledge-based systems*, Sydney, UT-KBS-91-42 (pp. 81-90), The Netherlands: University of Twente.

Yin, R. K. (1989). *Case Study Research Design and Methods*. Newbury Park: Sage Publications.

Young, R. (1989). Human interface aspects of expert systems. In L. Murray and J. T. Richardson (Eds.), *Intelligent Systems in a Human Context* (pp. 20-34). Oxford University Press.

Young, R., Green, T. R. G. and Simon, T. (1989). Programmable user models for predictive evaluation of interface designs. In K. Bice and C. Lewis (Eds.), *Human Factors in Computer Systems, CHI89 Wings for the Mind* (pp. 15-19). Reading, Mass.: Addison-Wesley.

Yourdon, E. (1979). *Managing the Structured Techniques*. Englewood Cliffs, New Jersey: Prentice-Hall.

Appendix A

APPLE questionnaire

APPLE expert system questionnaire

Q1. Which degree are you enrolled for ?

Q2. What software packages have you used ? (Please specify)

Q3. What kinds of computers have you previously used ? (Please tick)

Macintosh

IBM PC (or similar)

Workstation (such as the Sun)

Other (specify)

You are asked to circle a number on a five point scale for each of the following questions

Q4. How would you rate your knowledge of crop disorders?

Poor 1 2 3 4 5 Good

Q5. How useful do you think the Apple Expert System would be for teaching students about apple disorders (in association with lectures)?

Poor 1 2 3 4 5 Good

Q6. How useful do you think the Apple Expert System would be for teaching students about apple disorders (without lectures)?

Poor 1 2 3 4 5 Good

Q7. How useful do you think the Apple Expert System would be for orchardists in the field diagnosing apple disorders?

Poor 1 2 3 4 5 Good

Q8. Do you find the "Help" facility at the beginning sufficient?

No 1 2 3 4 5 Yes

Q9. Would you like a "Help" facility available throughout the session?

No 1 2 3 4 5 Yes

Q10. How do you prefer to input information?

English phrases and sentences No 1 2 3 4 5 Yes

Menus No 1 2 3 4 5 Yes

Direct manipulation (eg, using the mouse)
No 1 2 3 4 5 Yes

Q11. How would you like conclusions (such as the likelihood of various diagnoses) to be presented to you?

English phrases and sentences No 1 2 3 4 5 Yes

Graphics (eg, bar graph) No 1 2 3 4 5 Yes

Numerical table (eg, values from -5 to +5)
No 1 2 3 4 5 Yes

Q.12 Would you like to engage in a dialogue with the expert system, asking questions whenever you wanted to ?

No 1 2 3 4 5 Yes

Q.13 Would you like answers to your questions to be tailored to your level of expertise (with regard to apple disorders)?

No 1 2 3 4 5 Yes

Q.14 At the end of a session would you like to be able to see the reasoning behind the conclusions given?

No 1 2 3 4 5 Yes

Q.15 Would you like answers that related to the underlying causes of problems, such as relevant bio-chemical information?
No 1 2 3 4 5 Yes

Q.16 Would you find pictorial information like pictures of diseased leaves helpful ?

No 1 2 3 4 5 Yes

Q.17 Do you feel you know at each stage what the expert system is trying to do?

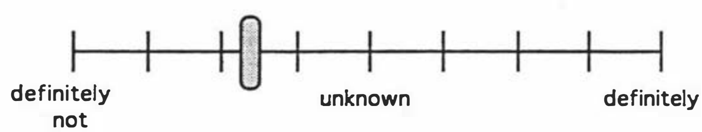
No 1 2 3 4 5 Yes

Q.18 How would you prefer to indicate confidence in your answer to a question?

Text :
"Definitely", "Definitely not", "Possibly" No 1 2 3 4 5 Yes

Numerical input :
Input a value between -5 and +5 No 1 2 3 4 5 Yes

Move a pointer along a linear scale :



No 1 2 3 4 5 Yes

Q.19 How helpful did you find the following facilities?

a. Why Poor 1 2 3 4 5 Good

b. Clarify Poor 1 2 3 4 5 Good

Appendix B

APPLE questionnaire results

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	Q1	Q2		Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q10	Q10	Q11	Q11	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q18	Q18	Q19	Q19
2	BBS			1	5	4	4	4	4	3	4	4	5	3	3	4	4	4	4	5	3	4	3	4		
3	BA			1	3	2	3	4	4	2	5	1	4	4	1	5	4	5	5	5	3	4	1	1	2	2
4	BBS			1	4	2	4	3	5	4	4	3	3	5	4	4	5	4	2	2	3	4	1	3	4	3
5	BBS			1	5	3	3	5				5	5	5	5	2	3	5	5	5	1		5		2	
6	BBS			1	4	2	5	3	5	2	4	5	4	5	1	4	4	5	4	5	4	4	3	5	3	4
7	BBS			1	4	3	5									4	4	5		4	4	3	4	5	4	2
8	BBS			1	3	1	4	2	4	1	5	3	4	4	1	3	4	5	3	4	1	1	4	3	2	2
9	BA			1	3	2	3	2	5	3	2	5	3	4	4	5	5	5	1	3	4	3	4	5	5	5
10	BA			1	4	2	5	5	3	2	4	4	4	5	3	1	5	5	3	5	4	5	3	4	5	5
11	BA			1	4	2	5	2	5	1	1	5	1	1	5	5	5	5	5	4	4	1	1	5	2	2
12	BBS			1	3	1	4	3	4	1	1	5	3	4	4	2	5	5	5	3	5	1	3	5	3	5
13	BBS			1	3	2	4	3	1	1	3	5	3	5	3	4	4	5	4	3	3	3	1	2	3	4
14	BA			1	4	2	4	3	5	5	3	3	5	4	2	5	5	5	5	5	4	5	3	4	4	4
15	BAG			1	4	4	5	4	2	1	4	4	3	4	1	5	3	5	5	2	5	4	1	5	4	4
16	BAG			1	3	2	3	1	5	4	3	1				4	4	4	3	4	3	2	4	1	4	3
17	BAG			1	4	3	3	3	5	3	5	5	3	4	1	5	5	5	4	5	4	4	3	5	4	4
18	DAG			1	4	4	3	3	5	2	3	5	5	3	3	3	5	5	5	5	4	4	4	4	4	4
19	BAG			1	1	1	4	5	5			5	5	5	1	5	5	5	5	5	3	5	1	5	5	5
20	BBS			1	4	4	5	3	3	1	4	5	5	4	2	3	4	5	5	5	4	4	2	4	4	4
21	BBS			2	5	4	5	4	5	5	1	5	5	5	1	5	4	5	3	5	1	2	4	5	5	5
22	BBS			2	4	5	5	3	2	2	4	5	4	4	1	2	5	5	5	4	4	4	1	4	4	3
23	MBS*			2	5	3	4	1	5	3	5	5	5	5	5	5	4	3	5	5	5	3	2	5	5	5
24	BBS			2	5	4	5	4	4	1	5	5	5	4	3	1	4	4	5	5	5	5	1	4	5	3
25	BORT			2	3	3	4	5	3	3	4	5	4	4	4	4	5	5	4	4	5	3	4	5	4	5
26	BAG			2	4	4	4	3	5	1	4	5	5	4	2	3	4	4	5	2	5	1	5	5	4	4
27	BAG			2	4	2	2	4	5	4	5	1	5	1	2	5	4	5	5	5	2	2	5	2	4	4
28	BAG			2	3	2	3	2	4	1	4	5	4	4	2	3	4	5	3	3	4	4	1	3	4	4
29	BAG			2	3	3	4	5	5	1	5	1	5	2	2	2	5	5	2	1	3	5	1	1	5	5
30	BORT			2	3	2	4	4	5	3	5	5	4	4	4	5	5	5	4	5	5	2	4	4	4	5
31	BORT			2	4	2	2	5	1	3	5	4	4	5	2	3	4	5	2	3	5	2	5	4	2	4
32	DAG			3	5	3	4	3	5	4	3	5	4	3	4	5	5	5	1	5	3	5	4	3	4	4
33	DAG			3	5	4	4	4	5	5	4	5	5	3	5	5	5	5	4	5	4	5	4	2	4	4
34	BORT			3	4	3	4	4	5	2	3	5	3	5	4	4	5	5	4	5	4	3	5	4	4	1
35	BORT			4	4	3	4	4	3	3	5	4	5	5	5	2	4	5	4	4	5	3	4	4	5	5
36	BORT			4	5	2	4	5	2	1	3	5	5	4	4	4	5	5	5	4	5	4	2	4	5	5
37	BORT			4	4	2	4	1	5	2	4	4	4	3	3	5	4	4	3	4	4	5	2	4	5	5
38	BORT			4	5	3	4	5	4	1	3	5	3	4	3	4	5	5	5	5	4	5	3	4	4	4
39	MBA			4	3	2	5	3	4	5	4	3	3	5	4	5	4	5	5	4	4	3	4	3	5	4
40	BORT			5	3	2	4	1	5	4	1	5	3	5	3	5	5	5	5	4	2	2	5	5	3	3

Appendix C

Extract from "Communicating with a knowledge-based system" (Kemp, 1992b)

The intention in this paper is to examine the results for two of the groups only, B and C. These are the users who can be described as having a limited and good knowledge respectively of the domain. Their responses to questions concerning interface styles, explanation facilities and alternative knowledge sources will be considered. Boxplots are used to compare the results for each group. They can be used with interval data to show the overall level of values, the spread of the data, outliers, how the levels of the variables compare and how the spreads of the variables compare. Boxplots are based on the median; unlike measures that use the mean they not affected by outliers. For this reason they are a suitable tool for analysing exploratory data.

Boxplots of the results for the natural language option are shown in Figure 3. The rectangular box for each group depicts the extent of the middle half of the data (between the 25th and 75th percentiles). The horizontal line in the box for group C denotes the median. For group B the median coincides with the top edge of the box. The horizontal lines at the top and bottom of each plot indicates the range of the main body of each set of data. The shaded area around each median takes into account the variance and is used for testing significance at the 5% level. For example, if the two shaded areas in Figure 3 did not overlap then any difference between the medians would be statistically significant. So the boxplots in Figure 3 show that the answers for group B ranged from 1 to 5 with the median at three. The spread and the median is the same for group C but we can see that the main body of data is distributed symmetrically about the median for group C in contrast to the data for group B which seems to be skewed to the bottom. Overall, approval of the natural language option was rather lukewarm with the median for each group only three.

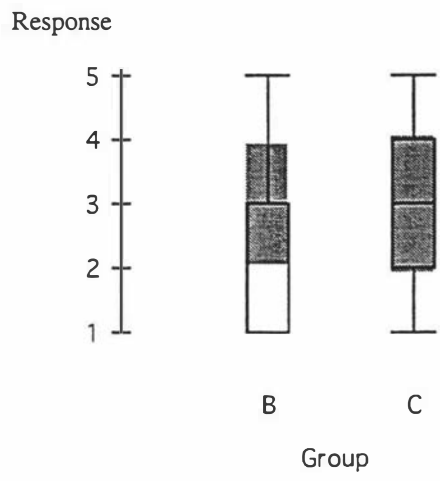


Figure 3 Natural language input

This is not the case for the question about menu input (see Figure 4). Even though the spread (taking outliers, the small circles, into account) is the same, the median for group B is five whereas that for group C is three. Quite clearly the grey areas do not overlap and the difference between the medians is statistically significant. Finally, as was expected, given the ease of communicating via a mouse, direct manipulation received a very high approval rating; the median for each group was five.

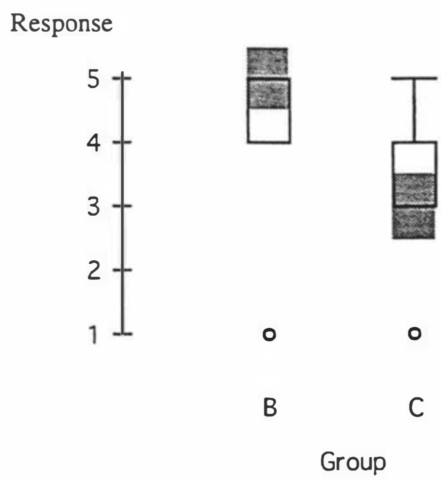


Figure 4 Menu input

With regard to conclusions, such as the likelihood of certain diagnoses, respondents were asked whether they would like them to be presented in English phrases and sentences (Figure 5), graphically or as numerical values on a scale (see Figure 6). If we look at the boxplots of results for the numerical option, the spread of data is much greater for group B, with a median of two, whilst the median is four for group C. The shaded areas do not coincide and the difference between the medians is statistically significant. As expected the more expert users appeared able to interpret numerical output with some ease. Their grounding in the domain, presumably, allows them to translate easily between states of belief and numerical equivalents.

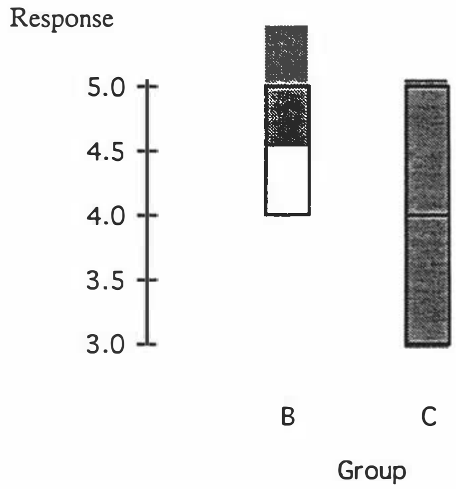


Figure 5 Natural language presentation of output

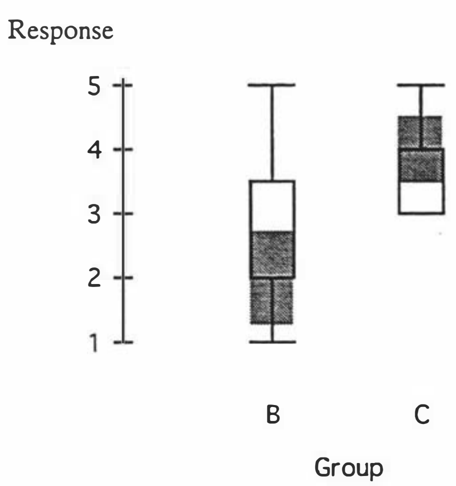


Figure 6 Numeric presentation of output

Users also had to say whether they wanted to engage in dialogue with the system, asking questions whenever they wanted. This is an important issue as much research is currently being carried on in this area. An analysis of the results showed that whilst both groups supported the idea, group C were much keener with the median at five with that for group B considerably lower at three (see Figure 7). As the grey areas do not overlap, the difference between the medians is statistically different. This indicates, perhaps, greater confidence on the part of the more expert group. This result is also in line with the previous finding that group B members are not particularly happy inputting information using natural language. Obviously, dialogue requires active participation and group B members may be reluctant either to type or actually frame questions.

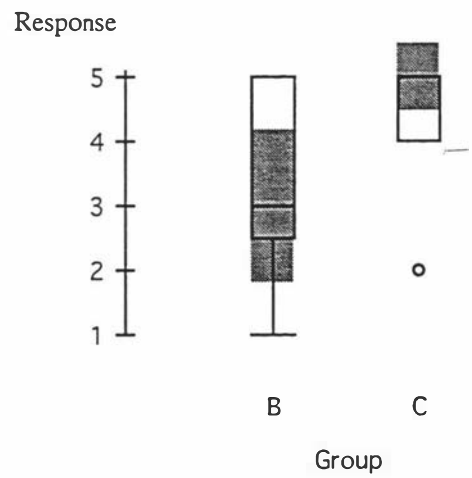


Figure 7 Interactive dialogue

A second question inquired whether users would find information like pictures of diseased leaves helpful. It was anticipated that both the more and less expert groups would feel confident using information in its natural form. This was confirmed as the median for each group was four (see Figure 9). Nonetheless, there still appeared to be a difference between groups. All of group C answered at the 4 or 5 level (the only occasion this happened), whilst the values for group B ranged from 1 to 5. To check whether there was likely to be any dependence between the enthusiasm and the expertise of these users, a contingency table was set up where the number of responses for each group up to and including three were put in one category and those at 4 and 5 in another. A Chi-squared test was then performed. The null hypothesis is that the two factors are statistically independent, that is there is no relationship between expertise and enthusiasm. The value of Chi-square was 4.091 with 1 degree of freedom. The probability that there is no connection is very small (between .05 and .025) so the null hypothesis is rejected. It appears, therefore, that those with expertise were keener than the other users to see knowledge represented in more than one way. It must be stressed that the sample, in this exploratory study, is too small for any definitive conclusions to be drawn but this result indicates that the way information is presented may be of vital importance to one group and not quite so significant for another. Exactly how users want information to be presented may have to be ascertained during the analysis phase. There are implications here both for knowledge acquisition and the architecture of the knowledge-based system.

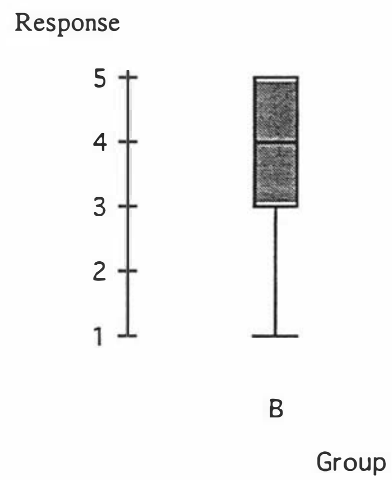


Figure 9 Pictorial Information

Appendix D

Enrolment at Massey University

Enrolment at Massey University

The steps that have to be undertaken by prospective first time students are outlined and then the minor variation on these procedures for students already within the system will be described.

Any intending first time student is asked to send a "request for enrolment pack" card obtained from either the career adviser at school or the Enrolment office at Massey University. These cards can be returned to the university from September onwards. In early December the Enrolment Pack arrives. The Pack contains an enrolment form, an enrolment handbook, tuition fees booklet, lecture and examination timetable and course booklet (usually referred to as a faculty handbook). An application for student allowances and information on the student loan scheme are sent shortly after this. These prospective first time students have to return their enrolment form by January 31. Enrolment applications are accepted after this date but only on payment of a late fee. Students have to enter on the enrolment form not only their personal details but also the papers for which they intend to register. The papers that are listed on the enrolment form must make up an acceptable grouping, permitted by the degree or diploma regulations for the faculty. In those cases where students are allowed to specialise in a subject, the core papers have to be included as well as acceptable supporting papers. Then, in February, final course approval is given by the dean of the chosen faculty in which a student has enrolled. Students have to report in person at a time and place specified in the enrolment handbook.

Students enrolled at Massey receive their packs in the mail during December. The closing date for their enrolment applications is January 10 with the proviso that late applications can be accepted on payment of the appropriate fee. Again students have to attend in person to have their course details confirmed.

Procedures are a little different for extramural students who obviously do not have to come to the campus for course approval. Their applications are dealt with entirely by post. Whether or not a student is internal or extramural, they can make changes in their course up to 31 March. Internal students have to obtain the required signatures from heads of departments.

Various checks have to be made by the administration. Students have to meet certain criteria before they can be accepted as students. They must fulfil at least one of the conditions laid down for enrolment in the University calendar, must have the necessary authorisations if they are classed as international students and must not have been excluded for failing to make sufficient academic progress (either at Massey or other universities) or for some disciplinary offence such as cheating or failing to make the required payments. Previous qualifications from tertiary institutions have to be scrutinised when students move from a course at another institution to one at Massey. Moreover, when fees are paid at the national rate (as opposed to international), students have (in their first year of study) to prove that they are New Zealand citizens or permanent residents. A passport or birth certificate (originals or notarised copy) has to be submitted. Another administrative problem arises when there are limits on the number of students admitted to a paper. In some circumstances, the course controller decides who to accept and in others students are admitted in order of enrolment date until the paper is full.

Information about the work of the Liaison Office was gained through discussions with the Liaison Officer. The author also accompanied two members of staff on a visit to a secondary school. It should be noted that this source of expertise had not been given its due priority in the organisation model and this deficiency had to be remedied.

The activities of the Liaison Office are numerous. Its staff visit schools all round the country (especially those in the vicinity of Massey University) twice a year. In the first visit, seventh form students are given a general picture about what university study at Massey involves. Documents handed out to the seventh formers include "Flying Start". The steps involved in the enrolment process and the criteria for admission to the university are clearly explained in this booklet. Potential students are also referred to the faculty handbook for more up to date information (it is printed later in the year) and the University Calendar to check rules, regulations and course requirements. At the end of the booklet a checklist of key planning dates shows when each stage of enrolment should be completed.

During June and July seventh formers get the opportunity to spend a day at the university. Introductory talks are given by deans of the various faculties about degrees within their disciplines. Further activities are laid on. For instance the Faculty of Business Studies brought back former students to talk about their current jobs. Students are also free to attend lectures on subjects of interest to them. Informally, seventh formers also talk to Massey University students about the papers that they are taking and life on campus.

The seventh formers intending to enrol at Massey University are interviewed by staff of the Liaison Office during a return visit to the school. Before their interview, the seventh formers are advised to plan their course later in detail, using the material in their enrolment pack and if necessary the university calendar.

Interviews take place, individually or in groups (if several are intending to enrol for the same qualification) to provide more specific assistance. It should be pointed out that typically these seventh formers also receive advice from the school's Careers Adviser and/or the seventh form dean. The staff of the Liaison Office at this point try to resolve any outstanding problems. Some students for example cannot decide what course to take: whether to enrol for a BSW or a BEd, for example. Others may have chosen a specialist subject but are unsure what other areas would support this. Whether or not students are clear in their mind about their future course, they are advised to enrol for two first year papers in any subject that they may wish to specialise in subsequently. This gives a student some flexibility if their first choice proves unsatisfactory.

Individual counselling is a very important activity of the Liaison Office. The help given to students before their arrival does not just disappear. In order to provide some continuity, the staff are also available to first year students at Massey University to help them resolve course planning problems.

Appendix E

Enrolment questionnaire

Enrolment Questionnaire

I would be grateful if you could complete the following questionnaire. The information that you provide is confidential and is being collected for research purposes only (in the initial planning of a computerised enrolment system).
Thank you for your co-operation.

Elizabeth Kemp
Department of Computer Science.

Questionnaire

Please fill in the following details:

Q1. Age...

Q2. Sex (Please tick one of the boxes) Male ☐ Female ☐

Q3. Degree or diploma for which enrolled...

Q4. Major(s) for which studying...

Q5. Which papers did you enrol in this year? Please list the paper numbers.

....

....

Q6. Have you any experience with the following computers? Please tick the appropriate box(es).

Macintosh ☐

Other PC ☐

Q7. Have you used any of the following ? Please tick the appropriate box(es)

Word Processor ☐

Database ☐

Spreadsheet ☐

Programming language ☐

Q8. If you have experience with both text and window based computers which do you prefer to use and why?

....

....

Please rate the usefulness of the following sources of information about enrolment on a scale from 1 to 7 where 1 means no help and 7 means very helpful. Circle the appropriate number. **Only rate those sources that you have used.**

	No help						Very helpful
Q9. Enrolment pack	1	2	3	4	5	6	7
Q10. University Calendar	1	2	3	4	5	6	7
Q11. Faculty handbook	1	2	3	4	5	6	7
Q12. Department handbook	1	2	3	4	5	6	7
Q13. Staff of Liaison Office	1	2	3	4	5	6	7
Q14. Other University staff							
Dean	1	2	3	4	5	6	7
Professor	1	2	3	4	5	6	7
Lecturers	1	2	3	4	5	6	7
Q15. Fellow students	1	2	3	4	5	6	7
Q16. Other (please specify)	1	2	3	4	5	6	7

Did you have any of the following problems when enrolling this year? Tick the appropriate box.

	Yes	No
Q17. Resolving lecture and examination clashes?	☐	☐
Q18. Knowing how many papers you can take from outside your faculty?	☐	☐
Q19. Planning a course that you can complete in three years (for example, making sure that you have enough credits to complete a BSc)?	☐	☐
Q20. Knowing which papers you can take next year?	☐	☐
Q21. Planning a course when you have failed a pre-requisite or obtained a restricted pass?	☐	☐

Q22. Did you have any other problems when enrolling?

....

....

....

....

....

....

Q23. If you had access to a computerised enrolment system that gave you help and advice, would you use it ? Please tick the appropriate box.

Yes	No	Possibly
☐	☐	☐

If your answer was "Yes" or "Possibly", how important would it be to have the following facilities ? Please rate their usefulness on a scale from 1 to 7 where 1 means no use and 7 very useful. Circle the appropriate number.

	No Use						Very useful
Q24. Explanations of enrolment terminology?	1	2	3	4	5	6	7
Q25. Information about paper content, work load and assessment?	1	2	3	4	5	6	7
Q26. Help with planning the current year's course?	1	2	3	4	5	6	7
Q27. Assistance in planning for the future?	1	2	3	4	5	6	7
Q28. Information about the cost of papers?	1	2	3	4	5	6	7
Q29 Examples of a typical enrolment for a major?	1	2	3	4	5	6	7
Q30. Possible careers in your chosen field?	1	2	3	4	5	6	7

Q31. List any other facilities you might like to use.

....
....
....
....
....

Appendix F

Enrolment questionnaire spreadsheet

	A	B	C	D	E	F	G	H	I	J	K	L
1	Code	Age	Sex	Degree	Major 1	Major2	Paper-Q	Mac	PC	Wp	DB	Spreadsheet
2	THIRD	20	M	BS	INFO SYS		57.302	Y	Y	Y	Y	Y
3	THIRD	24	F	BA	PSY	INFO SYS	57.302	Y	Y	Y	Y	Y
4	THIRD	41	M	BS	INFO SYS		57.302	Y	Y	Y	Y	Y
5	THIRD	20	M	BS	INFO SYS		57.302	Y	Y	Y	Y	Y
6	THIRD	21	M	BS	INFO SYS		57.302	Y	Y	Y	Y	Y
7	THIRD	22	M	BS	INFO SYS	ACC	57.302	Y	Y	Y	Y	Y
8	THIRD	31	F	BA	INFO SYS	MAORI	57.302	Y	Y	Y	Y	Y
9	THIRD	21	F	BS	INFO SYS	FIN	57.302	Y	Y	Y	Y	Y
10	THIRD	21	F	BS	INFO SYS	MAN	57.302	Y	Y	Y	Y	Y
11	THIRD	30	M	Sc	COMPSCI		57.302		Y	Y	Y	Y
12	THIRD	20	M	BS	INFO SYS		57.302	Y	Y	Y	Y	Y
13	THIRD	26	M	BS	INFO SYS		57.302	Y	Y	Y	Y	Y
14	THIRD	21	M	BS	INFO SYS		57.302	Y	Y	Y	Y	Y
15	THIRD	20	F	BS	INFO SYS	FIN	57.302	Y	Y	Y	Y	Y
16	SECOND*	24	M	BS	INFO SYS		57.302	Y	Y	Y	Y	Y
17	SECOND*	33	F	BS	INFO SYS		57.302		Y	Y		
18	THIRD	36	F	BA	INFO SYS		59HCIY		Y	Y	Y	Y
19	THIRD	21	M	Te	INFO ENG		59HCIY		Y	Y	Y	Y
20	THIRD	26	F	Sc			59HCIY		Y	Y	Y	Y
21	THIRD	16	M	Sc	COMP SCI		59.327	Y	Y	Y	Y	Y
22	THIRD	24	M	Sc	COMP SCI	PHIL*	59.327	Y	Y	Y		Y
23	THIRD	24	F	BA	INFO SYS		59.327	Y	Y	Y	Y	Y
24	THIRD	19	M	Sc	COMPSCI		59.327	Y	Y	Y	Y	Y
25	THIRD	30+	M	BA	COMP SCI		59.327	Y	Y	Y	Y	
26	THIRD	21	M	BA	COMP SCI		59.327	Y	Y	Y	Y	Y
27	THIRD	20	M	BS	INFO SYS		59.327	Y	Y	Y	Y	Y
28	THIRD	32	M	Sc	COMPSCI	*	59.327	Y	Y	Y	Y	Y
29	THIRD	23+	M	Sc	COMP SCI		59.327	Y	Y		Y	Y
30	THIRD	37	M	Sc	COMPSCI		59.327	Y	Y	Y	Y	Y
31	THIRD	21	M	Sc	COMP SCI		59.327	Y	Y	Y	Y	Y
32	THIRD	21	M	Sc	COMP SCI		59.327	Y	Y	Y	Y	Y
33	THIRD	40	M	Sc	COMP SCI	H&SAFETY	59.327	Y	Y	Y		Y
34	THIRD	33	M	BA	COMP SCI		59.327	Y	Y	Y	Y	Y
35	THIRD	25	M	BA	COMP SCI		59.327	Y	Y	Y	Y	Y
36	THIRD	24	M	BS	INFO SYS		59.327	Y	Y	Y	Y	Y
37	SEC or THIRD	22	M	Sc	COMP SCI		59.327	Y	Y	Y	Y	Y
38	THIRD	22	M	Sc	COMP SCI		59.327	Y	Y	Y	Y	Y
39	THIRD	21	M	Sc	COMP SCI		59.327	Y	Y	Y	Y	Y
40	THIRD	20	F	Sc	COMP SCI		59.327	Y	Y	Y	Y	Y
41	THIRD	22	F	Sc	COMP SCI		59.327	Y	Y	Y	Y	Y
42	THIRD	23	F	Sc	COMP SCI		59.327	Y	Y	Y	Y	Y
43	THIRD	20	F	Sc	COMP SCI		59.327	Y	Y	Y	Y	Y
44	THIRD	20	M	Sc	COMP SCI		59.327	Y	Y	Y	Y	Y
45	THIRD	22	F	BA	PSY		75.305					
46	THIRD	21	M	BA	PSY	EDUC	75.305					
47	THIRD	20	F	BA	PSY		75.305					
48	THIRD	20	F	BA	PSY		75.305	Y		Y		
49	THIRD	20	F	BA	PSY		75.305	Y	Y	Y		Y
50	THIRD	23	F	BA	PSY	EDUC	75.305			Y		
51	THIRD	22	F	BA	PSY		75.305	Y		Y	Y	Y
52	THIRD	23	F	BA	PSY	EDUC	75.305		Y			
53	THIRD	21	F	BA	PSY		75.305	Y	Y	Y	Y	Y
54	SECOND	21	F	BA	SOC W*		75.305	Y		Y	Y	Y
55	THIRD	23	M	Sc	PSY		75.305	Y	Y	Y		Y
56	THIRD	21	M	BA	PSY		75.305	Y		Y	Y	Y
57	THIRD	21	F	BA	SOC W*		75.305		Y	Y		Y
58	THIRD	35	M	BA	PSY		75.305	Y	Y	Y		
59	THIRD*	21	F	Sc	PSY		75.305					
60	THIRD	21	F	Sc	PSY		75.305		Y	Y		
61	THIRD	21	F	BA	PSY		75.305		Y			
62	THIRD	21	F	BA	PSY	EDUC	75.305					
63	THIRD	22	F	Sc	PSY		75.305	Y	Y	Y		Y
64	THIRD	40+	F	BS	STATS	OR	60304	Y	Y	Y		Y
65	THIRD	20	M	Sc	MATHS		60304	Y		Y		Y
66	THIRD	20	M	Sc	MATHS	COMP SCI	60304	Y	Y	Y	Y	Y
67	THIRD	22	M	Sc	OR		60304	Y	Y	Y	Y	Y
68	THIRD	21	F	Sc	MATHS	STATS	60304	Y	Y	Y	Y	Y
69	THIRD	20	F	Sc	MATHS		60304	Y	Y			Y
70	THIRD	20	F	Sc	MATHS		60304	Y	Y	Y		Y
71	THIRD	22	F	BA	MATHS		60304	Y	Y	Y	Y	Y

	A	B	C	D	E	F	G	H	I	J	K	L
72	THIRD	24	M	BA	MATHS	LING	60304	Y		Y		
73	THIRD	23	F	Sc	COMP SCI	MATHS	60304	Y	Y	Y	Y	Y
74	THIRD	21	M	Sc	MATHS		60304	Y	Y	Y		
75	SECOND	21	M	Sc	MATHS	PSY	60304	Y	Y	Y		Y
76	THIRD	21	F	Sc	COMP SCI		60304	Y	Y		Y	
77	THIRD	20	F	BA	GEOG*		45.323		Y	Y	Y	Y
78	THIRD	20	M	Sc	GEOG		45.323	Y	Y	Y	Y	
79	THIRD	21	M	BA	GEOG		45.323		Y			Y
80	THIRD	25	M	Sc	GEOG		45.323	Y		Y	Y	Y
81	THIRD	23	M	Sc	GEOG		45.323					
82	THIRD	21	M	BA	GEOG*		45.323		Y	Y		
83	THIRD	20	M	Sc	COMP SCI	STATS	45.223	Y	Y	Y	Y	Y
84	THIRD	21	M	BA	*		45.223		Y	Y		
85	SECOND	19	F	BA	REG PLAN	VAL	45.223	Y		Y	Y	Y
86	SECOND	19	F	BA	PLAN*	ENV	45.223	Y				Y
87	THIRD	22	M	BA	PSY		45.223	Y	Y	Y	Y	Y
88	THIRD	22	F	Sc	BOT		45.223	Y		Y		Y
89	SECOND	19	F	BA	GEOG		45.223					
90	SECOND	20	F	BA	GEOG		45.223		Y	Y		
91	SECOND	19	F	Sc	GEOG		45.223	Y	Y	Y		Y
92	SECOND	20	M	BA	*		45.223					
93	SECOND	20	F	BA	GEOG*		45.223	Y	Y	Y	Y	Y
94	SECOND	20	M	BA	GEOG*		45.223	Y	Y	Y	Y	Y
95	SECOND	19	M	BA	GEOG*		45.223	Y			Y	
96	SECOND	19	M	BA	GEOG*		45.223	Y	Y	Y	Y	Y
97	SECOND	19	M	Sc	ZOO	GEOG	45.223		Y	Y		
98	SECOND	19	F	Sc	CHEM		45.223	Y		Y	Y	Y
99	SECOND	21	F	Sc	GEOG	ENV	45.223		Y	Y		
100	SECOND	39	M	BA	BOT/GEOG		45.223	Y	Y	Y	Y	Y
101	SECOND	45	M	Sc	GEOG		45.223		Y	Y		Y
102	SECOND	21	M	BA	GEOG*		45.223					
103	SECOND	19	F	BS	INFO SYS	AG BUS	57.202	Y	Y	Y		Y
104	SECOND	32	M	BS	INFO SYS		57.202	Y	Y	Y	Y	Y
105	SECOND	29	M	BS	INFO SYS	FIN	57.202	Y	Y	Y	Y	Y
106	SECOND	20	M	BS	INFO SYS	MARK	57.202	Y	Y	Y	Y	Y
107	SECOND	22	F	BS	INFO SYS	COMM MA	57.202	Y	Y	Y		Y
108	SECOND	29	F	BA	INFO SYS		57.202	Y	Y	Y	Y	Y
109	SECOND	27	F	BS	INFO SYS	FIN	57.202	Y	Y	Y	Y	Y
110	SECOND	29	F	BA	INFO SYS		57.202	Y	Y	Y	Y	Y
111	SECOND	22	F	BS	INFO SYS	ACC	57.202	Y	Y	Y		Y
112	THIRD	22	M	BA	PHIL	HIST	57.202	Y	Y	Y	Y	Y
113	FIRST	19	M	BS	INFO SYS		57.202	Y	Y	Y	Y	Y
114	SECOND	19	M	BS	INFO SYS		57.202	Y	Y	Y	Y	Y
115	SECOND		M	BS	INFO SYS	MAN	57.202	Y	Y	Y	Y	Y
116	SECOND	28	M	BS	INFO SYS	TOUR	57.202	Y	Y	Y	Y	Y
117	SECOND	23	F	BS	INFO SYS	FIN	57.202	Y	Y	Y		Y
118	THIRD	30S	M	OTHER			57.202	Y	Y	Y	Y	
119	SECOND	20	M	BS	INFO SYS	FIN	57.202	Y	Y	Y	Y	Y
120	SECOND	20	M	BS	INFO SYS	MAN	57.202	Y	Y	Y		Y
121	SECOND	20	F	BA	INFO SYS	COMP SCI	57.202		Y	Y	Y	Y
122	SECOND	19	F	BS	INFO SYS	ACC	57.202	Y	Y	Y	Y	Y
123	SECOND	19	M	BS	INFO SYS	ACC	57.202	Y	Y	Y		Y
124	SECOND	19	M	BS	COMP SCI	FIN	57.202	Y	Y	Y	Y	Y
125	SECOND	20	M	BS	INFO SYS	PROP	57.202	Y	Y	Y	Y	Y
126	SECOND	24	M	BS	INFO SYS	FIN	57.202	Y	Y	Y	Y	Y
127	SECOND	19	F	BS	INFO SYS		57.202	Y	Y	Y	Y	Y
128	SECOND	19	M	BS	AGBUS		57.202	Y	Y	Y	Y	Y
129	SECOND	25	M	BS	MARK		57.202	Y	Y	Y	Y	Y
130	SECOND	20	M	Sc	INFO SYS		57.202	Y	Y	Y	Y	Y
131	THIRD	27	M	BS	INFO SYS	MAN	57.202	Y	Y	Y		Y
132	SECOND	21	M	BS	INFO SYS		57.202	Y	Y	Y	Y	Y
133	SECOND	20	F	BS	INFO SYS	MARK	57.202	Y	Y	Y	Y	Y
134	SECOND	27	M	BS	INFO SYS	MARK	57.202	Y	Y	Y	Y	Y
135	SECOND	22	M	BA	INFO SYS	JAP	57.202	Y	Y	Y	Y	Y
136	SECOND	34	F	BS	INFO SYS	ACC	57.202	Y	Y	Y		Y
137	SECOND		F	BS	INFO SYS		57.202	Y	Y	Y	Y	Y
138	THIRD	20	F	Sc	ECOL		20.205	Y	Y	Y	Y	Y
139	SECOND	29	M	Sc	GEOG		20.205	Y	Y	Y		Y
140	THIRD	21	M	Sc	BOT		20.205					
141	SECOND	19	F	Sc	PLASCI	ENV SCI	20.205	Y		Y	Y	Y

	A	B	C	D	E	F	G	H	I	J	K	L
142	SECOND	21	F	Sc	BIOL		20.205	Y				Y
143	THIRD	21	F	Sc	ZOO		20.205	Y				
144	THIRD	35	M	Sc	BIOL	SOC	20.205		Y	Y	Y	Y
145	SECOND	19	F	Sc	ZOO	BOT	20.205	Y	Y	Y	Y	Y
146	THIRD	22	F	Sc	ECOL	PLA SCI	20.205	Y	Y	Y	Y	Y
147	THIRD	20	F	Sc	ECOL	PLA SCI	20.205	Y	Y	Y		Y
148	SECOND	25	M	Sc	ZOO	GEOL	20.205	Y				
149	SECOND	21	F	Sc	ECOL		20.205	Y	Y	Y		
150	THIRD	20	F	AH	TECH ENG		20.205		Y	Y	Y	Y
151	SECOND	21	M	Sc	BOT	ZOO	20.205		Y	Y		Y
152	SECOND	20	F	Sc	PLA SCI		20.205	Y		Y	Y	Y
153	THIRD	23	M	Sc	ECOL		20.205	Y		Y		Y
154	THIRD	24	F	Sc	ZOO		20.205	Y	Y			
155	THIRD	20	M	AH	SOIL SCI		20.205		Y		Y	Y
156	SECOND	35	M	Sc	ZOO	BOT	20.205		Y			
157	THIRD	24	M	AH	ECOL		20.205	Y	Y	Y	Y	Y
158	THIRD	20	F	Sc	ZOO		20.205	Y		Y		
159	THIRD	21	M	Sc	BIOL		20.205		Y	Y		Y
160	THIRD	22	M	Sc	ZOO		20.205	Y	Y	Y		Y
161	THIRD	20	F	Sc	PLA SCI		20.205					
162	SECOND	20	F	Sc	PLA BIOL		20.205		Y	Y		
163	SECOND	21	M	Sc	BIOL		20.205		Y	Y		
164	SECOND	19	F	Sc	ZOO		20.205	Y				
165	SECOND	19	F	Sc	ZOO	BIOL	20.205	Y		Y	Y	Y
166	SECOND	19	F	Sc	ZOO	PLA SCI	20.205			Y		
167	SECOND	19	M	Sc	ZOO		20.205	Y				Y
168	SECOND	19	M	Sc	BOT	ZOO	20.205	Y		Y		
169	SECOND	19	M	Sc	ZOO		20.205	Y		Y		Y
170	SECOND	19	M	Sc	ZOO		20.205	Y	Y	Y	Y	Y
171	SECOND	19	F	Sc	ZOO	BIOL	20.205	Y				
172	SECOND	20	M	Sc	PLA SCI		20.205	Y		Y		
173	SECOND	21	M	Sc	PLA SCI		20.205		Y	Y		Y
174	SECOND	21	F	Sc	CHEM	MAORI	20.205	Y				
175	THIRD	21	F	Sc	BOT		20.205	Y	Y	Y		Y
176	SECOND	35	F	Sc	BOT	PSY	20.205					
177	SECOND	20	F	Sc	ZOO	PLA HEA	20.205	Y		Y		
178	THIRD	23	F	BS	ZOO		20.205	Y	Y	Y	Y	Y
179	THIRD	29	M	Sc	ZOO		20.205		Y			
180	SECOND	19	F	Sc	ZOO		20.205	Y	Y	Y		
181	SECOND	19	M	Te	CSE		59.211	Y	Y	Y	Y	Y
182	SECOND	21	F	Sc	COMPSCI		59.211	Y	Y	Y	Y	Y
183	SECOND	20	M	Te	INFOENG		59.211	Y	Y	Y	Y	Y
184	SECOND		M	Sc	COMP SCI		59.211	Y	Y	Y	Y	Y
185	THIRD	25	M	BA	INFO SYS		59.211	Y	Y	Y	Y	Y
186	SECOND	20	M	Te	CSE		59.211	Y	Y	Y	Y	Y
187	SECOND	19	M	BA	COMP SCI	PHIL	59.211	Y		Y		Y
188	SECOND	20	M	BA	COMP SCI		59.211	Y		Y	Y	Y
189	SECOND	21	M	Sc	COMP SCI		59.211	Y	Y	Y	Y	Y
190	SECOND	21	M	Sc	COMP SCI	ELECT	59.211	Y	Y	Y	Y	Y
191	SECOND	18	M	BA	COMP SCI		59.211	Y	Y	Y	Y	Y
192	SECOND	19	F	Sc	COMP SCI	MATHS	59.211	Y	Y	Y	Y	Y
193	SECOND	18	M	Sc	COMP SCI	ELECT	59.211	Y	Y	Y	Y	Y
194	SECOND	23	M	Sc	COMP SCI		59.211	Y		Y	Y	Y
195	SECOND	19	M	Te	CSE		59.211	Y	Y	Y	Y	Y
196	SECOND	19	M	Te	INFO ENG		59.211	Y	Y	Y	Y	Y
197	SECOND	20	M	Te	CSE		59.211	Y	Y	Y	Y	Y
198	SECOND	19	M	Sc	COMP SCI	ELECT	59.211	Y	Y	Y	Y	Y
199	SECOND	19	M	Te	INFO ENG		59.211	Y	Y	Y	Y	Y
200	SECOND	21	M	BA	REL STUD	COMP SCI	59.211	Y	Y	Y	Y	Y
201	SECOND	19	M	Te			59.211		Y	Y	Y	Y
202	SECOND	19	M	Te	INFO ENG		59.211		Y	Y		
203	SECOND	22	F	Sc	MATH	COMP SCI	59.211	Y	Y	Y	Y	Y
204	SECOND	21	M	Sc	MATH	COMP SCI	59.211	Y	Y	Y	Y	Y
205	THIRD	21	F	Sc	PHYS	MATH	59.211	Y				
206	THIRD	20	M	Te	CSE		59.211	Y	Y	Y		Y
207	SECOND	19	M	Te	CSE		59.211	Y	Y	Y	Y	Y
208	SECOND	20	M	Sc	COMPSCI	ELECT	59.211	Y	Y	Y	Y	Y
209	SECOND	22	M	Sc	COMP SCI		59.211	Y	Y	Y	Y	Y
210	SECOND	20	M	Sc	COMP SCI	MATH	59.211	Y	Y	Y	Y	Y
211	SECOND	20	M	Sc	COMP SCI	MATH	59.211	Y	Y	Y	Y	Y

	A	B	C	D	E	F	G	H	I	J	K	L
212	SECOND	20	F	Sc	COMP SCI	PSY	59.211		Y	Y	Y	Y
213	SECOND	20	M	Sc	MATHS	STATS	59.211	Y	Y	Y		Y
214	SECOND	19	M	Te	CSE		59.211		Y	Y	Y	Y
215	SECOND	39	M	Sc	PHYSIOL		59.211	Y				
216	SECOND	47	F	BA	OP RES		59.211	Y	Y	Y	Y	Y
217	SECOND	19	M	Te	CSE		59.211	Y	Y	Y		Y
218	FIRST	26	M	BA	INFO SYS*	SOC	57.102	Y	Y	Y	Y	Y
219	FIRST	18	F	BA(AR)	MEDIA	PSY	57.102	Y	Y	Y	Y	Y
220	FIRST	18	F	BA	PSY	SOC	57.102	Y	Y	Y	Y	Y
221	THIRD	33	M	AH			57.102	Y	Y	Y		
222	FIRST		M	Sc	COMP	SCI&MAN	57.102	Y	Y	Y	Y	Y
223	FIRST	25	F	VET			57.102	Y	Y	Y	Y	Y
224	FIRST	19	M	VET			57.102	Y	Y	Y	Y	Y
225	SECOND	24	M	BA(AR)	JAP		57.102	Y	Y	Y	Y	Y
226	FIRST	18	F	VET			57.102	Y		Y	Y	Y
227	FIRST	18	M	BA	GEOG		57.102	Y	Y	Y	Y	Y
228	FIRST	18	M	BA	GEOG*		57.102	Y		Y	Y	Y
229	SECOND	25	M	AH	RUR VAL		57.102	Y		Y	Y	Y
230	FIRST	19	M	Sc	ENV SCI	GEOG	57.102	Y		Y	Y	Y
231	FIRST	19	F	VET			57.102	Y	Y	Y	Y	Y
232	FIRST	18	F	VET			57.102	Y	Y	Y	Y	Y
233	FIRST	19	F	BA			57.102	Y		Y	Y	Y
234	FIRST	18	F	BA	ENG	GER	57.102	Y	Y	Y	Y	Y
235	THIRD	22	M	AH			57.102	Y	Y	Y	Y	Y
236	FIRST	19	F	BA	MATH		57.102	Y		Y	Y	Y
237	FIRST	30	M	BA	INFO SYS		57.102	Y	Y	Y	Y	Y
238	FIRST	18	F	BA	*		57.102	Y	Y	Y	Y	Y
239	FIRST	20	F	Sc	GEOG		57.102	Y	Y	Y	Y	Y
240	THIRD	21	F	ED	EDUC		57.102	Y	Y	Y	Y	Y
241	FIRST	35	F	BA	PSY		57.102	Y	Y	Y	Y	Y
242	FIRST	25	F	Sc			57.102	Y	Y	Y	Y	Y
243	FIRST	19	M	VET			57.102	Y	Y	Y	Y	Y
244	FIRST	18	M	BA	PSY		57.102	Y		Y	Y	Y
245	FIRST	31	M	BA			57.102	Y	Y	Y	Y	Y
246	FIRST	18	F	BA	ECON		57.102	Y		Y	Y	Y
247	FIRST	18	F	VET			57.102	Y		Y	Y	Y
248	FIRST	18	F	VET			57.102	Y		Y	Y	Y
249	SECOND	19	F	BA	EDUC	PSY	57.102	Y	Y	Y	Y	Y
250	FIRST	18	M	VET			57.102	Y		Y	Y	Y
251	FIRST	18	M	VET			57.102	Y	Y	Y	Y	Y
252	FIRST	18	M	BA	*		57.102	Y	Y	Y	Y	Y
253	FIRST	18	M	Sc			57.102	Y	Y	Y	Y	Y
254	FIRST	18	M	VET			57.102	Y	Y	Y	Y	Y
255	FIRST	18	F	VET	CHEM	BIOL	57.102	Y		Y	Y	Y
256	FIRST	19	M	VET			57.102	Y	Y	Y	Y	Y
257	FIRST	20	M	BA	PSY		57.102	Y	Y	Y	Y	Y
258	FIRST	19	M	OTHER	INTARCH*		57.102	Y		Y	Y	Y
259	FIRST	18	M	Sc	GEOG		57.102	Y	Y	Y	Y	Y
260	FIRST	19	F	VET			57.102	Y	Y	Y	Y	Y
261	FIRST	19	F	VET			57.102	Y	Y	Y	Y	Y
262	FIRST	19	M	VET			57.102	Y	Y	Y	Y	Y
263	FIRST	19	M	VET	BIOL		57.102	Y	Y	Y	Y	Y
264	FIRST	18	F	BA	PSY		57.102	Y	Y	Y	Y	Y
265	FIRST	20	F	Sc	BIOCHEM		57.102	Y	Y	Y	Y	Y
266	FIRST	19	M	OTHER	MED*		57.102	Y		Y	Y	Y
267	FIRST	19	F	Sc	BIOCHEM	PHYSIO	57.102	Y	Y	Y	Y	Y
268	FIRST	17	F	OTHER	MED*		57.102	Y	Y	Y	Y	Y
269	FIRST	21	F	BA(AR)	GER	FREN	57.102	Y	Y	Y	Y	Y
270	FIRST	21	F	BA	PSY		57.102	Y	Y	Y	Y	Y
271	FIRST	24	F	BA(AR)	LINGS		57.102	Y	Y	Y	Y	Y
272	FIRST	31	M	BA	SOC	SOC ANTH	57.102	Y	Y	Y	Y	Y
273	FIRST	19	F	BA	PSY		57.102	Y	Y	Y	Y	Y
274	FIRST	38	F	CP			57.102	Y	Y	Y	Y	Y
275	FIRST	20	M	AH	TECH		60.103	Y	Y	Y	Y	Y
276	FIRST	20	F	AH	TECH		60.103		Y	Y		
277	FIRST	19	M	AH			60.103	Y		Y		
278	FIRST	24	M	AH	HORT		60.103	Y				Y
279	FIRST	18	M	AH			60.103		Y	Y		Y
280	FIRST	19	M	AH			60.103		Y	Y	Y	Y
281	FIRST	18	M	AH			60.103		Y	Y	Y	Y

	A	B	C	D	E	F	G	H	I	J	K	L
282	FIRST	20	F	AH			60.103		Y	Y	Y	Y
283	FIRST	18	F	AH	TECH		60.103		Y	Y	Y	Y
284	FIRST	19	M	AH			60.103		Y	Y		Y
285	FIRST	20	M	AH	TECH		60.103		Y		Y	Y
286	FIRST	31	M	Sc	COMP		60.103	Y	Y	Y	Y	Y
287	FIRST	19	M	Sc	CHEM	BIO	60.103					
288	FIRST	26	M	Sc	CHEM	BIO	60.103					
289	FIRST	18	M	AH			60.103	Y	Y			Y
290	FIRST	18	M	AH			60.103	Y	Y	Y		Y
291	FIRST	18	M	AH			60.103			Y		
292	FIRST	18	F	AH	ECON		60.103	Y	Y	Y		Y
293	THIRD	20	F	Sc	ECOL		60.103	Y	Y	Y		Y
294	FIRST	24	F	Sc			60.103	Y		Y	Y	Y
295	FIRST	19	M	AH			60.103			Y		
296	FIRST	19	M	AH			60.103		Y	Y		Y
297	FIRST	19	F	AH			60.103	Y		Y		
298	FIRST	21	M	BA	COMP SCI		60.103	Y	Y	Y	Y	Y
299	FIRST	19	M	AH	TECH		60.103		Y		Y	Y
300	FIRST	20	M	AH			60.103		Y	Y	Y	Y
301	FIRST	19	M	AH			60.103		Y			Y
302	FIRST	17	M	AH			60.103		Y	Y		Y
303	FIRST	18	F	AH	SCIENCE		60.103		Y			Y
304	FIRST	18	F	AH			60.103		Y	Y		
305	SECOND	20	M	BA	PSY		60.103	Y	Y	Y	Y	Y
306	FIRST	20	M	Sc	COMP SCI		60.103	Y	Y	Y	Y	Y
307	FIRST		F	AH			60.103					
308	FIRST	19	M	AH			60.103					
309	FIRST	18	M	AH			60.103	Y		Y		
310	FIRST	19	M	AH			60.103		Y			
311	FIRST	19	M	BA			60.103	Y	Y	Y	Y	Y
312	THIRD	21	M	BS	MARK		60.103	Y	Y	Y		Y
313	FIRST	23	M	Sc	CHEM		60.103	Y		Y	Y	Y
314	THIRD	23	M	BS	MARK		60.103	Y	Y	Y	Y	Y
315	FIRST	21	M	BS	MARK		60.103	Y	Y	Y	Y	Y
316	THIRD	21	F	BS	MAN		60.103	Y	Y	Y		Y
317	FIRST	19	M	AH			60.103	Y	Y	Y	Y	Y
318	THIRD	20	M	BS	MARK		60.103	Y	Y	Y	Y	Y
319	FIRST	22	M	Te	CSE		60.103	Y	Y	Y	Y	Y
320	FIRST	21	M	Sc	COMP SCI		60.103	Y	Y	Y	Y	Y
321	THIRD	21	F	BA	GEOG		60.103	Y	Y	Y		
322	FIRST		F	Sc	ZOO		60.103					
323	FIRST	22	F	Sc	COMP SCI		60.103	Y	Y	Y	Y	Y
324	FIRST	27	M	AH			60.103		Y	Y	Y	Y
325	FIRST	19	M	AH			60.103		Y	Y	Y	Y
326	FIRST	18	F	Sc	ECOL		60.103			Y		
327	FIRST	19	M	AH	ECON		60.103	Y		Y	Y	Y
328	FIRST	18	F	AH			60.103			Y		Y
329	FIRST	18	M	AH	TECH		60.103	Y	Y	Y	Y	Y
330	FIRST	18	M	Sc	PSY		60.103	Y	Y	Y	Y	Y
331	FIRST	19	F	AH			60.103	Y		Y	Y	Y
332	SECOND	20	F	Sc	MOL BIOL	GEN	60.103	Y		Y		
333	FIRST	18	M	AH	ECON		60.103	Y		Y		Y
334	FIRST	26	F	Sc	MICROBIOL		60.103	Y	Y	Y	Y	Y
335	FIRST	18	F	AH			60.103	Y	Y	Y	Y	Y
336	FIRST	27	F	CP			60.103			Y		
337	SECOND	21	F	B ED	EDUC		60.103	Y		Y	Y	Y
338	FIRST	25	F	Sc			60.103	Y	Y	Y	Y	Y
339	FIRST	20	M	Sc	PHYS	COMP SCI	60.103	Y	Y	Y		Y
340	SECOND	20	M	Vet	BIOCHEM	MOL GEN	60.103	Y		Y		
341	SECOND	20	F	BA	ECON		60.103			Y		
342	THIRD		M	BS	ACC		60.103	Y	Y	Y	Y	Y
343	FIRST	29	F	BS	ACC		60.103	Y	Y	Y	Y	Y
344	FIRST	18	F	Te	CSE		112	Y	Y	Y	N	Y
345	FIRST	18	M	Te	INFO ENG		112	Y	Y	Y		Y
346	FIRST	18	M	Sc	COMPSC		112	Y	Y	Y	Y	Y
347	FIRST	18	M	Sc	COMPSC	ELECTC	112	Y		Y	Y	Y
348	FIRST	18	M	Sc	PHYSICS	ELECTC	112	Y	Y	Y	Y	Y
349	SECOND	20	F	Sc	COMPSC	MATHS	112	Y	Y	Y	Y	Y
350	FIRST	19	F	Sc	COMPSC	PHYSICS	112	Y	Y	Y	Y	Y
351	FIRST	27	M	Sc	COMPSC		112	Y		Y		Y

	A	B	C	D	E	F	G	H	I	J	K	L
352	FIRST	18	M	Te	COMPSC	SYSTENG	112	Y	Y	Y	Y	Y
353	FIRST	23	F	Sc	COMPSC	ZOO/GEN	112	Y		Y		Y
354	FIRST	19	M	Te	COMPSC		112	Y	Y	Y		Y
355	FIRST	18	M	Sc	MATHS		112	Y	Y	Y		Y
356	FIRST	19	M	Te	CSE		112	Y	Y	Y	Y	Y
357	FIRST	19	M	Te	CSE		112	Y	Y	Y	Y	Y
358	FIRST	18	M	Te	CSE		112	Y		Y	Y	Y
359	FIRST	18	M	Sc	COMP SCI	ELECT	112	Y	Y	Y	Y	Y
360	FIRST	18	M	Sc	MATH PHYS		112	Y	Y	Y		Y
361	FIRST	20	F	Sc	MATH	COMP SC	112	Y	Y		Y	Y
362	SECOND	22	F	Sc	COMP SCI		112	Y	Y	Y	Y	Y
363	FIRST	18	M	OTHER	LECT ENG*	ELECTR EN	112	Y	Y	Y	Y	Y
364	FIRST	18	M	Te	CSE		112	Y	Y	Y	Y	Y
365	FIRST	18	M	Te	INFO ENG		112	Y		Y	Y	Y
366	FIRST	22	M	Sc	COMP SCI	MAN SCI	112	Y	Y	Y	Y	Y
367	FIRST	25	M	Sc	COMP SCI		112	Y	Y	Y	Y	Y
368	FIRST	19	M	Sc	MATHS		112	Y	Y	Y	Y	Y
369	FIRST	27	M	Sc	COMP SCI		112	Y	Y	Y	Y	Y
370	FIRST	26	M	Sc	COMP SCI		112	Y	Y	Y	Y	Y
371	SECOND	21	M	Sc	MATHS		112	Y				
372	FIRST	18	M	Te	INFO ENG		112	Y	Y	Y	Y	Y
373	FIRST	20	M	Sc	COMP SCI		112	Y				Y
374	FIRST	19	F	BA	COMP SCI	MATHS	112	Y	Y			Y
375	FIRST	28	F	Sc	COMP SCI		112	Y				Y
376	FIRST	29	F	OTHER	COMP SCI		112		Y	Y	Y	Y
377	FIRST	19	M	OTHER	MED*		112	Y	Y	Y	Y	Y
378	FIRST	18	M	Te	CSE		112	Y	Y	Y	Y	Y
379	FIRST	19	M	Sc	COMP SCI		112	Y	Y	Y	Y	Y
380	FIRST	19	M	Te	INFO ENG		112	Y		Y	Y	Y
381	FIRST	19	M	Te	CSE		112	Y	Y	Y		Y
382	FIRST	19	M	Te	INFO ENG		112	Y	Y	Y	Y	Y
383	FIRST	19	M	Sc	COMP SCI	MATHS	112		Y	Y		Y
384	FIRST	19	M	Te	CSE		112	Y	Y	Y	Y	Y
385	FIRST	19	M	VET			112	Y	Y	Y	Y	Y
386	FIRST	18	F	Sc	MATHS		112	Y		Y	Y	Y
387	FIRST	18	F	Sc	MATHS		112	Y		Y	Y	Y
388	FIRST	18	F	Sc	MATHS		112	Y		Y	Y	Y
389	FIRST	18	M	Sc	COMP SCI	ELECTRON	112	Y	Y	Y	Y	Y
390	FIRST	19	F	Sc	COMP SCI	PSY	112	Y		Y	Y	Y
391	FIRST	18	M	OTHER	ENG*		112		Y	Y	Y	Y
392	FIRST	19	M	Sc	COMP SCI		112	Y		Y	Y	Y
393	FIRST	19	M	Sc	COMP SCI	ELECTRON	112	Y	Y	Y	Y	Y
394	FIRST	21	F	Te	CSE		112	Y	Y	Y	Y	Y
395	FIRST	18	M	BA	COMP SCI	PSY	112	Y	Y	Y	Y	Y
396	FIRST	18	M	Te	INFO ENG		112	Y		Y	Y	Y
397	FIRST	18	M	OTHER	MATHS	PHYS	112	Y	Y	Y	Y	Y
398	FIRST	18	M	Te	CSE		112		Y	Y	Y	Y
399	FIRST	18	F	BA	PSY		112	Y				
400	FIRST	19	F	Te	INFO ENG		112	Y			Y	Y
401	FIRST	18	F	Te	OPS RES		112	Y	Y	Y	Y	Y
402	FIRST	19	F	Te	INFO ENG		112	Y	Y	Y	Y	Y
403	FIRST	19	F	Te	OPS RES		112	Y	Y	Y	Y	Y
404	FIRST	19	M	Te	INFO ENG		112	Y	Y	Y		Y
405	FIRST	20	M	BA	COMP SCI		112	Y		Y		
406	FIRST	19	M	Te	INFO ENG		112	Y	Y	Y		Y
407	FIRST	18	M	Te	INFO ENG		112	Y	Y	Y	Y	Y
408	FIRST	18	M	Te	CSE		112	Y		Y	Y	Y
409	FIRST	18	M	Te	CSE		112	Y		Y		Y
410	FIRST	20	M	Sc	COMP SCI		112	Y	Y	Y	Y	Y
411	FIRST	21	F	Sc	COMP SCI		112	Y	Y	Y	Y	Y
412	FIRST	19	M	Te	CSE		112	Y	Y	Y	Y	Y
413	FIRST	18	M	Te	CSE		112	Y		Y		Y
414	FIRST	20	F	Sc	COMP SCI		112	Y		Y		
415	FIRST	20	M	Te	CSE		112	Y	Y	Y	Y	Y
416	FIRST	22	M	Sc	COMP SCI	MATHS	112	Y		Y	Y	Y
417	FIRST	19	M	Te	INFO ENG		112	Y		Y	Y	Y
418	FIRST	30	M	Sc	COMP SCI		112					
419	FIRST	19	M	Te	INFO ENG		112	Y		Y	Y	Y
420	FIRST	18	M	Te	INFO ENG		112	Y	Y	Y	Y	Y
421	FIRST	23	F	Te	CSE		112	Y	Y	Y	Y	Y
422	FIRST	20	M	Sc	COMP SCI	PHYS	112	Y	Y	Y		Y

	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	Prog lang	text	Window	Enrolment	UC	FH	DH	Liaison	Dean	Proflect		Students	Other
2	Y		Y	3		3	3			3	4	4	
3	Y	Y		7	5	7	5	3	6	5	5	4	
4	Y	Y		1	1	2	2	1	4	1	7	2	
5	Y	Y		7		7	7	5			5	5	5(RELS)
6	Y			6	6	6	6	5	6	6	6	6	
7	Y			2		4					6	6	
8	Y	Y		6			6				7	7	
9	Y	Y		3	5	2	2	4	3	3	6	5	
10	Y		Y	6		5					6	6	
11	Y		Y										6 (CN)
12	Y		Y	3	3	6	6	4	2	2	5	5	
13	Y	B	B	3	6	4	4	2	2	2	6	6	
14	Y	Y		4	3	5	5	3	2	2	5	3	
15	Y		Y	7	5	4	6	5	5	5	7	4	
16	Y	Y		5	1	3	7	2	1	1	1	6	
17	Y			7	7	7					7	6	
18	Y		Y	5		6	6					7	
19	Y		Y	6	4	6				6	7	6	
20	Y		Y	5		5	5			7	6		
21	Y		Y	3	6	6	4		7	7	7		
22	Y		Y	6		5			5	5	5	4	
23	Y	B	B	5	3	7	7	7	1	1	5	5	
24	Y		Y			5	4		5		5		
25	Y		Y				4			6			6(HOD)
26	Y		Y	5	4	5	6	4	4	5	6	6	
27	Y		Y	6		2	2				6	6	
28	Y		Y	6	3		3			6	5	2	
29	Y		Y	6	6	5			6	5	5	7	
30	Y		Y	5	6	7	7		7	7	5	4	7(INIT)
31	Y		Y	3	2	2			6		3	2	
32	Y	Y		5	6	4	4	4	6	2	2	4	
33	Y			4	7	6			5	7	6		
34	Y		Y	7	6		6			6		5	
35	Y	B	B	5	6	4	4		5	7	5	6	
36	Y		Y	7	4	6	6	5	4	4	5	4	
37	Y		Y	6	6	6			6	5	6	4	
38	Y		Y	4	2	3	3	1	2	3	4	5	
39	Y		Y	5		7		6	7		6		
40	Y	B	B	3	6	4	5	6		7	6	4	
41	Y		Y	4	3	5	6	2	4	6	5	5	
42	Y		Y		4	6		7		6	6	5	
43	Y		Y	7		5	5					7	
44	Y		Y	5		6	5			6	6	2	
45				7	3	4	4	5	2	2	5	5	
46				6	4	6	5	5	5	6	6	5	
47				3		6		4	5		5	4	
48				6		6	6		4	4	4	6	
49			Y	6	4	6	3	5	4	5	4	5	
50				5	5							6	
51				6		7		5		5	5	5	
52				5		3					5	5	
53	Y			7				6				6	
54	Y			6	5	6					4	5	
55	Y		Y	4		2	2		6		5	4	7(EXP)
56			Y	6	1	4	5	3	1	2	3	5	
57	Y		Y	4	7		5	6	7	7	7	5	
58			Y	5	5	5		1	7	6	3	4	
59				7	5	7						7	
60				6	5	6		6	7	6	6	5	
61				6	2	6	6	1	2	4	6	7	
62				6	1	1	1	3			6	6	
63	Y	B	B	5	1	5	6	4	5	5	6	3	
64				4	4	4					6	6	
65				7	6	5	7		6	6	6	4	
66	Y	Y		7	3	5	5					4	
67	Y		Y	3		5	6			5	7	6	
68	Y		Y	7	4	7	7	7	5	5	5	7	
69	Y		Y	6	3	5		1	1		4	5	
70				6	6		7	6			7	6	
71			Y	7	3	5	7	3	6	6	6	6	

	M	N	O	P	Q	R	S	T	U	V	W	X	Y
72	Y			5		6	6		7	7	7		
73	Y			7	7	7	7	6	7	7	6	4	
74			Y	4		5	7		6		7	5	
75				5	5	6		6		6	6	4	
76	Y		Y	4	2	6	5	3	3	5	5	6	
77			Y	7	6	3	1	1	1	5	4	7	
78			Y	7	6	7						6	
79				6						3	3	3	
80			Y	4	5	4	3		5	5	5	3	
81				7	7	7			7			7	
82				6	6	3	3	1	6	4	4	6	
83	Y			4	6	6	5	4	7	6	6	6	
84				5	6	5	5	5				4	
85				5	5	3	3	5	5	5	6	6	
86				7	5	7	7	7	6	4	4	6	
87	Y			5		6	6				5	5	
88	Y		Y	4		6			5	4	3		
89				5	4	6					4		
90				5	2	7	7		— 5	5	5	2	
91	Y		Y	7	4	7	7	1				4	
92				6	3	6						6	
93	Y		Y	5	2	5		6			7	7	
94			Y	7	7	5	3	5	5	5	6	6	
95				6	3	6						4	
96			Y	6	6	6	5	4			5	5	
97				7		6	1				5	6	
98		B	B	6		6			5	5	5	6	
99				5	5	6	6	4	6	4	4	3	
100	Y			6	7	6				6	6	4	
101				6		5							
102				6	1	5	3	1	2	2	6	7	
103				5	1	6	6	5			6	7	
104	Y	B	B	6	3	5	5	5	4	4	6	5	
105	Y	B	B	5	4	6	6	5	2	5	6	1	
106	Y	Y		6	6	5	5	6	4	5	6	5	LIB 3
107	Y			6	6	5	5	4			5	6	
108	Y		Y	5		5	5			4	4	2	
109	Y		Y	7	5	6	5	2			4	3	
110	Y			6	4	6	6	4		6	7	6	
111	Y		Y	6	6	5	6	5	5	5	5	7	
112	Y		Y	1	4	4		2					
113	Y			6		5	5	6			6	5	
114	Y			3		5	5					6	
115	Y		Y	3		4	5					4	
116			Y	5	3		6	5		6	6	5	
117				6	4			5			4		
118	Y									6			
119	Y		Y	4	5	4	5	4	3	4	4	5	
120				4	2	2		2				4	
121	Y	B	B		5	7	7	6	5	5	5	7	
122	Y			5	3	5					7	5	
123	Y		Y	7	4	7	7	1	1	1	1	5	
124			Y	3		4	4				3	4	
125				4	1	3	5	1	3	1	1	6	
126			Y	6	6	4	5	6	6	4	6	7	
127	Y		Y	4		4	5	6	5			6	
128			Y	7			5						
129	Y		Y	5	7			5			3	5	
130	Y	Y		7	5	6	4	4	4	3	5	5	
131	Y	Y											
132	Y			3	4	4	5	5	4	3	2	7	
133	Y		Y	5		6		6			6	7	
134	Y	Y											
135	Y		Y	5	5	4	4	5	5	5	6	7	
136	Y		Y	7	7						7		
137	Y	Y		7	7	7	6				6	4	
138	Y		Y	6	1	6	3	1	1	1	1	5	
139	Y			5	6	5	4	5	6	4	4	5	
140				6	3	6	3			6	6	1	
141	Y		Y	7		7		5			7	6	

	M	N	O	P	Q	R	S	T	U	V	W	X	Y
142	Y		Y	5		6				5	4	5	
143			Y	7	6	4	5	6	7	6	6	4	
144	Y		Y	5	3	3	3		7	7		6	
145	Y	B	B	5		5	5	4				4	
146	Y		Y	5	6	6	6	1	2	2	6	6	TECH -6
147			Y	5	7		6	5	1	6	5	6	
148				7	7	2	2	2	1	2	3	6	
149				4	4	4	4	4	4	4	7	7	
150		B	B	6	2	6	6	3	3	5	6	6	
151	Y		Y	5		7	7			5	6	7	
152		B	B	6		6		1		5	5		
153				4		6			6	6	5	6	
154			Y	6		6	7		7		5	5	
155				5	5	6	3	3	2	5	6	6	
156				1	1	1	1	1	1	4	4	3	
157			Y	6	6	4	4	3	2	2	2	2	
158	Y		Y	5	6	7	7		6		7	7	
159	Y			2	4		5		5	5	5	6	OVERSO -
160				5	6	6	6		6		6	5	STUDT- 6
161				6	1	7	7	1	7	7	7	7	
162				6	6	7	7	4	6	6	6	4	
163				6	3	6			7		6	3	
164				6		6			6	6	6	6	
165	Y		Y	6	1	6	2	4	6	5	3	7	
166			Y	6	2	3	4	1	1	1	4	4	
167			Y	7	6	7						5	
168	Y			5	3	6	6				6	5	
169	Y		Y	2	5	2	2	4	4	4	3	3	
170	Y		Y	3	5	6					5	5	
171				6	2		6		6		4	6	
172				5		6			6	7	7	5	
173				5	4	4	4	4	6	4	6	4	
174				6	5	4	5	5	7	6	6	7	
175	Y		Y	6	7	6	4	6	7	1	5	6	
176				6	5	5							
177				6	5	5	5	5	4	5	5	4	
178		B	B	7	4	6	6	5	7	6	6	6	
179	Y			6	5	5	4	3	6	6	6	3	
180				4	6	6	6			4	5	5	
181	Y		Y	1	2	7	3	1	1	1	1	3	
182	Y		Y	3	3	4	4	4	4	3	4		
183	Y		Y	7		7					5		
184	Y			6							5	2	
185	Y	Y		3	4	5	5			5	6	5	
186	Y		Y	6		6						6	
187	Y			5	4		4				4	6	
188	Y		Y	4								4	
189	Y		Y	7	5	6	7	4	6	6	6	7	
190	Y	B	B	5		3	5					7	
191	Y		Y	5	2	6	6	4	2	4	5	6	
192	Y		Y	6		4	6	7			6	6	
193	Y		Y	2	1	2	2		1	1	5	7	TUTORS-7
194	Y			4		5				6			
195	Y	Y		5	1	4	3	1	1	1	1	5	
196	Y	Y		7	1	6		2				2	
197	Y		Y	4	1	6	6	1	1	1	1	5	
198	Y		Y	5		5						6	
199	Y		Y	7		6							
200	Y		Y	5	7	6	6	6			4	6	
201	Y		Y			7							
202	Y		Y	6	2	7	1	2	1	3	5	5	
203	Y		Y	4	5	3	3	4	1	2	5	4	
204	Y			6	5	6	5		6	6	6	6	
205	Y			6		6			7	7	7	6	
206	Y	Y		3		4	5		4	4	5	6	
207	Y	Y		4	4						4	5	
208	Y	B	B	6	4	5	5	4	6	5	5	6	
209	Y		Y	6	6	6	6				6	6	
210	Y		Y	5	4	5	4	2	2	2	2	3	
211	Y			5	2	6	6	4			5	6	

	M	N	O	P	Q	R	S	T	U	V	W	X	Y
212	Y		Y	3	1	3	3	3	3	5	5	6	
213	Y		Y	4	1	5	6	1	3	3	3	4	
214	Y	Y		5	5	6	6						
215	Y		Y	5		5	5						
216	Y		Y	6	4		6		4		5		
217	Y	B	B	6		7							
218	Y		Y	7	7	7	7	7	5	6	7	5	
219	Y		Y	3	4	5						6	
220	Y		Y	4	4	6	6	3			3	7	
221			Y	4	5	6	6	6	3	3	7	5	TUT-6
222	Y		Y	7	7	7	7	7	7	6	4	6	
223	Y		Y	7	5	7	7	5	6	6	6	1	
224			Y	6	2	7		6					
225	Y		Y	5	7	5	5	4	1	3	6	7	
226			Y	7	2				5	5	5	7	
227	Y		Y	5	2	6	5	7	4	5	6	6	
228			Y	5	4	4						6	
229	Y		Y	6	3	5	3	4	1	1	3	4	
230	Y		Y	7		7	7	7					
231			Y	3	3	6	6	2	6	6	6	6	PASTS-7
232			Y	5	6	6	4					7	
233	Y		Y	3		3	3	6				2	
234		Y		4	6	5	5	4	6	6	4	7	TUT -2
235			Y	6		6			6	6	6		
236				7	7	7	7	6	6	5	5	7	
237			Y	4	5	4	4	4	4	4	4	5	
238				4	6	6						3	
239	Y		Y	4	4	7	6	7	3	3	5	7	
240		Y		5	1	4	6	2			5	3	
241		Y		6							3	4	
242			Y	5	6	7	5	3	7			5	
243				5	5	2	2	5				5	TUT -5
244			Y	6	5			4		6	7	7	
245				5	6	6	6	1	4	1	5	1	
246			Y	5	2	5						4	
247			Y	4		4						5	
248			Y	6	4	5	5	3	5	6	5	6	
249	Y			4	2	6	4	6	2	2	2	4	
250	Y		Y	7	3	7	7	5	5	5	5	6	
251	Y		Y	6	5						7	5	
252				3	2	4	3	2	2	3	3	6	
253			Y	1	1	1	1	1	1	1	1	1	
254			Y	6	6	7			4	6	5	7	
255				5	5								
256			Y	5	3	4						6	
257		B	B	5	6	3	2	6			4	7	
258			Y	5				5		5	5	6	
259				4	5	6	6	7	7	6	4	4	
260			Y	4	4	5	7	7	7	7	7	7	DEAN'S A
261			Y	4	2	5		7					
262			Y	5	7					5			
263	Y		Y	5		6					4	6	
264			Y	5			5	6					
265	Y		Y	4	1	7	6	3	5	3	5	6	
266			Y	3	6	5						4	
267			Y	7	6	7	7	7	7		7	7	
268			Y	5	2	5		6	7			7	
269				6	5	7		3			6	6	
270		B	B	6	6	6	6				5	5	
271		B	B	5	3	5	5	6	6		6	5	
272	Y	B	B	7	7	7	7	7	7	7	7	5	
273		Y		7	2	7	7	5				4	
274		Y		6	5	6	6	5			5		
275			Y	7		6			6	6	6	6	
276				7		3			6		6	6	
277				6	2	6		5			5	5	
278		Y											
279				7	6	6	4	7	5	6	5	7	
280	Y		Y	6	5	7	7	3	7	6	6	7	
281			Y	3		6	5			4	5	6	

	M	N	O	P	Q	R	S	T	U	V	W	X	Y
282				4		6				6	5		
283				5		6		6				5	
284	Y			2	1	3		2					
285				6	3	4	4	6	5	6	6	6	
286			Y	4	4	4	4	4	4	4	4	5	
287				6	6								
288				4	2	5	5	1	6	2	2	5	
289			Y	7	5	6	3	4			6	5	
290			Y	2		4		6	5			6	
291				7	4	4	4	4	5	5	3	5	
292				7	2		6	4				4	
293				7		6	7				7	5	
294		Y		5		5	5				3	7	
295			Y	6	4			7				5	
296			Y	5	4	6	3					6	
297				6	2	6	4					6	
298	Y		Y	5	2	2					4	7	
299	Y			5	4	6	2	3			5	5	
300			Y	4	6	6	4	5	1	4	2	3	
301				5	3	5	3	3	3	6	6	7	MENTOR -
302	Y		Y	5	1	2	6	3				5	
303				7	1	7	7	4	2	6	6	6	MENTOR -
304			Y	6	4	7	6			6	6	6	
305				6	6	6			5		6	6	DEAN'S AS
306	Y		Y	4	4	6	5	4	4	5	6	6	
307				7	5	7		7			5	7	
308				4	4	4	4	4	4	4	4	7	
309				5	1	5						6	
310			Y	6	2	7	6	6	1	1	1	6	
311	Y		Y	6	4	6	5	1	3	4	4	7	
312				4		5	5				5		
313	Y			7	7			7					
314	Y	Y		5	1	1	1	1	1	5	5	7	
315	Y			5	4	4	4	5	4	4	5	5	
316			Y	3	1	4	3	2	3	3	3	5	
317	Y		Y	5	3	7	7	7	7			6	SCH CAR -
318	Y		Y	2	1	5	4	7	1	1	5	7	
319	Y	B	B	5	4	6	5	6	4	4	4	6	
320	Y	Y		5	4	3	2	2	2	2	2	5	
321			Y	7	7	6	6	7	5	5	5	5	
322				5				6				4	
323	Y	B	B	5	3	5	5	6				3	
324			Y	6	6	6					6	5	
325				3	4	6	4	5	2	4	5	4	
326				5	5	5						5	
327	Y			5	1	6	6	6	2	2	2	5	
328				5	3	5	4	6	5	5	5	6	
329	Y			5		5	5	3			4	6	
330				3	3	1	4	6	3	5	3	3	
331					4							7	
332				7		7	7					5	
333				5	1	6	1	1	1	1	1	6	
334	Y		Y	6	6	6	6			6	6	6	
335	Y		Y	7	1	7	7	5	4	1	1	7	
336				6		6					7	7	
337				5		5					5	6	
338			Y	7	6	7	7	7			6		
339	Y		Y	3		5							
340	Y			3	7			5	1		7	4	
341				5	5	5	5	3	6	4	4	6	
342	Y			6	5	5	4	2	1	1	1	2	
343			Y	4	4	4					4	7	
344	Y		Y	5	1	6		7	4	4	4	2	PAST STU
345	Y		Y	6		4		5			6		
346	Y	Y		7		6	6	2					
347	Y		Y	2	3	3	2						
348	Y	B	B	6	6	6	7		2			6	
349	Y		Y	5	5	6	7			5		7	
350	Y		Y	6	5	6	6	4	5	6	6	5	
351	Y	B	B	7	7	7	7		7	7	6	4	

	M	N	O	P	Q	R	S	T	U	V	W	X	Y
352	Y		Y	2	1	1	1	4	1	2	2	7	
353	Y		Y	4					5			5	
354	Y		Y	6		6						3	MENTOR -
355	Y		Y	4		4	4	5			4	5	
356	Y	Y		5		4						5	
357	Y	Y		5		6						2	
358	Y	Y	Y	7		6	5	4			4	7	
359	Y		Y	4		6	6	6					
360	Y		Y	5		7		7			6	6	
361			Y	4	6	4	4	4			4	6	
362	Y		Y	6	6	6	6	7	6	7	7	7	
363	Y		Y	6				6				4	
364	Y			6	6	6	5	6	7	6	6	5	
365	Y			5	4	5	5	6	5	4	4	5	
366	Y		Y	6		7		5		5		3	POLY -5
367	Y		Y	4	4	5					4	5	
368	Y		Y	7		6							BROS-5
369	Y		Y	3	3	4	4	5	3	3	3	4	
370	Y		Y	4	4	5	5						
371	Y			6	3	6		6	6				
372	Y		Y	6		6							
373	Y			7	7	7	7		7	7	7	7	
374	Y			5	5	5	5	5	4	6	6	5	
375	Y		Y	4	4	4	4	4	3	6	6	7	
376			Y	3	4	4	4	4	3	3	3	4	
377	Y		Y	3	6	4	3	1	4	4	4	1	
378	Y						5					4	
379	Y	Y		5								7	
380	Y		Y	6		5						7	
381	Y		Y	7		7							
382	Y		Y	5	3	6	5	3	5	4	3	6	
383	Y		Y	3	3	3							
384	Y		Y	7	4	5	5			5	6	5	
385	Y	Y		7		6			7	4	4	1	
386	Y			6	4	4	4	6	1	3	3	5	
387	Y			5		5	5	6	4	4	4	5	
388	Y			6	2	4	5	6	5	3	2	5	
389	Y	B	B	6	5	7	7	5	7	6	7	6	FAM - 6
390	Y			4	4	4	4	4	4	4	4	7	
391			Y	6									
392	Y		Y	6	7	7	4	5	1	5	7	7	
393	Y		Y	7	1	5					4	6	
394	Y			5	3	5						6	
395	Y		Y	2	6	6		6	4	4	4	6	
396	Y		Y	3		4				2	2	5	
397	Y		Y	5	2	4	4	5	1	1	1	1	
398	Y		Y	2			3	4				5	
399				4	4	4			5				
400	Y			3	5	5							
401	Y	Y		6	5	5	5		6			7	
402	Y			5	5	6						6	
403	Y			6	3	6	4	4	4	5	6	7	
404			Y									7	
405	Y		Y	7	5	5	5	2	2	2	3	5	
406		Y		5	4	6							
407	Y		Y	3	3	6	6	1					
408	Y		Y	7	2	1	1	1	3	6	3	7	
409	Y		Y	5	2	3		2	2			3	
410	Y	Y		5	7	4	5	7	6			6	
411	Y		Y	3	3	6	4	3	6	6	6	7	
412	Y	Y		5	4	6	6	5	7	5		5	
413	Y			5	4	4	4		3	3	3	5	
414	Y		Y	5	5	4	6	5	4	5	6	5	
415	Y		Y	5	4	5	4	2	1	2	2	5	
416	Y		Y	6	6	5						6	
417	Y		Y	4	4	5	3	4	4	4	5	6	
418				4	6	7	7					6	
419				2	6	2	3	3	4	5	5	4	TUT - 6
420	Y		Y	5	2	4							OWN RES
421	Y			5		6		7				5	
422	Y		Y	3			3						

	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
1	17	18	19	20	21	23	24	25	26	27	28	29	30
2	Y	Y	Y	N	N	P	5	5	6	2	3	5	5
3	Y	Y	N	N	Y	P	5	7	5	5	7	6	7
4	Y	N	N	Y	N	Y	5	7	7	7	7	5	6
5	N	N	N	N	N	P	7	7	7	7	7	6	6
6	N	N	N	N	N	Y	7	7	7	7	7	6	6
7	Y	Y	N	N	Y	Y	5	6	6	7	5	6	7
8	Y	N	N	N		Y	5	7	6	7	7	4	7
9	Y	Y	Y	Y	Y	Y	6	5	4	4	6	6	2
10	N	N	N	Y	N	P	5	7	6	7	4	4	7
11	N	N	N	N	N	P	4	7	1	4	7	7	1
12	N	N	N	N	N	P	4	5	5	5	4	5	5
13	N	N	N	Y	N	Y	5	6	6	5	6	4	5
14	N	N	N	Y	N	Y	3	7	6	6	7	5	4
15	Y	N	N	N	N	P	5	7	7	5	6	6	4
16	N	N	N	N	N	Y	4	7	6	6	7	4	5
17	N	N	N	N	N	P	5	7	5	6	5	6	6
18	N	Y	N	N		P	7	7	7	7	7	7	7
19	Y	N	N	N	Y	P	2	7	6	5	1	4	2
20	Y	N	N	N	N	Y	7	7		5			5
21	N	N	N	Y	N	Y	7	7	7	7	5	5	4
22	N	Y	Y	Y	N	P	2	7	5	6	5	3	3
23	N	N	N	N	N	Y	4	7	7	6	7	7	7
24	N	Y	N	N	N	P	4	6	6	6	6	5	4
25	N	Y	N	N	N	N	7	2	4	4	4	4	1
26	Y	Y	Y	Y	Y	Y	5	7	6	6	7	6	4
27	Y	Y	N	N	N	Y	3	6	5	4	6	6	6
28	N	N	N	Y	N	Y	4	5	5	7	7	3	1
29	Y	Y	Y	N	Y	Y	7	7	7	7	7	7	7
30	Y	N	N	N	N	P	7	7	7	7	7	7	7
31	Y	Y	Y	N	N	Y	3	7	6	5	3	5	7
32	Y	Y	Y	N		Y	5	7	7	7	5	7	7
33	N	N	Y	Y	N	P	4	6	7	6	1	3	3
34	N	N	N	N	N	Y	5	6	5	6	2	3	3
35	Y	Y	N	N	N	Y	4	7	6	6	5	5	5
36	Y	Y	Y	Y	N	P	4	5	6	7	7	6	6
37	N	N	N	N	N	Y	6	6	6	6	7	6	4
38	N	Y	Y	Y	Y	Y	7	7	7	7	7	7	7
39	Y	Y		Y		Y	7	7	3	3	7	7	4
40	Y	N	N	N	N	Y	6	7	7	7	6	5	7
41	Y	Y	N	Y	Y	Y	5	7	6	6	4	5	5
42	Y	Y	Y	N	N	P	4	7	6	6	6	5	7
43	Y	Y	N	N	N	P	3	7	7	7	7	7	7
44	N	Y	N	N	N	Y	3	5	5	3	2	2	2
45	N	N	N	N	N	P	4	6	6	6	6	4	6
46	Y	N	N	N	N	Y	4	7	6	6	6	6	6
47	Y	N	Y	N	N	P	6	7	7	6	7	4	6
48	N	N	N	N	N	Y	7	7	5	6	7	7	7
49	Y	Y	Y	Y	N	Y	5	7	7	6	6	5	6
50	N	N	N	N	N	P	6	7	7	7	7	6	7
51	N	N	N	N	N	P	1	7	7	5	6	5	6
52	Y	N	N	Y	N	Y	6	7	7	6	5	7	6
53	N	N	N	N	N	Y	7	7	5	5	7	7	7
54	N	N	N	N	N								
55	N	N	Y	N	N	P	6	7	5	6	4	3	5
56	N	Y	N	Y	Y	P	5	7	7	6	6	6	7
57	Y	N	N	Y	N	P	5	6	4	4	7	7	4
58	N	Y	Y			Y	5	5	5	5	5	5	5
59	N	N	N	N	N	Y	4	7	4	4	7	7	7
60	Y	N	Y	Y	N	Y	5	6	6	7	5	4	7
61	Y	Y	Y	Y	N	Y	7	7	7	7	7	7	7
62	N	N	N	Y	N	P	6	7	7	7	6	3	7
63	Y	Y	Y	Y		Y	6	7	7	7	7	6	7
64	Y			Y									
65	Y	Y	Y	Y		P	7	7	7	7	7	7	7
66	Y	N	Y	N	N	P	1	6	5	4	5	4	2
67	N	N		N	N	P	3	6	5	5	6	4	6
68	N	N	Y	N	N	Y	7	7	7	7	7	7	7
69	Y	N	Y	N	Y	Y	3	7	5	5	6	5	3
70	N	N	Y	Y	Y	P	7	7	7	7	7	7	7
71	N	N	N	N	N	P	6	7	7	7	7	7	7

	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
72	N	N	N	N	N	P	1	5	5	5	7	3	5
73	N	N	Y	Y	Y	Y	7	7	7	7	7	7	7
74	Y	N	N	N	N	Y	3	6	5	4	6	5	7
75	Y	N	N	Y	N	P	7	3	3	6	7	3	3
76	Y	Y	Y	Y	Y	Y	6	6	6	5	3	6	5
77	Y	N	N	Y	N	Y	7	7	7	7	7	7	7
78	Y	N	N	Y	N	Y	6	7	7	7	7	7	7
79	Y	N	Y	N	N	P	5	7	7	7	5	6	7
80	N	N	Y	N	N	P	4	6	6	7	6	5	6
81	N	Y	N	Y	N	P	7	7	7	7	7	7	7
82	N	N	Y	N	N	Y	3	6	6	4	5	5	6
83	Y	N	Y	Y	N	Y	6	5	6	1	7	5	1
84	N	N	N	N	N	P	6	6	6	6	7		7
85	N	N	N	N	N	Y	7	7	5	6	7	6	6
86	Y	Y		Y		Y	5	7	7	7	6	6	7
87	N	Y	N	N	N	P	7	7	6	7	7	7	7
88	N	N	N	N	N								
89	N	Y	Y	N	N	P	3	7	7	7	7	6	7
90	N	Y	Y	Y	N	Y	5	7	5	5	7	4	7
91	Y	N	N	N	N	Y	7	7	7	7	7	7	7
92	N	N	N	N	N	P	3	7	5	6	7	5	7
93	N	N	Y	N		Y	5	6	6	5	6	6	6
94	N	N		N	N	Y	5	7	7	5	7	7	6
95	N	N	N	N	N	P	3	6	5	5	5	4	5
96	Y	N	N	Y	N	P	4	6	6	5	5	4	6
97	Y	N	N	N	N	P	4	7	4	6	7	6	6
98	N	N	N	N	N	Y	4	7	6	6	5	4	4
99	N	N	N	Y	N	Y	6	7	7	7	7	7	7
100		N	N	N	N	Y	5	7	4	5	7	6	5
101	N					Y		6			6		
102	Y	N	N	N	N	Y	1	6	4	4	6	5	5
103	Y	N	N	Y	Y	P	5	7	6	5	5	6	7
104	N	N		N		Y	5	7	5	6	6	6	5
105	N	N	N	N		Y	7	7	7	7	7	7	7
106	Y	Y	Y	N	N	Y	6	7	7	7	7	7	7
107	N	N	N	N	N	P	5	6	6	6	6	6	6
108	Y	Y	N	Y	N	P	6	6	6	3	5	3	2
109	Y	Y	Y	Y		Y	6	7	6	7	7	7	7
110	Y	Y	Y	Y	Y	Y	4	6	7	6	7	7	6
111	N	N	N	N	N	P	3	6	6	6	6	6	6
112	N	N	N	N	N	P	2	7	5	5	3	4	2
113	Y	N	N	N	N	P	5	4	3	4	6	6	5
114	N	N	N	Y	N	P	4	7	6	7	3	7	4
115	Y	N	Y	Y	Y	Y	6	7	7	7	7	4	5
116	N	N	N	N	N	Y	7	7	7	5	7	5	5
117	Y	Y	Y	Y	Y	Y	6	6	7	4	7	7	7
118	N		N	N	N	N							
119	N	N	N	Y	N	Y	5	6	5	5	5	4	5
120	Y	N	Y	N	N	Y	6	6	6	5	5		3
121	N	N	N	Y	Y	Y	7	6	7	7	5	6	5
122	Y	Y	N	N	Y	Y	4	7	6	7	6	7	7
123	Y		N	N		Y	7	7	6	6	7	7	7
124	N	N	Y	Y	N	Y	4	5	5	6	6	5	5
125	Y	Y	Y	N		Y	1	1	4	5	5	4	1
126	N	Y	Y	Y	N	Y	5	7	5	5	4	6	6
127	Y	N	N	N	N	P	6	6	5	4	5	4	6
128	Y	N	N	Y	Y	P	2	7	7	7	3	7	7
129	N	N	N	Y	Y	Y	4	7	7	7	7	3	7
130	N	Y	Y	Y	Y								
131													
132	Y	Y	Y	Y	Y	Y	7	7	7	7	7	7	7
133	Y	N	Y	Y	N	P	5	7	4	4	5	3	6
134													
135	N	N	N	N	N	Y	6	7	7	6	6	6	6
136	N	N	Y	Y	N	P	7	7	7	7	7	7	7
137	Y	N	N	N	N	P	6	7	5	5	6	6	6
138	Y	N	N	N	N	Y	4	7	6	6	7	5	4
139	Y	N	Y	N	N	Y	5	3	4	4	3	5	3
140	N	N	Y	N	N	P	6	6	6		5	6	7
141	N	N	Y	N	N	P	6	7	6	6	5	4	7

	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
142	Y	N	N	Y	N	P	5	7	5	5	6	5	7
143	Y	N	N	N	Y	P	6	7	6	6	5	4	7
144	N	N	N	N	N	N							
145	Y	N	Y	Y	N	Y	4	6	6	7	5	6	7
146	Y	N	N	Y	N	P	7	7	7	5	7	7	7
147	Y	N	N	Y	N	Y	6	7	4	4	6	5	7
148	Y	N	N	N	N	P	2	4	3	5	3	5	1
149	N	N	Y	N	N	Y	7	7	7	7	7	7	7
150	Y	N	N	Y	N	Y	7	6	5	6	7	4	6
151	N	N	Y	N	N	P	2	6	6	6	7	5	6
152	N	N	N	N	N	P	5	6		6	6	4	6
153	Y	N	Y	Y	N	P	6	6	4	5	6	5	6
154	N	N	N	N	N	Y	7	7	6	5	7	7	5
155	Y	N	Y	Y	N	P	6	6	7	5	5	5	6
156	Y	N	N	N	N	P	1	1	1	1	1	1	1
157	Y	N	N	N	N	P	5	5	3	3	6	3	5
158	N	N	N	N	N	P	4	7	7	6	7	7	7
159	Y	N	N	N	N	P	5	6	6	5	5	5	5
160	N	N	N	N		P	3	7	5	5	6	4	4
161	N	N	Y	N	Y	P	7	7	3	7	7	6	7
162	Y	N	Y	N	N	Y	6	7	6	6	7	5	6
163	N	N	N	N	N	P	4	7	5	6	3	2	7
164	N	N	N	N	Y	Y	7	7	7	7	7	5	7
165	Y	N	Y	N	N	Y	5	6	6	7	7	6	6
166	N	N	N	N	N	P	4	3	4	4	6	5	4
167	Y	N	N	N	N	P	6	7	4	4	7	1	7
168	N	N	Y	N	N	P	5	5	5	5	5	5	5
169		Y	N	N	N	P	4	2	4	3	3	3	2
170	Y	N	Y	Y	N	Y	4	7	7	4	3	3	5
171	Y	N	Y	N	N	Y	3	6	7	4	4	4	5
172	Y	N	N	N	Y	N							
173	N	N	N		N		5	3	5	4	5	5	5
174	N	Y	Y	Y	Y	Y	7	7	7	5	7	7	7
175	Y	Y	N	N	Y	Y	6	7	7	7	7	3	7
176	N	N	N	N	N	Y	7	7	7	6	5	4	4
177	Y	N	Y	N	N	P	4	7	6	6	6	6	6
178	N	Y	Y		N	Y	7	7	7	7	7	7	7
179	N	N	Y	N	N	P	5	7	5	4	5	6	7
180	N	N	N	N	N								
181	N	N	N	N	N	P	7	3	1	1	5	1	6
182	N	N	N	Y	N								
183	Y	N	N	N	N	Y	4	7	6	4	6	4	7
184						P	6	6	6	6	6	6	6
185	Y			Y		Y	6	7	7	7	6	7	7
186	N	N	N			Y			3	3	5	3	7
187	Y			Y		Y	4	6	5	4	5	2	1
188	N	N	N	N	N	P	3	6	5	5	6	4	6
189	N	N	N	Y	N	P	6	2	4	5	6	6	7
190	N	N	N	N	N	P	4	7	5	5	7	7	7
191	Y	Y	Y	Y	N	Y	6	6	5	6	4	5	6
192	N	N	N	N	N	P	5	6	3	5	6	7	7
193	N	N	Y	N	N	Y	4	3	7	7	2	6	7
194	N	Y	Y	N	N	Y	7	7	6	6	7	7	7
195	N	N	N	N	N	P	3	6	2	2	7	5	2
196	N	N	N	N	N	Y	4	6	4	4	5	6	7
197	N	N	N	N	N	P	7	7	7	5	5	6	4
198	Y	N	Y	N	N	P	5	6	3	2	6	4	5
199						P	5	7		6	6	7	7
200	Y					Y	6	6	6	6	7	6	6
201	N	N	N	N	N	Y	4	5	5	5	5	5	4
202	N	N	N	N	N	P	6	7	5	7	5	6	7
203	N	N	N	N	N	N							
204		Y	Y			P							
205	Y	N	N	N	N	P	4	6	5	7	6	7	7
206	N	N		N	N	P	6	7	5	6	7	5	6
207	N	N	N	N	N	N							
208	N	N	N	N	N	P	4	5	6	5	6	6	6
209	N	N	N	N	N	Y	6	7	7	7	7	5	7
210	Y	Y	N	Y	N	Y	3	6	6	6	4	4	3
211	Y	N	Y	Y	N	Y	5	7	6	6	4	6	7

	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
212	Y	N	Y	Y	N	Y	6	7	7	6	7	5	7
213	N	N	N	N	N	N							
214	N	N	N	N	N	P	1	7	4	6	4	6	7
215	N	N	N	N	N	Y	5	6	5	3	5	4	2
216	N	N	N	N	N	P	5	6	5	5	5	5	4
217	N	N	N	N	N	P	7	7	3	3	5	1	7
218	N	N	Y	N	N	Y	5	5	5	5	5	5	5
219	N	Y	Y	Y	N	Y	6	7	7	7	7	7	7
220	N	N	N	N	N	P	7	7	6	6	6	6	7
221	N	N		N	N	Y	4	5	3	6	6	4	5
222	N	N	N	N	N	Y	7	7	7	7	7	7	7
223	N	Y		N	N	Y	7	7	7	7	7	7	7
224	N	Y				P	6	1	7	6	6	4	4
225	Y	N	N	Y	N	P	5	6	5	6	6	4	5
226	N	Y	N	N	N	Y	5	7	7	6	6	7	5
227	Y	Y	N	N	N	P	5	5	5	6	6	5	7
228	N	N	N	Y	N	Y	6	7	6	6	6	5	6
229	N	N	N	N	N								
230	N	Y	Y	N	—	Y	6	7	7	7	7	7	6
231	N	N	N	N	N	Y	7	7	7	7	7	7	5
232	N	N	N	N	N	P	4	7	6	6	7	6	4
233	N	Y	Y	Y		P	4	5	5	5	7	6	7
234	N	Y	N	N	N	P	5	6	5	5	7	7	7
235	N	N	N	N	N	P	6	6	6	6	7	7	7
236	N	N	N	N	N	Y	5	7	7	4	7	7	7
237	N	Y	Y	N	N	Y	5	5	5	5	6	5	5
238	Y	N	N	Y	N	Y	3	7	6	6	4		3
239	Y	Y	Y	N	N	Y	5	7	7	4	3	6	5
240	N	N	N	N	N	P	4	6	5	6	4	4	5
241	N	N	N	N	N	Y	7	7	7	7	7	7	7
242	N	N	N	N	N	P	6	7	7	6	7	7	7
243	N	N	N	N	N	P	6	7	6	6	6	5	5
244	N	N	N	N	N	P	6	5	2	4	6	6	1
245	N			Y	N	P	6	6	3	5	7	2	5
246	Y	N	Y	N	N	P	4	6	5	5	4	5	6
247	N	N	N	N	N	N							
248	Y	N	N	N	N	Y	3	7	6	6	4	5	6
249	N	N	N	Y	N	Y	5	7	6	7	7	6	7
250	Y	N	N	Y	N	P	5	7	6	5	7	7	7
251	Y	N	N	N	N	P	6	7	4	3	5	7	7
252	Y	N	N	Y	N	P	2	6	6	6	3	1	1
253	N	N	N	N	N	P	7	7	7	7	7	7	7
254	Y	N	N	N	N	P	5	7	7	5	5	5	6
255	N	Y	N	N	N	Y	6	6	6	7	7	6	7
256	Y	N	N	N	N	Y	3	6	4	4	5	5	6
257	Y	N	Y	N	N	P	7	7	7	7	7	7	7
258	N	N	Y	Y	Y	Y	6	6	5	4	5	6	7
259	Y	N	N	N	N	P	5	4	5	4	5	6	6
260	N	N	N	N	N	P	7	7	7	7	6	5	6
261	N	N	N	N	N	P	5	7	4	4	6	2	7
262	N	N	N	N	N	Y	5	7	6	6	5	6	6
263	N	N	N	Y	N	Y	7	5	5	6	4	4	5
264	N	N	N	N		P	4	6	6	5	4	6	4
265	N	N	N	Y	N	Y	4	7	7	6	6	5	7
266	N	N	N	Y	N	P	7	7	6	5	7	6	7
267	N	N	N	N	N	P	7	7	6	7	5	7	7
268	Y	N				Y	7	7	7	7	7	6	7
269	N	N	N	N	N	Y	5	7	6	6	6	6	7
270	N	N	N	N	N	Y	5	5	5	5	5	5	5
271	N	N	N	N	N	Y	7	7	7	7	7	7	7
272	N	N	N	N	N	Y	6	7	6	6	7	6	7
273	Y	N	N	Y	N	P	7	7	7	7	7	7	7
274	N	N		N	N	Y	7	7	7	7	7	7	7
275	N	Y	Y	N	N	Y	4	7	7	7	7	7	7
276	N	Y	Y	N	N	Y	7	7	7	5	7	7	7
277	N	N	N	N	N	P	5	6	5	5	6	5	5
278	N	Y	N	Y	Y	N	4	3	4	4	4	5	5
279	N	N	N	N	N	P	4	6	6	6	5	4	5
280	N	N	N	N	Y	Y	5	6	7	5	5	3	6
281	N	N	N	N	N	N	5				3		

	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
282	Y	N	Y	N	N	Y	3	6	7	7	6	4	5
283	N	N	N	N	N	Y	4	6	5	6	6	5	7
284	N	N	N	N	N	P	3	4	1	3	3	2	4
285	N	Y	Y	Y	N	Y	7	6	7	7	7	6	7
286	Y	Y	N	N	N	Y	5	5	5	4	5	5	5
287	N	N	N	N		Y	6	6	5	5	4	4	4
288	N	N	N	N	N	Y	7	7	7	7	7	7	7
289	N	N	N	Y	Y	Y	5	7	6	6	4	5	7
290	N	N	N	N	N	Y	7	6	6	5	5	3	6
291	N	N	N	N	N	N							
292	Y	N	N	N	N	P	5	6	6	6	7	5	6
293	N	N	N	N	N	P	5	7	4	4	7	2	7
294	N	Y	N	Y	N	Y	6	7	7	7	5	7	7
295	N	N	N	N	N	P	6	5	3	4	2	5	6
296	N	N	N	N	N	Y	7	6	5	6	6	7	7
297	N	N	N	N	N	P	4	4	2	4	7	6	5
298	N	Y	Y	Y	N	Y	5	5	6	7	7	5	5
299	N	Y	N	Y	N	Y	6	4	6	5	4	6	3
300	N	N	N	Y		Y	7	7	6	7	5	6	7
301	N	N	N	N	N	Y	6	6	7	7	7	7	6
302	N	N	N	N	N	P	3	6	7	7	5		7
303	N	N	N	N		N							
304	N	N	N	N	N	P	6	7	6	5	6	6	7
305	N	N	N	N	N	Y	5	7	7	7	6	6	6
306	N	N	N	Y	N	Y	6	7	5	5	5	6	6
307	N					N							
308	N	N	N	N	Y	N	1	6	4	1	7	1	4
309	N	N	N	N	N	P	4	4	4	4	1		3
310	N	N	N	N	Y	Y	7	7	7	7	7	7	7
311	Y	Y	N	Y	Y	P	5	6	5	5	5	6	7
312	N	N		N	N	Y	3	7	5	5	7	7	5
313	N	N	N	N		Y	7	7	7	7	7	7	7
314	Y	N	N	N	N	Y	5	3	1	1	1	3	1
315	Y	Y	N	N	N	P	4	5	5	5	5	5	6
316	N	N	N	N	N	P	5	5	6	5	7	7	7
317	N	N	N	Y	N	Y	7	6	7	4	7	6	5
318	Y	N	N	N	N	P	5	7	7	1	4	1	5
319	N	N	N	N	N	Y	6	7	6	5	6	5	6
320	N	Y	Y	Y	Y	P	4	7	4	4	7	7	4
321	N	N	N	Y	N	Y	7	7	7	7	7	7	7
322	N	N	N	N	N	P	6	7	7	6	6	7	7
323	N	Y	N	N	N	Y	6	7	3	6	6	6	5
324	N	N	N	N	N	P	5	6	5	5	5	6	3
325	N	N	N	N	N	P	5	7	4	4	4	5	5
326	Y	N	N	N	N	Y	5	6	6	5	4	4	4
327	N	N	N	N	N	P	4	4	5	6	7	4	4
328	N	N	N	Y	N	N							
329	N	Y	Y	Y	Y	P	4	6	6	7	5	6	6
330	Y	Y	Y	Y	N	Y	5	5	6	6	3	6	6
331	N	N	N	N	N	Y	6	7	7	7	7	6	7
332	N	N	N	N	N	N							
333	N	N	N	N	N	P	7	7	7	7	5	7	7
334	N	N	N	N		P	7	7	7	7	7	7	7
335	N	N	N	N	N	P	7	7	7	7	7	5	7
336	N	N	N	N	N	P	6	7	7	5	7	7	6
337	N	N	N	N	N	P	5	7	5	5	6	4	5
338	N	N	Y	N	N	P	7	7	7	7	7	7	7
339	N	N	Y	N	N	Y	6	6	7	7	6		6
340	Y	N	N	N	N	Y	3	7	3	5	5	5	7
341	Y	N	N	N	Y	P	4	6	6	6	6	5	5
342	N	N	N	N	N	P	6	7	6	7	7	5	7
343	N	N	N	N	N	P	4	7	7	7	7	7	7
344	Y	Y	N	Y	N	Y	7	7	7	7	7	7	7
345	N	N	N	N	N	Y	6	7	5	5	6	5	7
346	N	N	N	N	N	P	6	6	5	6	6	6	5
347	N	Y	Y	Y	N	Y	7	7	6	7	7	7	7
348	Y	N	N	N	N	Y	2	5	7	6	5	7	6
349	N	N	Y	Y	N	Y	4	7	5	7	5	7	5
350	Y	N	N	Y	N	Y	6	7	7	5	3	6	7
351	N	N	N	N	N	Y	4	7	7	7	7	6	6

	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
352	N	N	N	N	N	Y	7	7	2	4	6	2	7
353	N	N	N	N	N	Y	5	7	5	6	6	6	6
354	N	N	N	N	N	P	6	5	5	6	6	4	4
355	N	Y	Y	N	N	Y	5	5		7	7	6	6
356	N	N	N	N		P	5	6	3	3	4	4	5
357	N	N	N	N	N	Y	3	5	5	4	3	3	2
358	Y	N	N	N	N								
359	N	N	N	N	N	Y	5	6	5	6	5	7	5
360	Y	Y	Y	Y		Y	7	7	7	7	7	7	7
361	N	Y	Y	Y		P	5	7	6	7	6	6	6
362	Y	N		N	N	Y	7	7	7	7	6	6	7
363	N	N	N	N	N	Y	7	7	7	5	7	7	7
364	Y	Y	Y	Y	Y	Y	7	7	5	7	7	7	7
365	N	N	N	N	N	P	7	7	5	7	6	6	7
366	N	N	Y	N	N	P	3	6	5	6	4	6	7
367	N	Y	N	N		Y	5	6	6	6	5	6	7
368	N	N	N	N	N	Y	5	5	5	7	2	4	2
369	N	N	Y	Y	Y	Y	5	7	6	6	7	7	7
370	Y	Y	Y	Y	N	Y	4	5	6	6	4	6	6
371	N	N	N	N	N	P	4	7	7	7	5	7	7
372	N	N		N		P	7	6	4	4	6		7
373	N	N	Y	Y	Y	Y	6	5	5	5	5	6	6
374	N	N	N	N	N	Y	7	7	7	7	7	7	7
375	N	N	Y	Y	Y	Y	4	4	4	4	4	4	4
376	N	N	N	N	Y	Y	4	5	5	5	4	3	7
377	N	Y	N	N	N	Y	5	7	6	6	5	7	7
378	N	N	N	N	N	P					7		6
379	N	N	Y	Y	Y	Y	7	7	7	7	7	7	7
380	N	N	N	N	N	Y	6	7	6	6	4	4	4
381	N	N	N	N	N	Y	7	7	7	7	7	4	7
382	N	N	N	N	N	Y	5	7	5	6	6	6	6
383	N	N	N	N	N	P	4	5	5	6	4	6	4
384	N	N	N	N	N	Y	5	5	5	5	7	7	7
385	N	N	N	N	N	P	7	7	6	5	3	3	5
386	N	Y	N	N	N	P	6	7	4	5	2	7	7
387	N	N	N	N	N	P	5	7	6	6	5	5	6
388	N	Y	Y	Y	Y	P	6	6	6	6	6	6	6
389	N	N	N	N	N	Y	6	7	7	7	7	5	4
390	N	Y	Y	Y	N	P	6	6	6	6	6	6	6
391	N	N	N			Y	6	4	4	4	4	4	4
392	N	N	Y	Y	Y	Y	6	6	6	6	6	6	6
393	N	N	N	N	Y	P	5	7	4	6	6	5	5
394	N	N	N	N	N	P	5	6	2	2	6	5	6
395	Y	N	Y	N		P	7	7	5	7	6	4	7
396	N	N	N	N	N	P	7	7	7	7	7	7	7
397	N	N	N	N	N	P	5	6	4	4	2	2	3
398	N	N	N	Y	N	P	5	6	6	6	6	6	5
399	N	Y	N	N	N	P	5	7	5	5	5	6	5
400	N	N	N	N	Y	Y	4	3	3	3	4	3	4
401	N	N	N	N	N								
402	N	N	N	Y	Y	Y	4	6	6	7	4	7	7
403	N	N	N	N	N	P	6	6	6	5	7	7	7
404	N	N	N	N	N	P	5	7	3	4	1	2	7
405	N	N	N	Y	N	Y	5	6	2	4	6	6	5
406	N	N	N	N	N	P	4	7	4	5	5	4	6
407	N	N	N	N	N	P	6	7	7	6	7	5	6
408	N	N	N	N	N	Y	7	6	2	2	5	6	7
409	N	N	N	N	Y	P	6	5	4	4	6	5	4
410	N	N	N	N	N	N							
411	N	N	Y	N	N	Y	5	6	6	5	2	5	5
412	N	N	N	N	N	P	5	7	2	4	7	7	7
413	N	N	N	N	N	P	5	6	5	5	5	6	6
414	N	Y	Y	Y	Y	P	4	5	4	4	5	4	5
415	N	N	N	N	Y	Y	4	7	5	4	4	4	6
416	N	N	Y	Y	N	P	6	7	6	7	4	5	7
417	N	Y	N	N	Y	Y	5	6	6	6	5	6	6
418	N	N	N	Y		Y	7	7	7	7	4	7	7
419	N	N	N	N	Y	Y	7	7	7	6	7	7	7
420	N	N	N	N	N	P	7	7	6	4	7	5	4
421	N	N	N	N		P	5	7	7	7	7	7	7
422	N	N	Y	Y	Y	Y	6	7	6	6	6	2	6

Appendix G

Enrolment questionnaire analysis

Majors

Accounting
Agricultural Business
Biochemistry
Biology
Botany
Chemistry
Computer Science
Computer Systems Engineering
Ecology
Economics
Education
English
Enviromental Science
French
Finance
Geography
German
History
Horticulture
Information Engineering
Information Systems
Japanese
Linguistics
Management
Marketing
Mathematics
Maori
Mathematical Physics
Media Studies
Microbiology
Molecular Biology
Operations Research
Philosophy
Physiotherapy
Physics
Plant Science
Psychology

Regional Planning
Religious Studies
Rural Valuation
Social Work
Sociology
Social Anthropologyy
Soil Science
Statistics
Tourist Management
Valuation
Zoology

Classes surveyed

Paper No	Count	%
59.111	79	18.8
59.211	37	8.79
59.327	27	6.41
57.102	57	13.5
57.202	35	8.31
57.302	16	3.80
60.103	69	16.4
20.205	43	10.2
45.223	20	4.75
45.323	6	1.43
60304	13	3.09
75.305	19	4.51

Breakdown by age

Group	Count	%
16	1	0.242
17	2	0.483
18	69	16.7
19	97	23.4
20	73	17.6
21	57	13.8
22	25	6.04
23	14	3.38

23+	1	0.242
24	13	3.14
25	10	2.42
26	6	1.45
27	7	1.69
28	2	0.483
29	7	1.69
30	3	0.725
30+	1	0.242
30s	1	0.242
31	4	0.966
32	2	0.483
33	3	0.725
34	1	0.242
35	5	1.21
36	1	0.242
37	1	0.242
38	1	0.242
39	2	0.483
40	1	0.242
40+	1	0.242
41	1	0.242
45	1	0.242
47	1	0.242

Appendix H

Enrolment questionnaire- responses to open-ended questions

H1 Responses to Question 8

Windows

75305
11. Easy to follow
13. Ease of usage

59HCI
2. Simple, fast, more information easily displayed
3. Because the interface seems to be simpler and clear

59327
32. Don't have to remember command names, easier to use
4. Easy to remember how to use
6. Find it and more interesting easier to use
7. Easier to use - more friendly
8. Windowed if appropriate software to task available
9. User friendly, easy to look at especially Macintosh
10. Reduced memory load due to visual feedback. no "where am I? what was I doing?" problems if I get sidetracked.
11. Visual stimulation
14. Intuitive
17. Consistent interface
18. Easier to rest your hand on a mouse and click the button than to hit the keys
19. Windows more helpful environment and less error prone
20. When you know DOS its OK but windows environment has very little learning. Unix open windows is OK but a bit fiddly. MAc easier. Window based is good for word processing esp good for spread sheets but file management is a lot of pain. I've got a PC which is good for file management.
23. Less memory load - don't waste time figuring out how to do what you want. Can get on with the task at hand. Menus help by prompting - giving ideas
24. Easier to keep track of what you are doing, better HCI
25. Easier to swap applications, control etc

57302
1. Easier to use
9. Easier to handle
10. Multitasking, easy of use
11. Its prettier
14. Feel they are more user friendly

45323
1. Simplified, easier to understand, less knowledge of computers needed
2 . Easier to learn to use
4. Much simpler and faster

<u>60304</u>
4. Easier to use requires smaller knowledge of computer specific commands
The mouse and windows are easier to use, clearer information etc.
6. Clearer and simpler
8. A lot more user friendly
11. Easy to use pull down menus at finger tips
13. Easier to see and follow and understand

<u>45223</u>
6. Window based computers seem more user friendly. In either case, simple basic instruction manuals would be of assistance
11. Easier to read
12. Easier to use
14. More user friendly

<u>57202</u>
6. Faster, user friendly
7. Easier to see and very friendly
10. No need to memorise commands- mouse driven
13. Ease of use GUI
14. Window based like Macs allow me to manipulate files and information faster and commands require less time and activity to complete.
17. More user friendly
21. Easier to operate and understand
22. Speed, easier to learn, use
24. More user friendly
26. Easier, faster
27. More user friendly
31. Easier
33. Easy to move around
34. Mac - quicker to get around however most of my major is IBM compatible

<u>20205</u>
1. Window - but not Macintoshes - they're horrible. Windows is easier to use and more user friendly
4. I'm more familiar with these
5. Easier
6. Quicker to use
7. Easy to use screen like paper
9. More user friendly, less need to learn specific languages or commands
10. More user friendly, easy to use, (never used any other)
14. Window - seems easier to escape your mistakes
17. Faster and easier to understand
20. Window based for WYSIWIG
28. Easier to access
29. Easier
30. Faster, easier to use and understand
32. More user friendly
33. Mac is more user friendly
38. It is more "user-friendly"

<u>59211</u>
3. Easier to learn and use, quicker to learn
6. Options with sub-menus allow you to see all choices available(& divide up into areas of operation ..file, edit etc).
8. More user friendly
11. Prettier
12. Its easier to make less mistakes, i.e. typing error commands
13. You can do more things at once and don't have to remember all the commands (with pull down menus) but I really hate pull down menus without keyboard short cuts
18. Depends on GUI, like Archimedes desktop, prefer window based in general, but need keyboard shortcuts
19. Easier to figure out
20. Easier and looks better
21. User friendly
22. Easier to use
23. Faster to learn, easier to use
24. More user friendly
30. I prefer window based because I hate typing commands. Window based makes full use of menus
31. Easier to learn
33. Because that's the same as my home PC
34. A lot easier don't have to learn commands
36. Because it is relatively easy to remember commands (or lack of them)
37. They're mouse driven and more user friendly

<u>57102</u>
1. Other window based computers. Allows multi-tasking and faster processing, database paradox, use of graphics.
2. I prefer the Macs. Windows are confusing and very hard to follow procedures
3. You can keep every screen active at once
5. Currently prefer DOS based applications, am starting to get familiar/proficient in windows. Hate Macintoshes !!
6. Much easier to work with, don't have to remember codes etc
8. More extensive applications
9. Easier to understand and easier for inexperienced users
10. Easier, more efficient to use
11. Easy to use
12. Easier to use and understand ie. finding your way around
13. Ease of use
14. More straightforward and user friendly
15. It is easier to navigate around the system
20. You can look about for what you want
22. More efficient
25. Easier, less hassle
27. Faster processing time
29. Easier
31. Its easier to operate
33. Easier to follow

34. Easier to use
35. Mac- more user friendly
36. Makes it quicker to use
38. Windows because you don't need a brain
41. Its easier to use
43. Easier to work out where you are and what you're doing
44, Able to do more than one thing at once
45. Easier to do things on that you haven't done before
46. More user friendly
47. Easy to use
48. It is easy to move between files, and much quicker so you can swap between files
49. Easier
50. Easier
51. Much easier to use, not so much to learn

<u>60103</u>
6. Window based because it can be opened better than a text
7. Mouse is easier to move around - a lot quicker
12. You can look around until you find what you want
15. Easier
16. Easier
21. Faster, easier to use when you have got use to
22. Just easier I think
24. More user friendly
26. Easier to use, find your way around
28. Easier to use and understand
32. Its clearer and easier to comprehend and follow
36. Simple to follow and understand
37. Its clearer
42. More user friendly
43. More user friendly
45. Easy to use
47. Window based because it is easier to use and info easier to get
50. More user friendly
60. Much easier to use
64. Easier to use and generally quicker access using a mouse
65. Clearer, easier to read
69. Easier with a mouse

<u>59112</u>
1. More user friendly
2. Because its easier/quicker don't have to worry about typing exactly the right thing
4. They are more self explanatory and you don't have to type in
6. Easier to understand, can see where you are going
7. Can see where you are going program
9. Window based computers are more user friendly, create nicer user environment, however, can inhibit user
10. More logical - to me, easier

11. Because its much easier to learn. After that, only windows if it has lots of shortcuts
12. With a mouse is simpler and easier to remember commands
15. Is more visual, no commands. Text is more memory efficient, more into system
16. These are more user friendly
17. More convenient and simple to use
19. Easier to handle
20. More convenient to use
23. Easier to learn and remember how to use! Generally a little slower then text based however
24. Ease of operation, visual options make it quicker.
25. Much easier to use and quicker to learn, also more 'user-friendly'
26. Had more experience(easy to follow) - limited experience with text
27. Less typing makes it easier as I type only slowly
29. Easy to get to grips with
34. Easier to use commands by moving and clicking mouse
37. Easily understood, even to first-timers
38. Generally easier to use and learn
39. Less hassle typing
40. More user friendly
48. Prefer windows, not confusing
50. Easier to understand and see what's happening
52. Nice to use
54. Easier
55. Requires less effort to learn new system
62. User friendly
63. Easier to handle
65. Its quicker and easier to use
66. Easier to understand
67. Easier to try out all the different things you can do
72. More efficient
73. Less typing, faster
75. User friendly
78. More friendly atmosphere
80. Easier to read- clearer, more organised

Both windows and text

75305
19. Either is fine

59327
3. Window based systems are easier to get straight in and use but find text based quicker once I knew the commands
15. Both text based for speed or heavy processing power and windows when speed isn't that necessary and a lot of control isn't needed

<u>57302</u>
7. Depends - don't like Word Perfect for Windows, much prefer text
12. Depends on the application and the computer

<u>57202</u>
3. Depends on what application but I do have no preference since I want to be flexible and its good to be good at both
19. Mixture of both - using mouse with windows is convenient but can be slow
23. Window-based computers are more practical. Text - complex detail

<u>20205</u>
8. Either, I don't really mind
13. Window text explanations eg QUATTRO / command gives pull down window, easy to use
41. IBM - both

<u>59211</u>
10. Both, depending on what I am doing
29. Text good for controlling the finer details, windows good for getting something going quickly

<u>57102</u>
40. Both are good
53. Both have benefits
54. Either as long as clear instructions that definitely work (no errors)
55. It doesn't matter which, both are acceptable

<u>60103</u>
49. Both as they both have their advantages

<u>59112</u>
5. Window based are easier but text based is more flexible
8. I'd like both of them but I haven't got any chance to use window yet
47. Either / or depending on the situation

<u>Text</u>

<u>59327</u>
12. Windows at first while learning the system, but once knew system would rather use text

<u>57302</u>
2. Window's based at first - less instruction needed, easier to use. Text-based now - as it is quicker when you know what you are doing
3. Gui interfaces are less flexible and sometimes it slows performance
4. Window based easier but prefer text as this may be more useful in the outside world, ie when applying for jobs
8. Because I've had more experience, but windows are more user friendly
13. Prefer DOS machines. I know exactly what is going on. It's what I learnt on
15. I have not been exposed to a windows based PC other than the Macs

<u>60304</u>
3. Initially windows due to ease of use, lack of possibilities for errors etc. Later CLGs due to power (usual compsci answers !)

<u>45223</u>
16. I find text based easier to use but windows, once mastered has advantages over abilities of text based.

<u>57202</u>
4. Little use with windows, so still prefer text based computers.
28. First system I used and which I feel most comfortable with
29. Once the command language is learnt they are quicker and less clumsy to use
32. IBM PC or clone, first used and more flexible than Mac
35. More familiar

<u>59211</u>
16. Windows are slow and pathetic with no control
17. I use text most often and find jumping back and forth a pain
28. I'm used to it
35. Windows too slow, not that intuitive compared to a well written UI text program
38. Text hard to learn but quick once you know what you're doing Window - Macs always crash but easier to learn new programs

<u>57102</u>
17. Easier to use and less longer time spent than windows
23. Used it for years
24. More familiarity with them
56. Easier
57. Easier to use

<u>60103</u>
20. Easier
40. Very user friendly
46. Structured commands have more control

<u>59112</u>
3. Text (MS-DOS) is fast and simple, flexible, not limited options
12. I find windows based systems insulting to my abilities. They are childish
14. Typing is more fun than clicking
36. IBM more user friendly
42. More freedom and user control if you understand the text based system
64. If knowledge of commands exists, much faster
68. PC because they are faster - I prefer text applications
70. Window based isolate the user from full potential of the system

<u>Neither</u>

<u>60304</u>
1 find all incredibly difficult more time spent trying to find out how to use them working with calculator and handwriting

<u>45223</u>
8. I don't know enough about them to say- I've had little experience on anything other than word perfect.

<u>57202</u>
16. No preference, the more experience at a diverse range of products, the better

<u>20205</u>
22. Only tried windows a bit

<u>57102</u>
52. No preference

<u>60103</u>
10. No experience
14. No experience
18. No preference - do not use computers often enough for it to be of concern
31. I have used both but have no preference
33. Don't use computers except in physics 1a (24.101) and in library
41. Both are pretty average

<u>59112</u>
43. Haven't used both
44. Haven't used windows
45. Haven't used windows

H2 Responses to Question 22

<u>57302</u>
1. Deciding on my major.
12. Trying to explain that although I don't live with my parents, my home and term address are the same.
15. Keep paper numbers constant instead of naming one paper with two different numbers from year to year.. Stop changing paper numbers

<u>59HCI</u>
3. I had a problem when trying to "confirm" the enrolment. It was difficult to find out the place, time, and who I needed to talk to.

<u>59327</u>
1. The information in the academic timetable contained many mistakes. This made it difficult to plan my timetable.
2. Choosing between papers which I could take at either 200 or 300 year level.
4. Enrolling in a paper only offered every two years-Logic. Was not offered but still included in handbook.
7. Yeah- the faculty handbooks in the enrolment pack are outdated by enrolment time.
8. Not aware of cost until October (general staff exemption did not fully cover)
9. change degree from BTech to BSc or others, which papers can be cross-credited.
11. Changing papers during the year.
13. Calendar and department information conflicting regarding major requirements.
16. Change in courses meant I could not do S/W engineering this year but could have done it if I had chosen to last year (1991).
18. I wanted to take a 2nd half year paper, but wasn't sure how my workload would go. It wasn't possible to enrol for the paper after March even though the paper didn't start until the middle of term 2. If I did enrol in the paper, did 2 weeks worth, and then decided I didn't like it, it was too late to withdraw and get a full refund or pick up another 2nd half year paper.
20. Didn't know content and usefulness of computer science courses. Also had to take a course I didn't want to take(59.311) because there wasn't another half year paper suitable to me. OK though. Also wanted to know how involved courses would be. Exam times changed. Initial timetable said I had two clashes but I don't now.
25. Trying to change my enrolment later to a non-science faculty paper.

<u>75305</u>
2. No, but in past years when you get a piece of paper stating you cannot take a paper and must quickly decide on a new one.
3. Only the really long queues and the vast amount of problems arising from student finances.
4. Wanting to take extramural papers as an internal student.
5. Designing my degree and diploma so they fitted in and didn't clash and so I get it finished next year.

7. Not a lot of communication between enrolment office(s) and departmental staff.
8. Amount of time it takes, waiting in lines, trying to get around everyone -especially trying to sort out cross-credits from other institutions.
9. Yes, papers that were said to be offered this year were cancelled when I went to enrolment day.
10. It took some time for information to get through to my faculty on papers from another degree to be passed for my current degree. I presume this is standard however.
13. Incorrect information eg. papers listed when are not actually being offered. Papers listed as available internally and are only available extramurally (and vice versa). Timetable information mistakes.
14. Fees and allowances as classified adult student with dependent.
16. The Dean had set my course the year before then when I enrolled this year, he said I couldn't complete a double major like he had said I would be able to. Also problems with having to increase the number of papers of 200-level psych to complete this year-changes for number of courses needed.
17. Specifically in my first year-it is a very daunting task in which you are making decisions blindly. No matter what question you ask the lecturers or advisers never give you the right type of answer.
19. Just a basic lack of information about what would be the best papers to take in regard to the area I am interested in working in.

<u>60304</u>
1. Enrolled for MEd Admin and was informed two days before the enrolment day in February whether one was accepted or not whereas had been informed of possible place on other course some time earlier.
2. Only waiting in the queue.
4. Extremely long queues on enrolment day. Unclear information about costs of some papers and enrolment for special topics i.e. taking 300 level paper at 400 level as a special topic-cost of this unclear.
5. Not being able to take the papers which would benefit the sort of career I would like to enter into, because of majoring requirements, and so ending up with useless papers which you find hard, and tend to sort of give up on.
6. Each department pushes their own subject papers, rather than thinking what is best possible combination of papers for the student. Do not plan for papers to do in following year so that students have a future plan so do not become unstuck with pre-requisites.
8. Finding enrolment papers (application). Find out where it was located on enrolment day.
9. Last year I had problems resolving lecture and exam clashes. The bureaucracy involved in resolving exam clashes across faculties (Soc Sci and Hum) was particularly time-consuming and the arbitrariness of some decisions difficult to accept.
10. The long queues and waiting to go next. I am an overseas student and I had trouble looking for my enrolment form on enrolment day. Actually, work was not yet down to it and it took me two weeks to wait before I was legally enrolled.
13. Following the university calendar and trying to get papers which don't clash either in timetable and/or exams.

45323

1. No information freely given on the different opportunities there are for doing double major, double degrees etc.
3. I attended university in California as an exchange student for my second year of study and my finishing requirements are still very unclear- I am a third year and should graduate but its just not clear- I always get confusing info.
5. Too many people- not enough staff etc.

45223

1. 1991- they lost my enrolment form.
14. Had a doctors appointment I missed!
17. As I was new to Massey (from Victoria) I found that people weren't as helpful as they could have been in letting me know what courses etc. I could take and who to see etc.
18. Only standing in queues for long periods to get one person to initial a paper you wish to do.

57202

3. Yes, the enrolment procedure was somewhat confusing but otherwise a lot better than Victoria.
4. They wouldn't accept my brother's birth certificate for whatever ludicrous reasons! They sent me their own guidelines by mistake! And also sent me someone else's enrolment pack!
6. For overseas students enrolments are always very late some of them found they have enrolled after 2nd term is already gone. Procedure and details for overseas student not clean and has many human mistakes- very poor.
7. I have four papers with lecture clashes and two papers with exams on the same day. The faculty book could be very confusing especially for students doing double majors.
8. Waiting in line- very long lines. Knowing the pre-requisites for a paper I wanted to take
13. Student finances- second chance provision sorting out introductory lectures from permanent ones- course content- not clearly explained in some "prescriptions."
14. Records of cross-credits from Manawatu Polytechnic had not arrived at Info. Sys. Dept. from registry so I had to establish this fact with I.S. staff.
21. Money, signs, photocopies of lecture notes in library to solve, clashes in timetable.
23. Immediately after enrolment with exam clashes.
24. Mainly distinguishing how many papers I needed to do from each major in order to attempt a double major. HODs in computing and finance very helpful over the phone during the holidays.
28. The usual hassles of finding my way round.
35. Co-requisites.

20205

1. Long lines.
2. Having to take a certain amount of papers to keep up with 3 year degree pass. Passed 6 subjects 1st year but had to take 8 this year- enrolment day orders.

3. Big queues!
5. I have had to miss the "whole" of whole plant physiology lectures due to clashes.
6. Difficult to assess work load of papers.
7. For the 2nd year running- enrolled in wrong course major.
8. The massive queuing problems that entailed standing in a line waiting to see one person for 2 seconds while everybody else seemed to spend a practical age talking with them.
9. Standing in line. Being incorrectly informed. Being passed along when you have problems. Being made to feel stupid. Having to come back the next day to correct stuff-ups (every year of my enrolment)
10. Standing in lines for 3-4 hours on end. Finding out 1 single thing which would take 2 seconds to ask but having to wait hours. Availability of lecturers and deans. Having to come back and collect forms etc.
13. No-> Increase in fees half way through the year due to number of EFTs already done.
15. Queuing.
18. Late paper change-> many problems.
19. Reading and understanding the volumes of information relating to procedure for enrolment is largely an unproductive waste of time.
22. Late enrolment procedures and that because I'm an overseas student.
24. Just the lines.
25. No, talking to dean and professor was helpful as they let me take a paper I didn't have pre-requisites for; queuing to long for enrolment.
32. Long course approval queues.
33. Takes to long to get course approval- a whole day.
43. Waiting in a long queue.

<u>59211</u>
5. YES The F**K W*TS lost my enrolment form for the first 2 weeks even though I sent it in early so it was three weeks before my course was sorted out - had to choose other papers in a hurry
5. A set of blindfolds, a rifle and bullets and a brickwall for the people who were responsible for the present enrolment system - and those who run it - for the last three years there has been problems with my enrolments
9. I hate standing in lines
12. Only my name was recorded nothing else. Also a BA paper I took had not been crosscredited.
19. trying to take an 8th paper (Maths) with a"D" in management last year
29. Apart from changing from A BTech (INFO Eng) to BSc
36. Only the queue to see the dean

<u>57102</u>
2. I found it very stressful, friends at other universities had nowhere near as much trouble as I did having to go and visit the different faculties for their signatures.
5. Yes, course approval people did not know what papers could be taken in degrees that are spread across faculties. All my enrolment info was misplaced by Massey. Course approval forms were missing.

6. Yes, I had done some papers at Victoria University- this meant I could cross credit these papers towards the BVSc but the vet faculty asked me to enrol as if I hadn't done these papers i.e. normal 1st year papers. It was not until the day of enrolment that I had to decide what subjects to do in their place. It seemed to take a lot longer than necessary and would have been easier for the faculty to tell me these papers could be cross-credited and choose other subjects ahead of enrolment day.
7. Yes, I was exempt two first year vet papers. I was told this only on the day when I arrived at Massey for enrolment , even though my form with the relevant details had been sent weeks before-hand, I quite unexpectedly had to suddenly find two different papers to take and would have appreciated it if I had been notified before enrolment day, so that you can be prepared.
9 Finding lab streams, sore feet.
12. Finding out more info re individual papers and what they might lead to in following years.
15. Trying to decide the day of enrolment on 3 new papers to take in place of my exemptions. But I found the people very helpful when I explained what kind of papers I was interested in.
21. The main enrolment form was difficult to interpret. There were huge amounts of requested info that you had to give.
25. Transferring from extramural to internal.
30. Trying to fill out the enrolment form and declarations etc. They were very unclear.
34. Deciding on appropriate papers to take.
36. I was asked to take another course from the one I planned to do, because they thought it would be more appropriate.
37. Having to wait around, waiting to talk to staff members about exemption subjects, and having to go back and forth across campus for this.
38. Coming from out of Palmerston Nth it was hard to get any information. I just got chucked in at the deep end when I enrolled and even then no-one seemed to be able to answer my questions. I just kept getting referred to someone else.
39. Yes, choosing papers to replace those exempted from. It would have been much more helpful to have been told about my exemptions prior to enrolment.
40. Being enrolled as extramural when I wanted to be enrolled as internal.
41. Knowing which papers to take to get a good spread for my degree.
42. Had to send my birth certificate to the administration department about 5 times because it got lost or wasn't recorded.
43. Weren't told before enrolment day that I had an exemption. Then I only had 1 day to work out which other paper I was going to take. I didn't have my lab streams worked out which made finding a paper that didn't clash with my timetable almost impossible.
44. Weren't told about exemptions in Vet course until enrolment day, causing panic as to what other papers to take.
45. Yes, they lost all my forms for course finalisation- it took more than one week for anyone to know who I was.
47. Hard to change papers after deciding you didn't want to do one.
57. Not being on class records when I had enrolled.

<u>60103</u>

1. Trying to cross-credit papers. I found out that I was not enrolled in a paper 57102 two-thirds of the way through this year . I think this was because I was enrolling from the Hort faculty.
2. Trying to cross-credit papers from a previous Polytechnic course.
3. Being repeatedly asked for info during Christmas holidays that had already been sent and returned i.e. birth certificate.
8. Transferring from Otago, it was hard to know what the papers I had passed there were able to be used towards a degree here.
14. difficulty in co-ordinating all the different hand-books.
20. Working out timetable.
22. Study right crap. Need a degree to work out all the different aspects of it.
23. You almost need a degree to fill out enrolment papers. Working out fees and study right eligibility etc was most confusing, and working out what papers (like birth certificate) had to be sent to who.
26. I changed majors at the very start of the year but for half of this year I was not officially on my new class roll. This proved annoying at times.
29. Wasn't sure if I was exempt from principles of science or not. If I was I had to choose 2 other science papers which I did and luckily I was exempt from the paper or I would have been in trouble.
32. Not realising the background needed for some papers. Misinterpreting paper's structure.
37. Time was a factor having to wait too long.
38. Enrolment is at the wrong time of year- at Christmas the last thing you want to think about is Massey . Hence twice I have picked a load of bogus papers two years running.
40. Queues are far too long and no refreshments for lining up i.e. beer.
46. When we enrolled we weren't given any warning of the difficulty of some of the papers.
64. Yes- being an extramural and trying to transfer to the internal roll.
65. Enrolment form very cluttered.

<u>59112</u>

1. Understanding the timetable.
9. Previous qualifications not recognised from outside New Zealand (College Board of United States.)
10. Not well staffed i.e. massive queues.
11. Just general disorganisation, like small details which get mentioned only briefly but are quite important eg. exemptions.
15. Hell Yes! Didn't get enrolled with term two, only registered on 1/3 computers, no mentor, couldn't get papers signed by Dean, never! Couldn't see him, haven't yet. No locker register, no lock/key, nearly couldn't take 59.112, a course I had been attending for over a term. Get it together.
16. Yes- those damn queues!
18. Received my enrolment package late so everything was delayed for approx. 3 weeks to the beginning of lectures.
27. Knowing about co-requisites.
28. Planning a 'future study plan.'
35. Stupid enrolment system lost my file, gave me the wrong enrolment pack.

36. Whoever stuffed the enrolments at the start of the year should be shot. My enrolment, and course approval materials were four weeks late, stuck in the computer thanks to this. I have failed my computer subjects extremely badly, thanks a lot!!!
39. Unsure about exactly what documents to send with enrolment form/study-right application.
45. Not knowing if the papers I chose was alright.
56. Not knowing enough about the papers I'd intended to enrol in.
62. Finding the building on time.
69. They didn't have my form.
75. Long queues.
78. No only the queues- but that can't be helped.
80. Enrolment sheet is cluttered.

Index by type of complaint

<u>Mistakes</u>

<u>Documents</u>
59327: 1, 4, 7, 13, 20
75305: 9, 13

<u>Others</u>
45223: 1
57202: 4
20205: 7
57102: 5,42, 45, 57
60103: 3, 26
59112: 15, 18, 35, 36, 69,
59211: 5, 12
60304: 10

<u>Course Approval</u>
59HCI 3
75305: 3, 8
60304: 2 ,4 ,8, 10
45323: 5
45223: 14, 18,
57202: 8
20205: 1, 3, 8, 9, 10, 15, 24, 25, 32, 33, 43,
57102: , 2, 9, 37
60103: 37, 40
59112: 10,16, 62, 75,78,
59211: 9, 36

<u>Course planning</u>

<u>Paper planning</u>
57302 : 1
59327: 2, 20

75305: 4, 5, 19
60304: , 5, 7, 13
45323 : 1
57202: 7, 8, 21, 23, 24, 35,
20205 : 6
57102: 12, 34, 41
60103: 20, 32, 46
59112: 1, 27, 28, 45, 56

Cross-Credits
59327 : 9
75305: 8, 10
57202: 14
57102: 6
60103: 1, 2, 8
59112: 9
59211: 12

Changing
59327: 11, 25
75305: 2
20205: 18
57102: 6, 7, 15, 39, 43, 44, 47
60103: 29
59211: 11, 29

Criticisms
57302 : 12, 15
59327: 8, 16, 18,
75305: 7, 14, 16, 17,
60304: ,4, 6, 9, 13
45323: 3
45223: 17
57202: 3, 6, 13,
20205: 2, 9, 13, 19, 22,
57102: 5, 21, 25, 40, 30, 38
60103: 14, 22, 23,38, 64, 65
59112: 15, 39, 80

H3 Responses to Question 31

57302
3. Lecture timetables, course controllers.
4. Could possibly have information concerning lecturers.

59HCI
2. Timetable of lectures, labs etc. for a chosen paper. who I needed to talk to.

59327
1. It might be useful to have some way of informing users about possible mathematical or statistical background that was assumed in some papers.
5. A list of staff to contact in regards to the courses i.e. don't rely upon one source of information, I prefer to talk to the staff.
6. The ability to slot in papers at will and to see resulting lecture time table including any clashes.
7. Lecturers/course controllers for papers.
8. Work backward from desired advanced level papers through prerequisites and recommended background.
9. Change degree from BTech to BSc or others, which papers can be cross-credited.
10. I hate staring at VDU's, I want to get the relevant info for me on hard-copy - so I can take it away to mull over.
12. Descriptions written by other students who have previously sat the paper.
18. What the average class size might be-based on previous years. The pre/co-requisites for the paper etc.
19. Internal/Extramural options and conditions for both. Paper Info
20. Information on lecturers and lecture style and textbooks and cost and study guides and what computers are used and if home PC's can be used and if S/W available for home PC's. Timetable/exam info. Very important to have an easy to navigate system eg. Hypercard designed well.
24. A menu with a trace.
25. Timetable shown.

60304
3. Lecturers on course so I know who to avoid.
4. To see who are the lecturers for the papers.
6. The credits assigned to each paper and other possible combinations of papers that add to the same number of credits.
7. Working out lecture timetable. Pass-rates from previous years for papers.

45323
3. Specific career advice for a degree.

57202
3. What about student allowances; I know this has nothing to do with enrolment on a course but it would be nice if such a thing was computerised/automated.
8. Names of lecturers, timetables, exam dates.

17. Info about who to contact about papers and advice. Staff contact
27. Info about sport within the university. Info about the help services i.e. workshop available.
35. Numbers enrolling (past years.) Pass rates.

<u>20205</u>
1. Approximate numbers enrolling in a paper. A general help for the computer for people who haven't used it before- maybe a message system so you could ask questions of lecturers etc. Como
Paper Info
2. Place sample test papers from exams (previous years), another with answers so we can test ourselves on the computer before exams.
4. Names of those lecturing the papers.

<u>59211</u>
25. Email
30. prerequisites, corequisites etc
31. previous years pass rates of each course

<u>57102</u>
20. Statistics for the pass rate and for the number of people that do get jobs in a particular field.
Paper Info
22. Required book lists; course commencement dates.
38. Saying where to go and when and who to see for particular questions.
45. Modem facility so that it can be accesses by peoples P.C's to save coming all the way to Massey until lectures start.
55. Bursary payment dates.

<u>60103</u>
12. A second hand book register. Tutorial time-table and room numbers for all subjects. Exam dates for whole year, terms tests and finals.
15. Possible assistance with papers and problems.
27. Cross-credit info and course change possibilities.
32. Background needed.
44. Timetable- correct one. Note of students which pass/fail course.
64. Support services available for a given paper.
66. Timetable alternatives of chosen course.

<u>59112</u>
6. What careers need what papers.
10. Corequisites-prerequisites-on computer information.
11. Timetable data.
12. Seeing what subjects you can major in with your current subjects, points/credits.
16. What papers can be chosen in the future with the current 1st year papers.
27. Some way of looking at different degree courses leading to similar careers.
33. Use modem to get enrolment information over phone, on home PC.
51. Information about fees, loans, eligibility for allowances and scholarships. Perhaps information about prospective employers for holiday jobs.

61. Papers you can take outside my faculty. Good combinations of papers.
62. Job prospects.
64. Information on lecturers.

<u>Index of requests</u>

Timetables
57302: 3
59HCI: 2
59327: 6, 20, 25,
60304: 7
57202: 8
60103: 12, 44, 66
59112: 11

Lecturing staff
57302: 3, 4
59327: 7., 20,
60304: 3, 4
57202: 8
20205: 4
59112: 64

Contact people
59HCI: 2
59327: 5
57102: 38
57202: 17

Planning
59327: 8, 9
59112: 6, 12, 16, 61
60304: 6
60103: 27.

Paper information
20205: 31
60103: 15, 32
59327: 1, 12, 18, 19, 20
57202: 35.
20205: 1, .2
57102: 20, 22
60103: 31, 44, 64
59112: 10.
59211: 30, 31
60304: 6, 7.

Computer related
59327: 10, .20, 24
20205: 1.

57102: .45.
59112: 33.
59211: 25.

Career advice
45323: 3.
59112: 6 , 62
59112: 27.

Finance
59327: 20.
57202: 3.
57102: 55.
59112: 51.

60103: 12
57202: 27.

Appendix I

Detailed analysis

I1 Enrolment questionnaire - spreadsheet, first year students, 1993

	1	2	3	4	5	6	7	8	9	10
1	Age	Sex	Macintosh	PC	Wp	DB	Spreadsheet	Prog lang	text	Window
2	19	M		Y	Y					
3	17	F		Y	Y					Y
4	19	M		Y	Y	Y	Y	Y		Y
5	17	F	Y	Y	Y		Y	Y		Y
6	17	F		Y	Y	Y	Y	Y		Y
7	26	M		Y						
8	18	M					Y			
9	20	M	Y	Y	Y	Y	Y	Y		Y
10	18	M	Y	Y	Y	Y	Y	Y		Y
11	18	F		Y	Y		Y			
12	18	M		Y	Y		Y	Y		
13	17	M	Y	Y	Y	Y	Y	Y		Y
14	18	F	Y			Y	Y			
15	18	M			Y					
16	20	M								
17	18	M		Y	Y	Y		Y	Y	
18	23	F		Y	Y	Y				
19	18	F		Y		Y	Y			
20	18	F	Y	Y	Y	Y	Y	Y	Y	
21	18	F								
22	17	M								
23	20	M		Y	Y	Y	Y	Y		Y
24	19	F		Y	Y	Y				
25	18	M	Y	Y	Y			Y		Y
26	18	M	Y	Y	Y		Y	Y		Y
27	19	F								
28	23	M		Y	Y			Y		
29	18	M	Y	Y	Y	Y	Y	Y		Y
30	24	M		Y	Y	Y	Y	Y		
31	18	M	Y	Y	Y	Y	Y	Y		Y
32	23	F		Y	Y					Y
33	21	M	Y							Y
34	18	M		Y	Y		Y	Y		Y
35	18	F			Y			Y		
36	18	M			Y	Y	Y			
37	19	F								
38	18	F		Y	Y	Y				Y

	11	12	13	14	15	16	17	18	19	20
1	Enrolment	UC	FH	DH	Liaison	Dean	Prof	Lect	Students	Other
2	1		1		4				7	PARENTS
3	5	2	6	3	1	3	3	3	7	
4	7		6	6	5		5			
5		4	4						3	
6			4						6	
7	5	5								
8	7	5	5	6	5				7	
9	7	7	7	7	7		7		3	
10	6	2	5		5	4		4	4	
11	5	4	4	7	2	4	6	4	6	
12	1	1	4	4	7	1	4	4	7	
13	4	1	6	6	7	4	6	6	4	
14	5	6	5	4	6	5	5	5	7	
15	6	5	7		6	6				
16	6								6	
17	4		6							
18	6	6	7	6	6				4	
19	5	5	4	4	7	5			7	
20	6		5	5			6		5	
21	6								7	
22	4	1	2	2	4	5	5	5	7	
23	5	4	6	4	4	5	5	5	6	
24	7	3	6	6	2	7	7	7	6	
25	3	4	5	5	6		4	5		
26	4	4	4	4		5	5	4	6	
27	6		7	6	4				3	
28	7	7	6	5	7					
29	4	5	4	4	6	5	5	5	7	BENNETS
30	7	6	7	6	7	7	7	6	5	
31	4	6	5	4	3		5	3	5	
32	6	1	5						7	
33	5	5	5	5	4	5	6	5	6	
34	4		6						6	
35	7	4	4	4	4	5	3	4	7	
36	7		7		5			5	5	
37	6	4	5	5	4	6	6	6	6	
38	6	5	5		7				7	

	21	22	23	24	25	26	27	28	29	30	31	32	33	34
1	17	18	19	20	21	Other	23	24	25	26	27	28	29	30
2							Y	4	6	6	6	6	6	6
3							N							
4							Y	5	6	6	6	5	5	5
5							Y	3	6	5	7	4	7	7
6							P	3	7	5	5	5	3	4
7							Y	7	7	7	5	7	7	7
8							P	2	7	7	5	7	7	7
9							Y	7	7	7	7	7	7	7
10							Y	5	5	5	5	5	5	5
11							P	7	7	3	5	5	6	7
12							P	7	7	7	7	7	7	7
13							Y	4	2	4	5	7	7	5
14							P	4	7	6	7	6	6	7
15							P	7	7	7	7	7	5	5
16							P		6	5	5	3	5	4
17							N							
18							Y	5	7	7	7	6	7	7
19							Y	7	7	7	5	5	5	7
20							N	7	6	1	1	4	3	6
21							P	7	7	7	7	7	7	7
22							P	4	7	6	6	4	5	5
23							Y	5	7	7	6	2	2	5
24							Y	5	6	6	6	5	5	6
25							Y	4	6	5	5	6	5	6
26							P	2	7	6	6	5	5	5
27							P	7	7	5	5	7	5	7
28							Y	6	7	7	7	7	7	7
29							Y	4	7	7	7	7	7	7
30							P	7	6	4	6	6	3	6
31							Y	5	2	5	3	5	5	5
32							P	5	5	5	5	3	6	3
33							Y	5	5	4	5	5	6	4
34							Y	3	6	5	5	5	3	4
35							P	1	6	7	5	7	6	7
36							P	5	7	7	7	5	6	7
37							Y	7	7	7	7	7	7	7
38							Y	6	7	6	6	6	3	5

I2 Student choice of paper

The following table (I.1)shows the choices made by each of the Computer Science students (referred to as A, B, C, etc). This information was taken from the university's Master Roll of Internal Students.

Paper Number

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	P	Q	R	S	T	U
21103			Y																	
23101	Y																			
23102	Y								Y					Y		Y				
23103													Y	Y					Y	
23104					Y															
24101		Y			Y	Y	Y		Y	Y	Y		Y	Y		Y	Y			Y
24102	Y	Y			Y		Y		Y	Y	Y	Y	Y		Y	Y	Y			Y
24240	Y															Y				
26120				Y	Y															
34101			Y								Y							Y		
34103																				
39101						Y												Y		
39106																		Y		
40100										Y										
45121								Y												
59111	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
59112	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
60101		Y		Y		Y	Y		Y	Y			Y				Y			Y
60102		Y		Y		Y	Y		Y	Y	Y	Y				Y	Y	Y	Y	Y
60103			Y		Y			Y						Y						
60104	Y			Y							Y							Y	Y	
60203												Y			Y					
60204												Y								
60208															Y					
61120			Y					Y						Y			Y			
61100		Y		Y		Y						Y							Y	Y
64111		Y											Y							
64341							Y												Y	
64342							Y												Y	
75101															Y					
75102			Y	Y				Y		Y	Y							Y		

Students' choices

Table I.1 First year students' choice of papers

I3 Information from Science handbook

Students who have decided upon their majoring subject are told to write down from the Schedule the majoring requirements for their main subject, including all prerequisites and corequisites. They should then choose several more 200- or 300- level papers to support or extend their major subject so that in total these papers are worth 160 200 and 300 level points. Only 48 300-level points are needed but more can be taken. Students are advised to check that they have at least 84 100-level points from the BSc schedule and an overall total of 300. If not, they should choose additional courses to fulfil the requirements. Finally, the courses chosen should be divided into three years of study aiming to obtain 98 or 112 points in the first year(100-level papers), 98-112 points all from 200-level papers or 98-112 from 200- and 100- level papers and all the remaining points in the third year. Those unsure of their majoring subject should plan a first year that includes the majoring requirements of several subjects. In their second year they should preserve the choice of at least 2 majors (p7, 1992). The regulations for the Computer Science major are shown in Figure I.1.

The relevance of points is explained to students. The point value of a paper indicates the total amount of time (for lectures, laboratories and study time) that a student may expect to spend on the paper to satisfy assessment requirements. For a full year's course each point roughly corresponds to one-half hour per week; for a 14 point paper such as 59.110 the workload would be about 7 hours per week.

First year
59.111, 59.112, 60.102 plus 70 additional points.

Second year
59.2111, 59212, 59.213, 60.208 plus three other papers, at least two of which should be 200-level papers from the BSc schedule.

Third year
59.311, 59.312 plus at least 16 points from 59.313, 59.324, 59.325, 59.326 and 59.327 together with additional supporting papers to complete the requirements for the degree.

Figure 1 Computer Science template

I4 Course outline for 59324

59.324 Knowledge-Based Systems

Course Outline 1993

This course provides an introduction to the basic concepts of knowledge-based systems. Important topics such as knowledge acquisition, knowledge representation and how to construct an expert system are all tackled.

Prerequisites
59.212 or 57.202

Practical Work
Students are expected to build an expert system using Nexpert Object on the Sun.

Assessment

Assignment 1	April	9	10%
Assignment 2	June	18	40%
Final	November	6	50%

Terms
Both assignments must be completed.

Textbooks
'Expert Systems and Applied Artificial Intelligence',
E Turban, Macmillan.

Lecture times

Tues	1pm	SSLB3	1/2
Thurs	12pm	SSLB3	1/2

Lecturers

Elizabeth Kemp	SST1.12	Ext 8574 (Course Controller)
Dr Daniela Stavreva	SST1.20	Ext 8781

Exam
6 November 1993

Effective weekly hours
This course carries an 8 point value. Students should expect, for the duration of the paper (half an academic year) to spend about eight hours per week on the course.

2 hours per week lectures
2 hours per week reading
4 hours per week assignments

I5 Interview with the Dean of Science

In an interview, the Dean of the Faculty of Science agreed that there were problems with the enrolment process. Despite all the time and effort put in by staff before and during the course approval process, he and his secretary spend much of their time in the first month of term one sorting out problems with students who wanted to change papers. Enrolment could be greatly simplified if the Science faculty had a few preset courses (as has the Technology faculty) or if students were not allowed to alter a course once approved. Changes of this kind though were opposed by the academic staff since they did not give students the flexibility to construct an appropriate course of study. If students were still to be given a large degree of choice, an advisory system was feasible for those students without significant problems. Many students though who enrol for a Science qualification have changed from other degrees (Eg BVetSci) and need much assistance to choose a major and build up a course of study.

Some more detailed issues were also discussed. It was not clear from reading the Science faculty regulations how a restricted pass in a subject was handled. A restricted pass means that a student failed to obtain 50% in a paper but can be credited with an "R" grade when the mark falls in the range 47 to 49. Nonetheless it still counts towards the degree. The Dean stated that there is no limit theoretically to the number of "R" passes that can be credited to a degree. In practice, students do not satisfy faculty regulations for graduation with more than three "R" grades but the Science faculty board decided not to formalise this in the regulations. An "R" pass in a subject does, though, prevent a student from taking a paper for which it is a prerequisite. Otherwise an "R" pass in a subject is sufficient so long as it is not a prerequisite. For example, when a Calculus paper is needed for the Electronics course then an "R" pass in the Calculus suffices.

The Dean finds it very useful when assisting students to know their points to date. This allows him in conjunction with a points template for the BSc (see appendix) to work out what else needs to be done. If a student has not yet obtained the necessary 84 points at first year then a suitable paper of the correct points value can be suggested to a student. Such templates are also useful for long term planning and show what papers should be taken at the first, second and third year.

Finally, the interview allowed the knowledge engineer to find out how students were timetabled for laboratory sessions. A student who is taking a first year Physics paper has to attend a three hour laboratory session each week. Students, though, do not select which session to attend. Staff with a great deal of experience in this area allocate students to what initially appears to be the most suitable slot. What happens in practice is that a few students cannot attend their allocated session and have to see the staff for a change to be made. An advisory enrolment system, therefore, would not have to check that there was a laboratory slot available for a student but just ensure that there were some laboratory sessions that a student could attend.

Appendix J

Extended model of communication

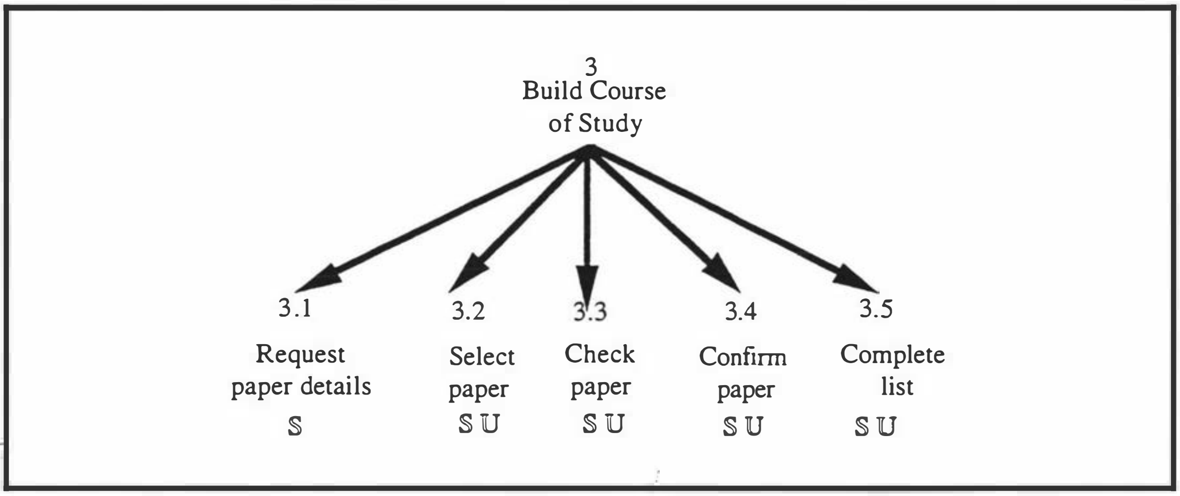


Figure 1 Build Course of Study

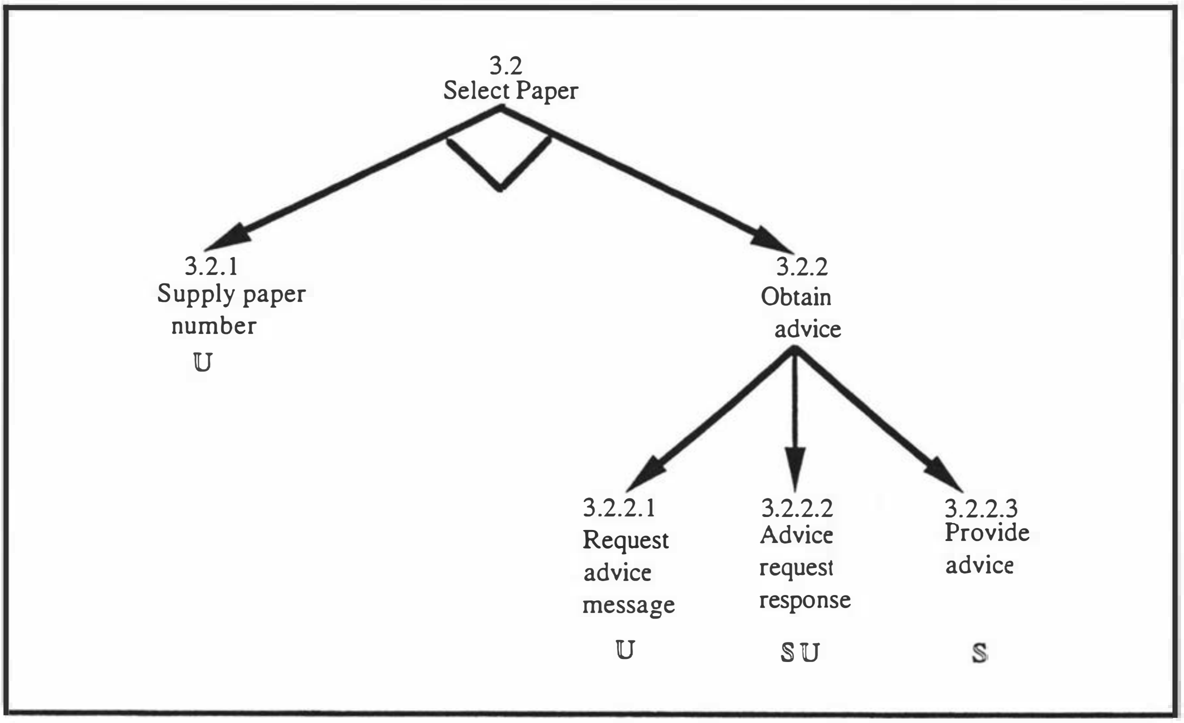


Figure 2 Select Paper

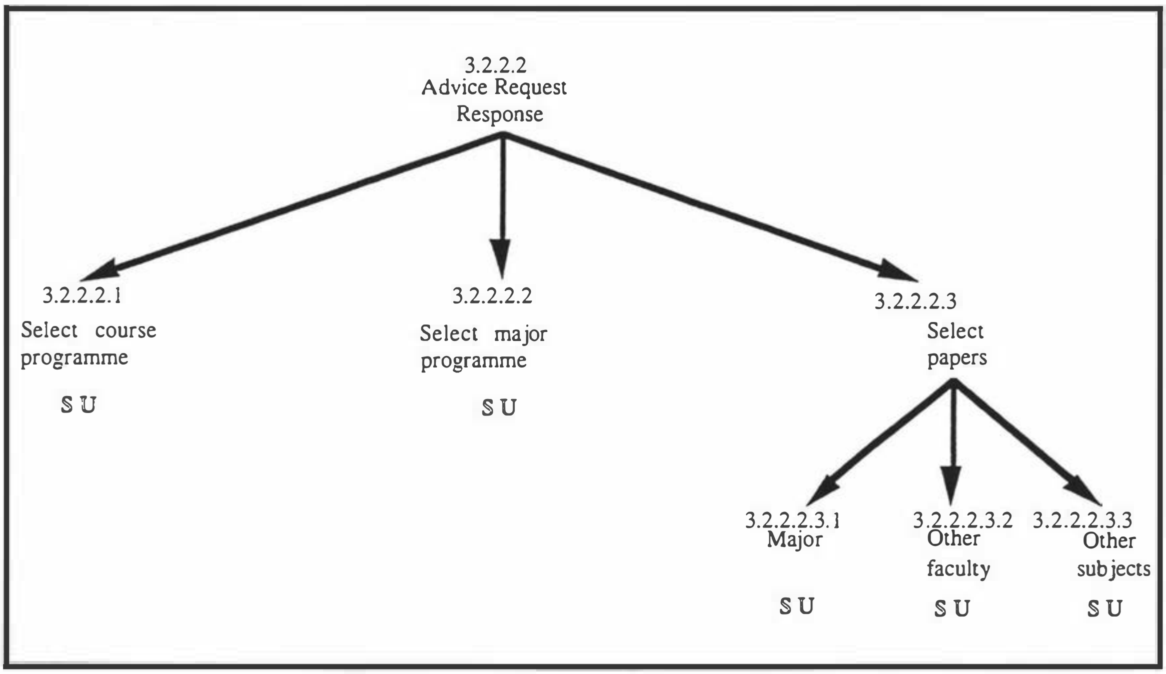


Figure 3 Advice Request Response

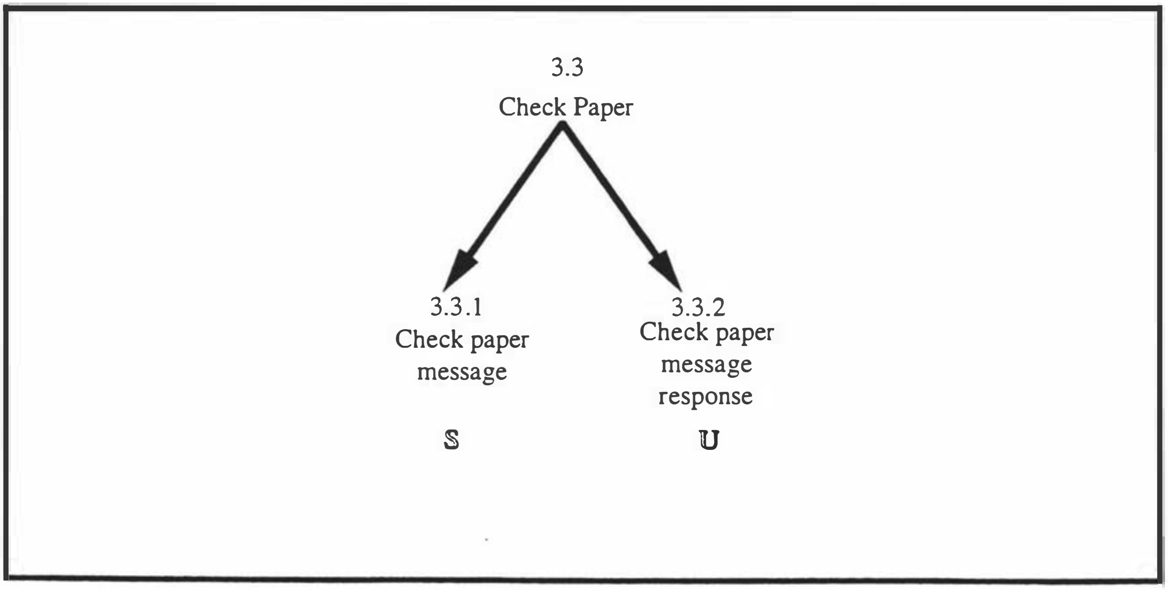


Figure 4 Check Paper

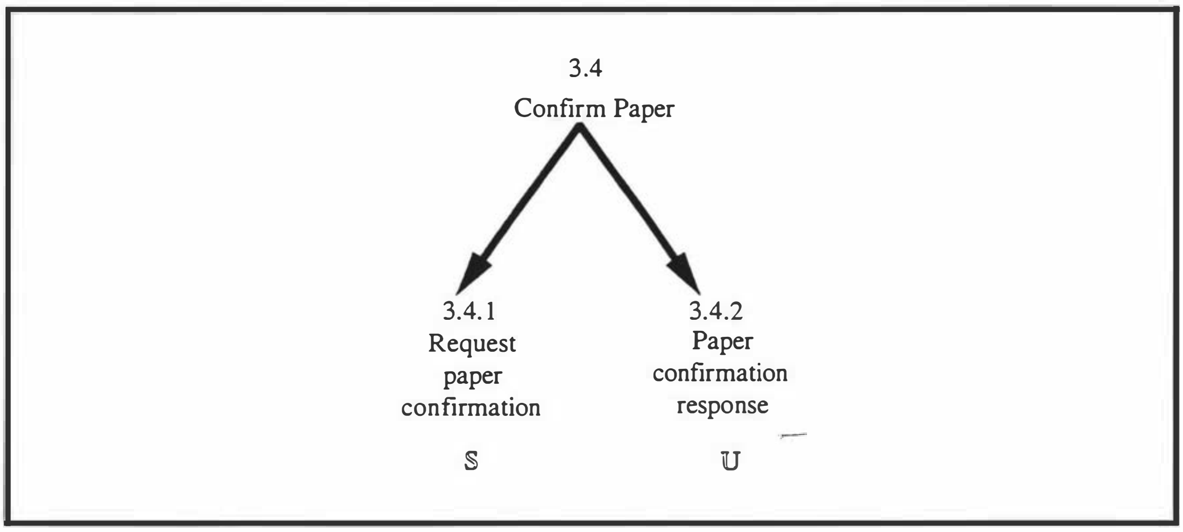


Figure 5 Confirm Paper

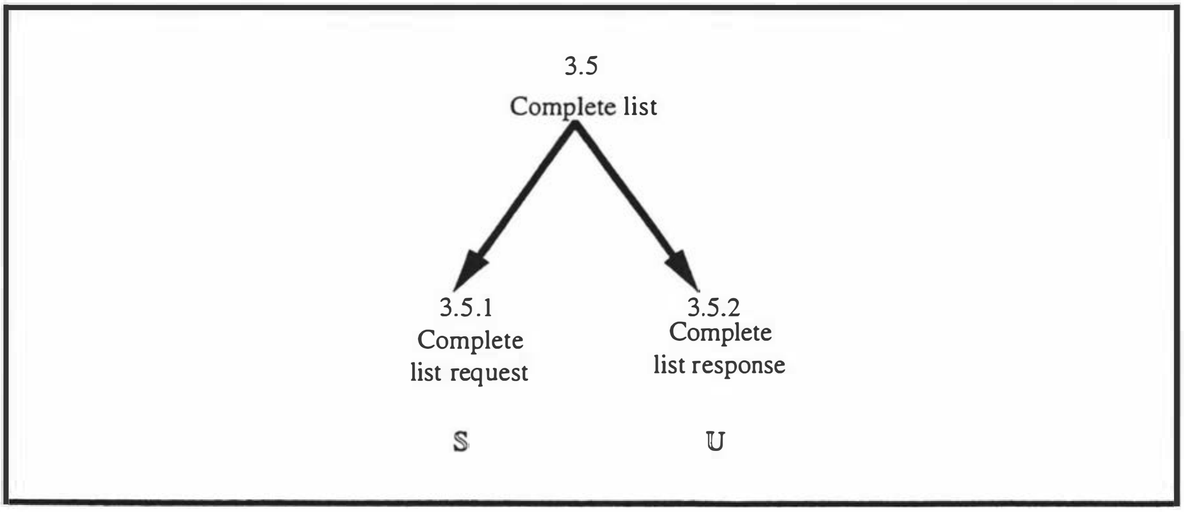


Figure 6 Complete List

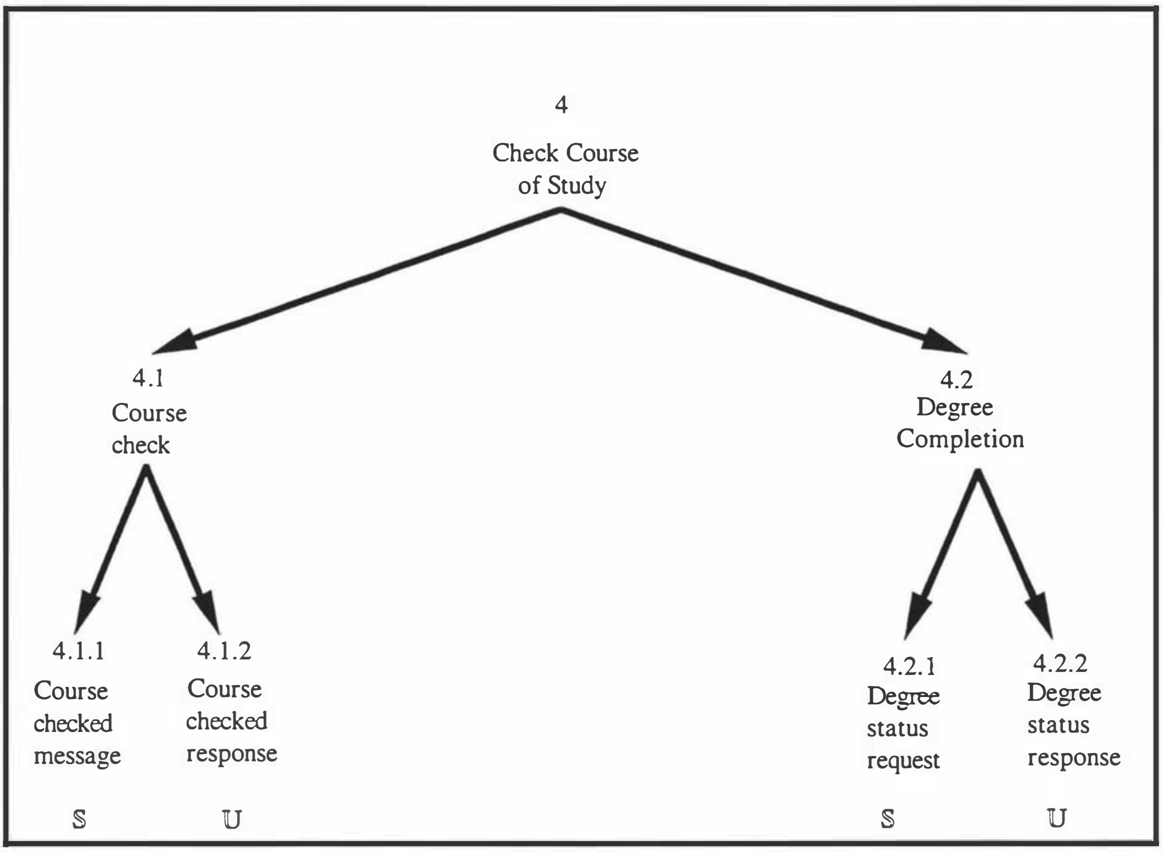


Figure 7 Check Course of Study

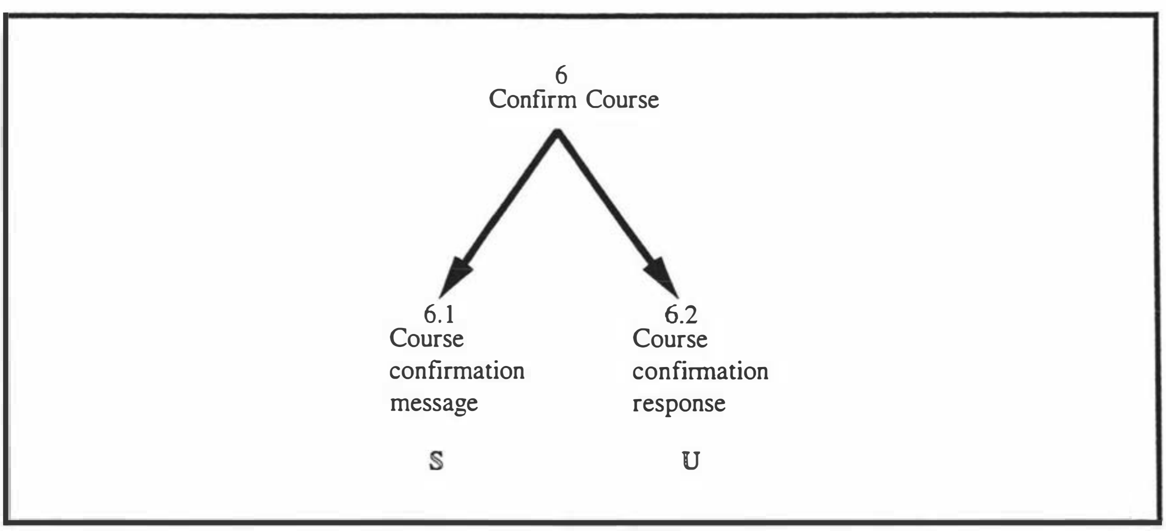


Figure 8 Confirm Course

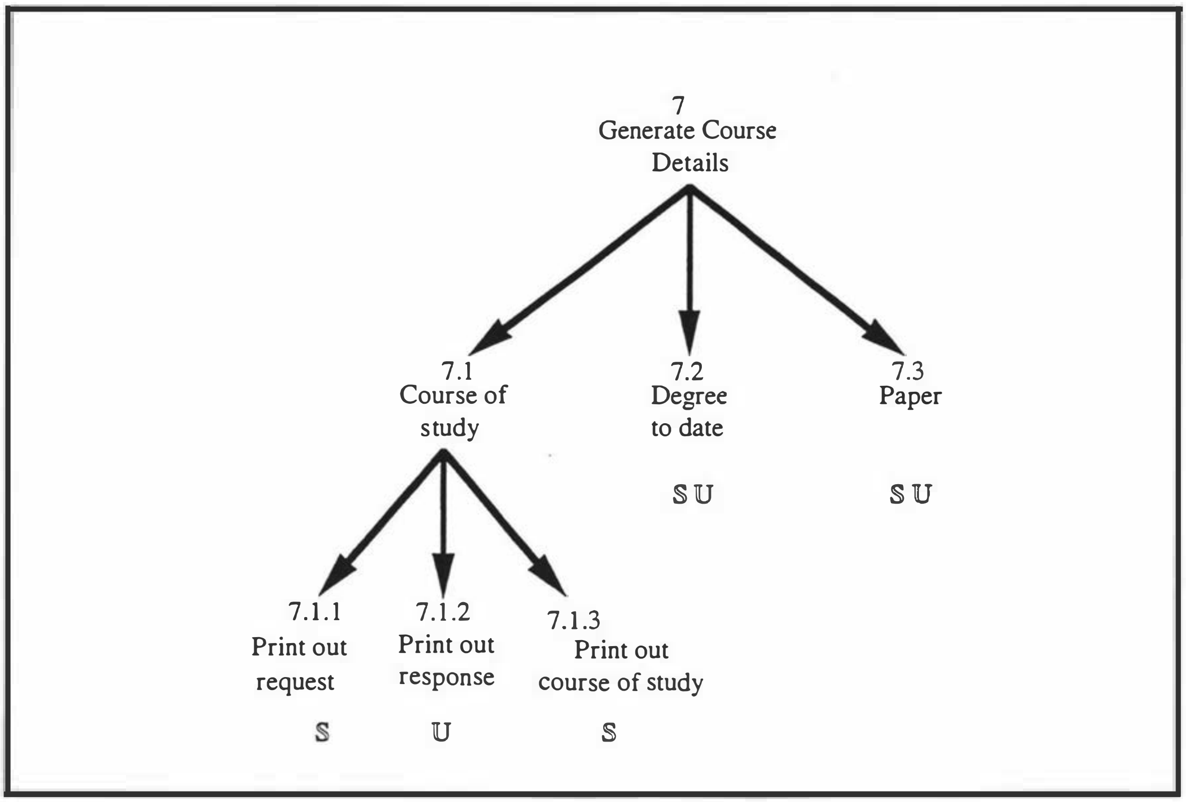


Figure 9 Generate Course Details

Appendix K

Storyboards

Storyboards

Users can scroll through the screens shown in Figures 1 to 4 to find out more about the enrolment system.

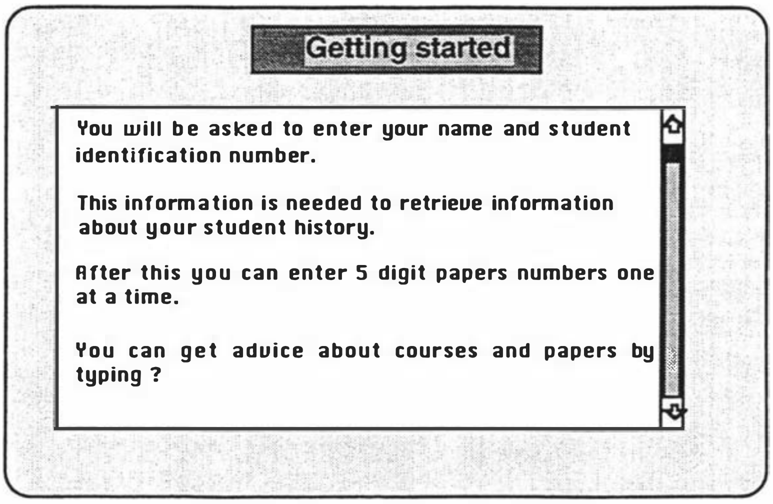


Figure 1 "Getting started" 1

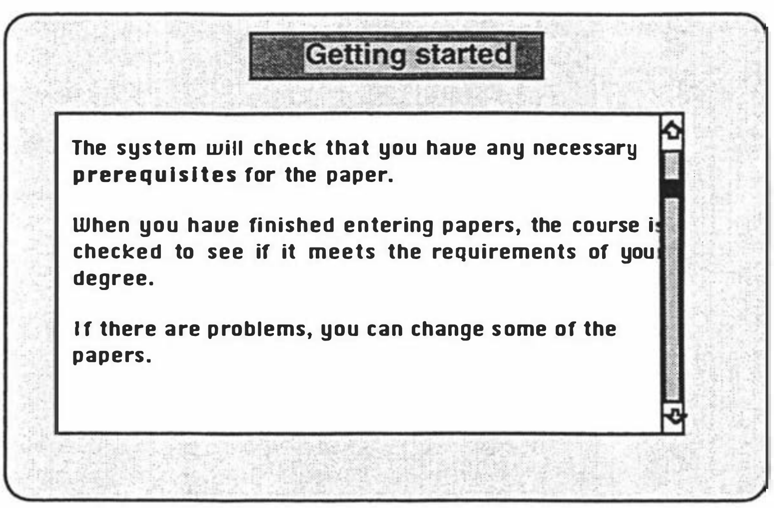


Figure 2 "Getting started" 2

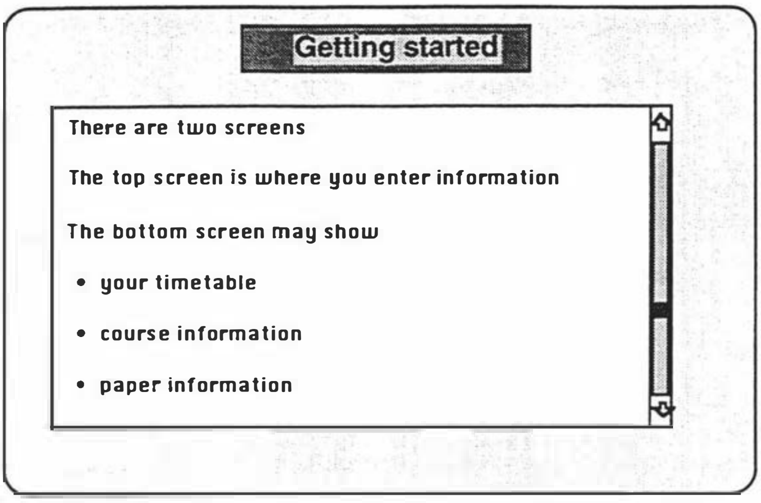


Figure 3 "Getting started" 3

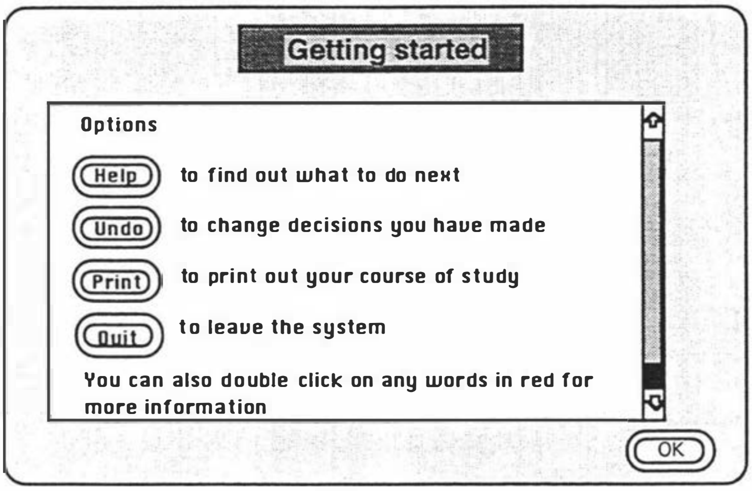


Figure 4 "Getting started" 4

The Help and Quit options are available when students are scrolling through the introductory material. These functions can both be accessed at all times. Quit obviously allows students to leave the system. Help messages are context-dependent and in this case informs students to select OK if they wish to start entering enrolment information or Quit if they wish to leave at any time. The only highlighted term that appears is the word "prerequisite". If students double click on this, the definition is shown in a dialogue box.

Student Information screens

Students enter their name and identification numbers (Figure 5). Help is available, even though, as Figures 6 and 7 show the instructions are self-explanatory. The only problem that may arise is the terminology "given" and "family name." The Help message that appears before OK is selected explains this terminology as well as explaining why this information needs to be entered.

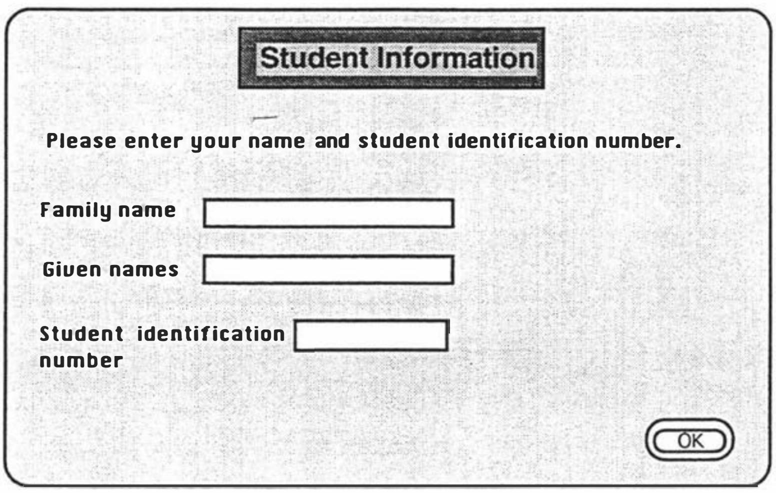


Figure 5 "Student Information" 1

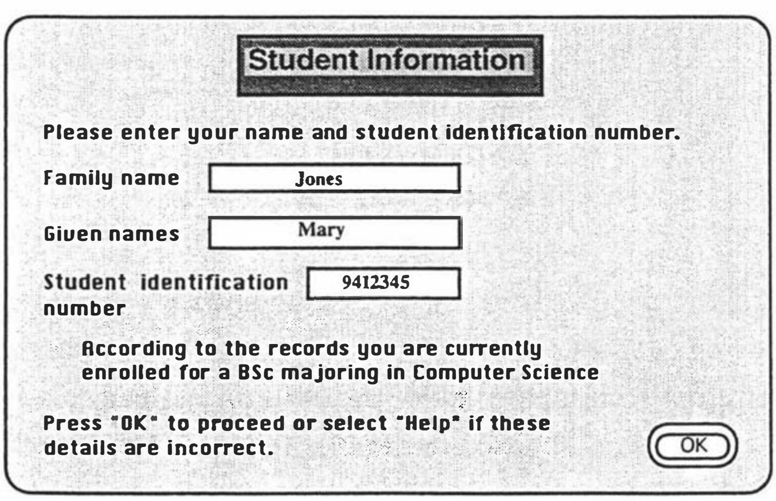


Figure 6 "Student Information" 2

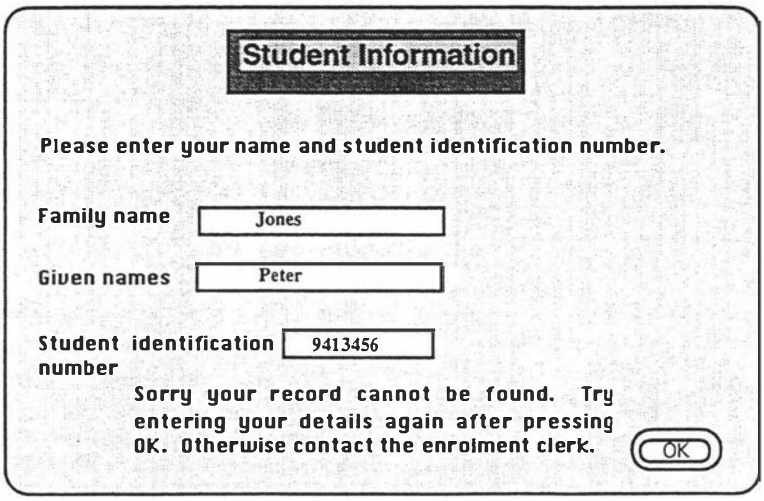


Figure 7 "Student Information" 3

Paper Selection

Students enter paper numbers one at a time into the system. Figures 8, 9 and 10 show a typical sequence of actions. First the student enters a paper number and the system checks that the paper can be taken. If the paper can be added to the course of study, a student is asked to confirm that he/she wishes to take the paper. When Yes is pressed the paper is entered into the course list. A student is then asked whether he/she has completed their course of study. If Yes is chosen the system moves onto checking the course. Otherwise, the information relating to the current paper is cleared and another paper number is entered (Figure 11). Finally, if a student wants to change the course of study they can use the Undo option to remove a paper.

The Help message available when students reach paper selection (Figure 8) explains how to enter a paper number and how to complete their course of study. When a Yes or No button is activated, the associated Help explains the significance of the decision students have to make.

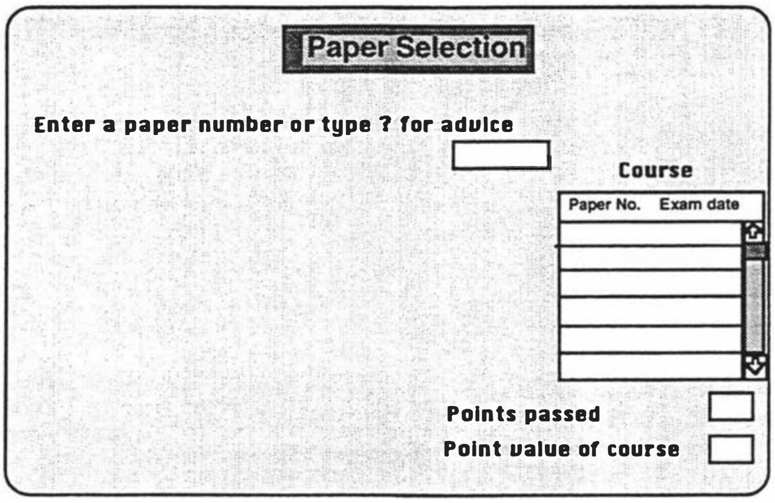


Figure 8 "Paper Selection" 1

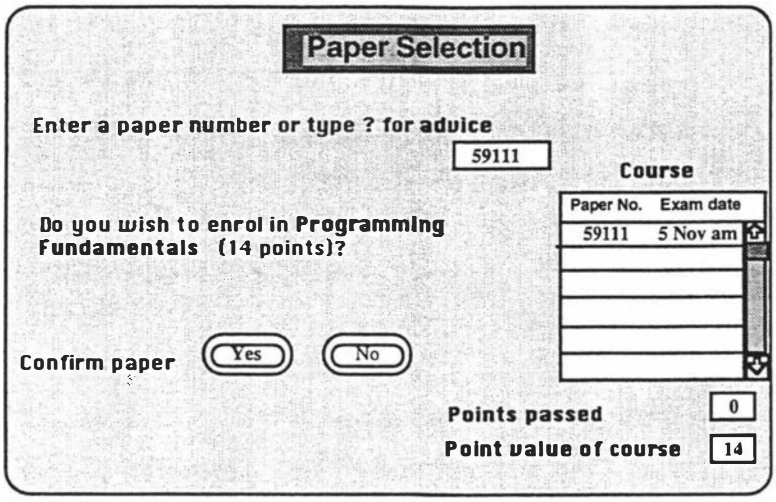


Figure 9 "Paper Selection" 2

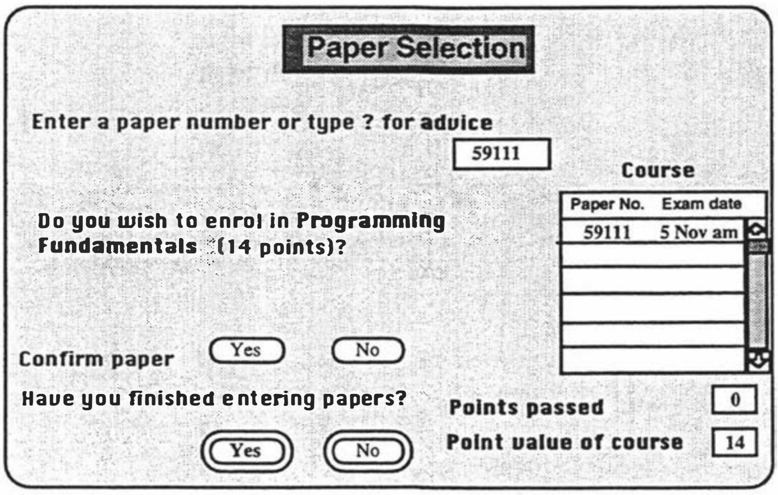


Figure 10 "Paper Selection" 3

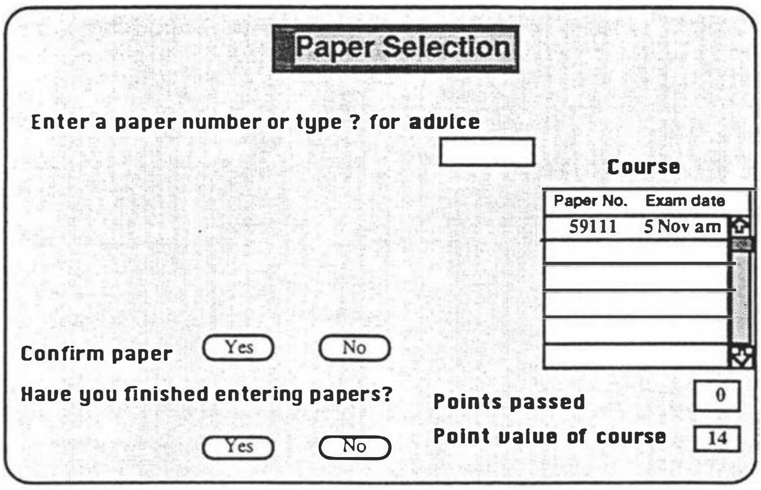


Figure 11 "Paper Selection" 4

Students for one reason or another (Figures 12 to 15) may not be permitted to take the paper chosen. If problems such as an examination clash arise, the Help message explains what options are open to a student. Definitions of highlighted terms like "examination clash" and "restricted" can be obtained by double clicking on the words.

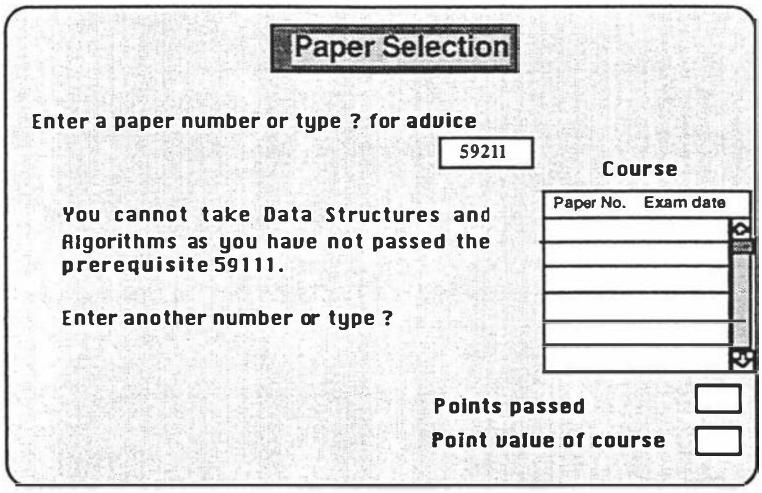


Figure 12 "Paper Selection" prerequisite message

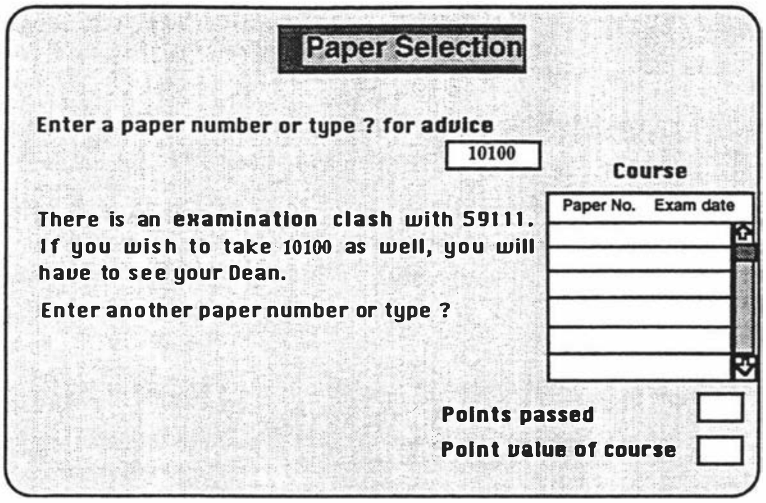


Figure 13 "Paper Selection" examination clash message

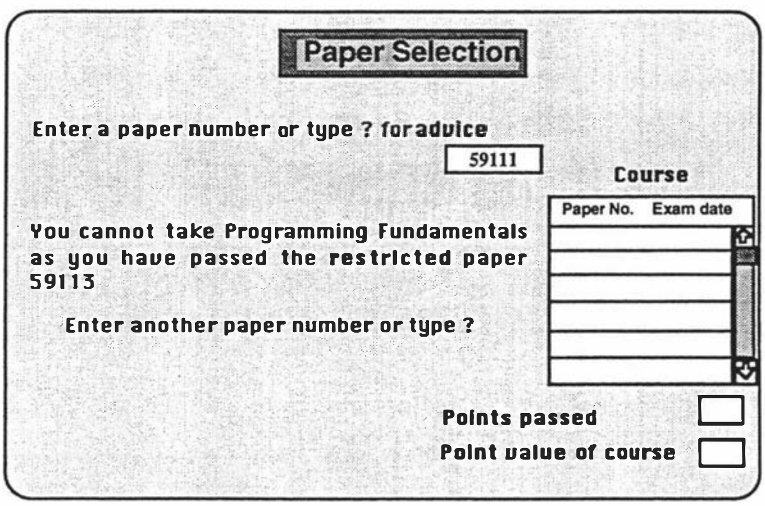


Figure 14 "Paper Selection" restriction message

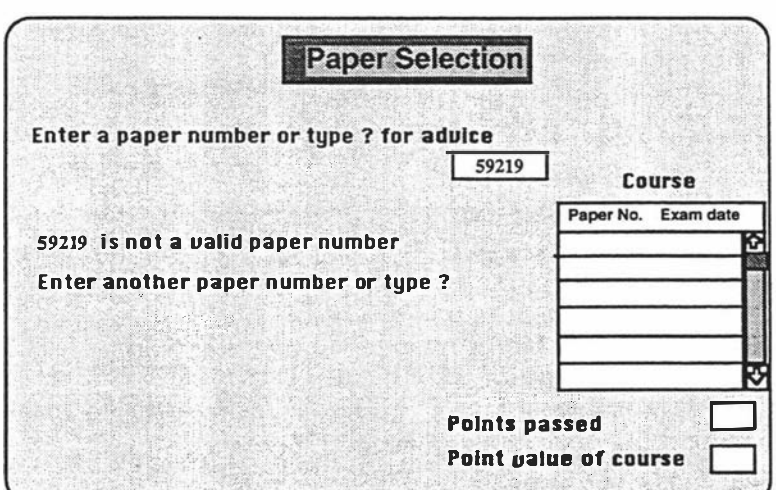


Figure 15 "Paper Selection" invalid paper number message

Course check

The knowledge-based system checks the course of study to see whether it meets faculty regulations (Figures 16 to 17). If the course of study fulfils all the requirements students are asked to confirm their enrolment. When Course Check is reached, a help message can be obtained which tells students about the nature of the checks that are being made. When students have to choose whether or not to confirm their enrolment, the context-dependent Help informs them of the significance of the decision that has to be made.

If a course of study does not meet the faculty requirements (Figure 18), instructions about how to proceed are given on the screen but can be amplified in the Help message.

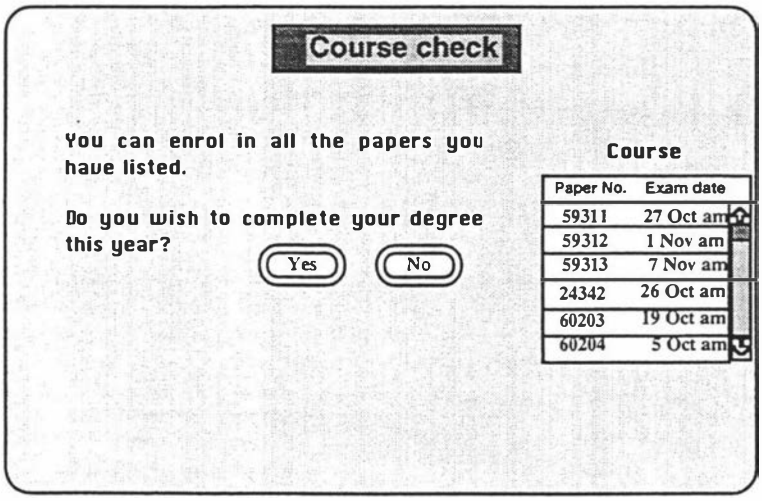


Figure 16 "Course check" 1

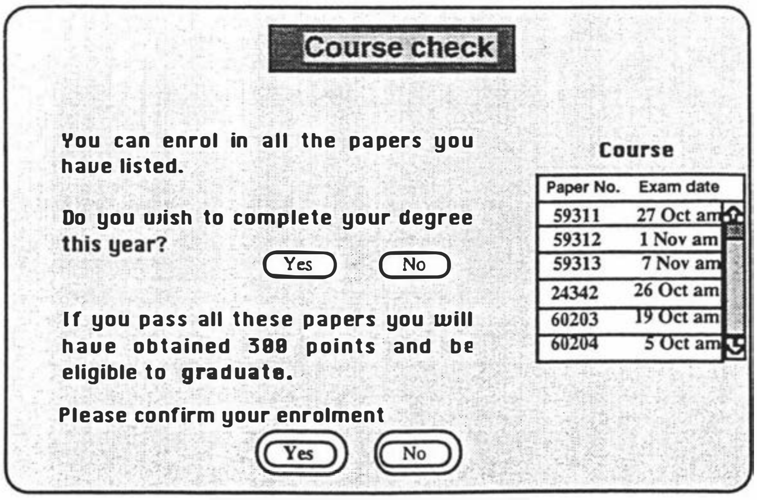


Figure 17 "Course check" 2

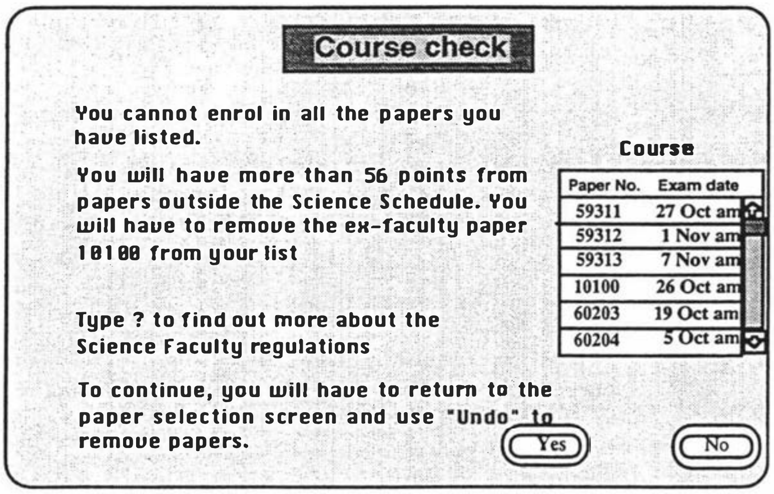


Figure 18 "Course check" 3

The Farewell screen (Figure 19) gives students the opportunity to print out whatever they require at this point. Even here, Help is available to tell students where to take the course approval form.

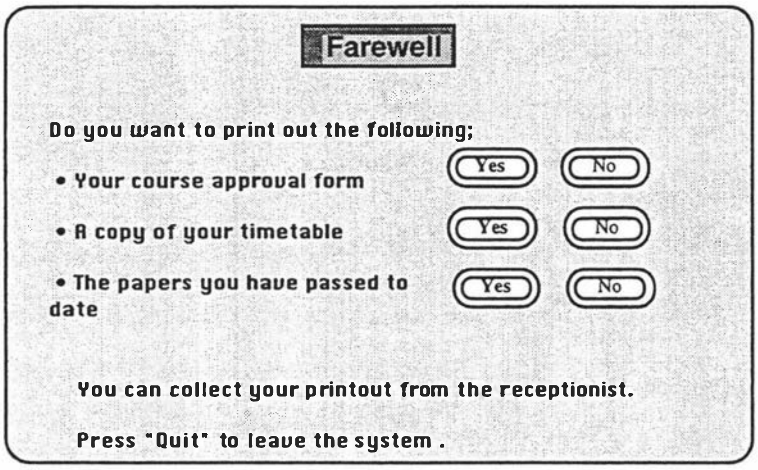


Figure 19 The "Farewell" screen