SES: A Soils Expert System.

A Thesis presented
in fulfilment of the requirements
for the degree of
Master of Science
in
Computer Science

by

E. G. Todd

July 1988 Massey University

Abstract

An Expert System Development Methodology is proposed, based on experimentation in developing a soils expert system (SES) which identifies a soil from incomplete field data. Tools for conceptual modeling of the soils domain are examined. The tools developed provide a means of recording the conceptual model of the knowledge from three different view points: the inference structures, the domain objects and the functional aspects. A review of the structures used in the knowledge bases of existing classification problems identifies eleven categories for grouping these structures. Using this information with the conceptual model, a detailed design of the knowledge base for SES is created. This design closely models those structures identified as important in the soils domain ensuring that important knowledge is represented explicitly.

Acknowledgements

This thesis is the result of a research program in which the author is indebted to the labours of many people and organizations. I would like to express my appreciation to Mr Ray. Kemp, my Supervisor, for his guidance and motivation throughout the time of this research. I wish to especially thank my husband, Arthur, for continually supporting and encouraging me to complete this piece of research.

I would also like to thank :

Dr Alan Palmer (Pedologist, Department of Soil Science, Massey University) who acted as the main domain expert and also Dr Bob Stewart (Geologist, Department of Soil Science, Massey University) and Mr Hugh Wilde (Pedologist, DSIR Soil Bureau)who both made contributions to the knowledge base of SES.

Members of the Staff of the Computer Science Department, who have given advice, assistance.and encouragement.

Table of Contents

List of Figures

Chapter 1: Introduction

During the nineteen sixties and early nineteen seventies Artificial Intelligence research concentrated on the development of general purpose problem solving strategies. The best known example is the General Problem Solver (GPS) built by Newell, Shaw and Simon (1963). It attempted to replicate the kind of problem solving humans use every day. From the nineteen seventies the direction of this research has altered. This change was initiated by the publication of research that concentrated on the development of programs for solving particular problems in specialized areas of domain knowledge. Such programs are currently referred to as expert systems.

Specifically, the research related to the MYCIN expert system (Shortliffe, 1976), triggered interest in the application of the results of Artificial Intelligence research to the wider software development arena. In the early 1980's the products of the Artificial Intelligence research centres have been taken up by innovative development units in both universities and industry. The main direction of this research has been towards the development of effective expert systems.

Medical domains are the most widely represented in the expert system literature. MYCIN is the best documented of these systems, and contains knowledge about the family of meningitis diseases. It has been extensively tested and modified. Research following on from MYCIN has led to a whole family of expert systems and expert system shells (Clancey,1986). EMYCIN (Empty MYCIN), is one of the earliest expert system shells. EMYCIN was built by removing the knowledge from the knowledge base of MYCIN. PUFF (Kunz et al, 1978) was built using EMYCIN by adding a knowledge base about pulmonary physiology. CENTAUR (Aikins,1983) was built by taking the knowledge in PUFF and redesigning the knowledge structures to represent the same knowledge using a mixture of frames and rules

Expert system technology has also been used successfully by the computing industry itself. The most notable example is R1 (McDermott, 1980), which helps to configure Digital Equipment Corporation's VAX series of computers. DIGITAL has claimed that this program has allowed it to gain a significant advantage over competitors. Other computer companies have followed this lead. PRIME has developed an expert system DOC (Littleford,1985) which analyses a memory image from a crashed Prime computer system. The expert system deduces the cause of the crash and recommends a course of action for repair. DOC only loads that part of the knowledge base that is relevant to the problem after ascertaining what model of CPU is involved and the significant peripheral devices.

The application of expert system technology is only just beginning to be widely reported. A literature search reveals that expert system technology is being applied in the space industry, in business management, the oil industry and by the military. An important application of expert system technology is in interpreting and applying complex codes of practice and specific sets of regulations. In New Zealand, BRANZ (Whitney, 1987) has successfully developed a system to help check that a building design complies with the fire regulations. Government Computing Services (Barton, 1987) has developed a system to help determine a client's unemployment benefit eligibility.

Much expert systems based research has been into knowledge representation. This has been developed in parallel with natural language processing. Both these lines of research are important for the development of large computerized data stores. Current data bases are limited to factual knowledge and lack the semantic and heuristic knowledge of an expert system knowledge base. Although special purpose query languages have been developed for accessing the data, often potential users are either unsure of exactly what they are searching for, or alternately, how to phrase the questions so the required information can be obtained. Natural language research has enabled the development of natural language interfaces to a number of database products. These help users access the information they require. Examples of a number of systems are outlined in Bonnet (1985).

The retrieval of computer based information and the use of application packages by non-computing personnel provides a diverse area for the application of expert systems technology in the commercial environment. Expert systems can be built for existing computerized knowledge sources and application packages. For these expert systems the domain of expertise is a combination of knowledge about the application and about how people typically wish to use the application. SACON (Bennett and Englmore, 1979) is a front end to an application which determines the resistance of different materials. SACON helps the user by giving advice on how to use the system for analysing structures.

Expert system technology is extending the limits to the type of knowledge that can be stored on the computer and the way in which this knowledge can be used. Expert system technology has expanded users expectations concerning the type of information they can request from a computerized system. In particular expert systems have knowledge of how they work and can therefore give some explanation of their actions and lines of reasoning.

1.1 Purpose of the Project

The purpose of this project is to investigate the suitability of using expert system technology in the field of soil science. The main aim is to evaluate relevant methodologies for analysis and design of expert systems. This study emphasizes the role of prototyping in expert systems development and the design of the knowledge base. Development of diagrammatic tools for developing and representing models of different aspects of the knowledge base form a significant part of this research.

The target application area is the identification of New Zealand Soils from incomplete field data. Not only should the system be able to identify a soil accurately from sufficient data but also it should be able to report when it is not possible to determine an unambiguous identification. At this point the system should offer help to the user by identifying the additional data that is required. It should highlight the important features that would either confirm or negate the most likely candidate soil types provisionally identified from the data so far.

1.2 The Soil Science Domain

The areas to which expert system technology have been applied are expanding rapidly. Systems that are used in specialized aspects of the earth sciences were one of the earliest application areas. PROSPECTOR, (Duda, Gaschnig and Hart, 1979) evaluates geological structures for the purpose of identifying and assessing the commercial viability of mineral deposits. The inputs to the system are the geological field data collected by geologists and the output is a site evaluation and maps of the deposit.

Other earth science associated expert systems include: DRILLING ADVISOR (Harmon and King, 1986), DIPMETER ADVISOR (Baker, 1984) and LITHO (Ganascia, 1984). DRILLING ADVISOR provides advice on solving problems encounted with drill bits when drilling exploration and production drill holes. The other two systems are used to interpret the data from down hole wire-line logging of drill holes, particularly in petroleum prospecting.

Massey University has specific expertise in pedology[1], one of the branches of the earth sciences. With the diversification of agriculture and horticulture in New Zealand the application of computer technology to the dissemination of information about soils is a timely project to tackle.

---

1 Pedology is the study of Soil Science.

The identification of a soil from field data has a number of parallels with the evaluation of geological field data. Both disciplines are basically concerned with describing three-dimensional layers including, their characteristics, boundary conditions and the processes involved in their formation. Both disciplines generalize the descriptions by producing classification systems; systems which permit specific instances of a layer or group of layers to be sorted, compared, correlated and contrasted.

Soils, like rock formations, are the product of the intersection of a number of closely interrelated processes. These processes are not discrete but progressively change over time and in space. Creating an hierarchical classification of soils is therefore inherently subjective. Classifying a specific soil also involves a degree of incidental association, as indicated by the following quote from Taylor and Pohlen (1970)

> " a full definition of a kind of soil includes a statement of both differentiating and accessory characteristics, of the permissible ranges in each, and of any likely accidental characteristics that may serve as phase distinctions. "

Before a soil can be classified the pedologist has to describe the soil. A soil description records both the soil forming factors and the soil morphology. Soil forming factors have been recorded in the site descriptor since Dokuchaiev (Neall, Palmer and Pollock, 1987) observed that

> " soils are products of extremely complex interactions of local climates, plants and animals, parent rocks, topography and the ages of landscapes"

The modern pedologist views the soil he can describe in the vertical section profile as a complete integrated, natural body that reflects the combined effects of the soil forming factors. From these direct observations pedologists have noted the associations between the site descriptors and the soil morphology. The association between the soil forming factors can be shown in a simplified soil-function equation

$$s = f(cl, o, r, p, t)$$

where:

- s = soil
- cl = climate
- o = organisms
- r = relief
- p = parent material
- t = time

This equation is a surface reflection of the processes that form a soil. Many of the soil forming processes are as yet poorly understood and the soil forming factors associated with a process are often uncertain.

For example a New Zealand podzol (Neall, Palmer and Pollock, 1987) forms under the following conditions:

- cl - adequate rainfall in humid and superhumid regions
- o - under kauri forest in Auckland and to a lesser extent under rimu or beech trees
- r - flat or rolling relief, not on steep sites
- p - coarse or medium textures parent material
- t - and enough time for the expression of the process on the soil morphology

The site description holds information about these soil forming factors and the profile description holds the information about the soil morphology. A profile description is a detailed inventory of the changes in the major soil characteristics, beginning at the ground surface and extending vertically down to the underlying rock material. Each soil is made up of layers that are termed horizons. For a specific soil at the site where the profile hole is dug characteristics are recorded and indexed by depth via the horizon designation. For each horizon characteristics such as colour, consistency, porosity, size and shape of aggregates and degree of compaction are recorded to build up a detailed description of the form and structure of the profile.

The classification system used for soil identification purposes is derived from the soil mapping units used on the soil maps accompanying the DSIR Soil Bureau Bulletins. This classification system is a simple hierarchy that forms a pyramid of units, figure 1.1. The soil type forms the smallest unit while the largest unit is the soil class.

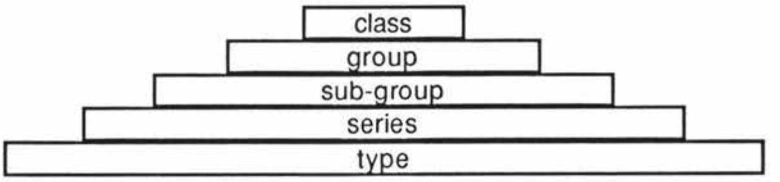


Figure 1.01 Hierarchy of Soil Classification Units.

The Soil Bureau bulletins are published on a County basis thereby forming spatially discrete units of knowledge. The information in these Soil Bureau bulletins and the associated maps is not easy to interpret correctly by either students of soil science or agriculture specialists. An expert system could complement this information by helping the user to interpret the field data they have collected. By identifying the most probable classification unit, the system could enabling the user to make more effective use of detailed soil properties and characteristics in the bulletins.

1.3 Organization of the Study.

Chapter two reviews expert system technology. The review summarizes how expert systems can be classified. The different components of an expert system, of which the knowledge base is the central component, are described. The chapter concludes with a discussion of current expert system development methodologies.

The five stage methodology proposed by Hayes-Roth et al (1983) was selected to guide the development of the Soils Expert System, SES. Chapter three discusses Identification, the first of these stages. An assessment of the feasibility of the proposed expert system. Guide-lines for determining whether a problem domain is suitable for an expert system solution are reviewed. The use of prototyping to determine the feasibility of an expert system is explored.

The conceptualization stage defines both the requirements for the system and the specification of the conceptual design of the knowledge base. The requirements of the users and the modules comprising the expert system are outlined in chapter four. Prototyping is used to clarify specific problem areas, for instance interface design.

Aspects of the specification of a conceptual model for an expert system are reviewed in chapter five. Three views of the domain knowledge are identified as important for describing the conceptual model of a knowledge base. These are used to develop the conceptual model of the knowledge base of SES.

The structural or low level design of an expert system corresponds to Hayes-Roth's formalization stage. A review of the knowledge base structures used in three existing expert systems forms the basis of chapter six. The common features of these knowledge base structures are identified.

Information gained from identifying the common structures in existing knowledge bases was used to redesign the prototype for SES. The evaluation of this prototyping exercise is described in chapter seven. The description of the types of knowledge base structures considered necessary for SES are based on the results of this evaluation.

The final chapter contains a summary of the main points discussed in the thesis with reference to the stated aims of the research. Additional applications for expert system technology in the soil science field are suggested. Extensions and improvements to the current project and associated topics are outlined as suggestions for further research.

Chapter 2: An Overview of Expert Systems

The types of programs termed "Expert Systems" range from simple decision trees to complex natural language processing systems, with some self learning capabilities. These expert system programs may be embedded within sophisticated pieces of computer controlled machinery, but most expert system programs interact with a definable human user group. The terms knowledge-based system and consultative system are both used as synonyms for expert systems. Knowledge-based system could be considered a generic term for expert systems while consultative system refers to a subset of expert systems that model the human expert in the role of a consultant.

This chapter describes the fundamental features of expert systems. The term "Expert System" is defined and the main components of these systems are identified. The three main ways of representing knowledge are reviewed followed by an introduction to expert system development methodologies.

2.1 What is an Expert System

Research efforts in Artificial Intelligence established that it was possible to capture and manipulate non-numeric and heuristic symbolic knowledge in association with known facts to create a viable computer reasoning system. Expert System technology applied this knowledge to specific domain areas. Partridge (1986) identifies the difference between traditional systems and expert systems in terms of differences in the way the problem is defined.

Traditional data processing systems model a "well-defined abstract problem" (Partridge, 1986). In these systems the flow of control within the programs is tightly specified and the data being manipulated are actual values, such as client's id numbers and transaction amounts. Data processing systems implemented on computers commonly automate manual systems that are used and understood by a relatively large number of people.

According to Partridge an expert system models "an inherently ill defined problem" . Expert systems manipulate data and knowledge in symbolic form. The execution sequence is controlled by the inferencing mechanism. Expert systems deal with specialized domains that are often understood by very few people. i.e. experts.

Human experts uses different reasoning methods than those applied in traditional data processing systems. Experts uses inductive, deductive and empirical reasoning methods; forming hypotheses, asking further questions to refine the quality of the data and finally interpreting the data, to come to the most likely conclusion. Experts can explain why particular inferences were made, which points were pertinent in the data and how the conclusion was reached. Furthermore, they can give directions as to what additional data is required so tentative conclusions can be refined. Experts build up a body of experimental knowledge that can be regarded as a set of heuristic rules. These are "rules-of-thumb" that allow an expert to interpret and form conclusions from incomplete data very quickly.

When an expert is consulted a dialogue is established between the participants. The expert asks questions, answers queries, points out important features, indicates additional data requirements and explains how conclusions were reached. An expert system is an attempt to build a restricted computer model of some aspects of an human expert. Therefore an expert system should be able to request data as required, explain its reasoning and justify its conclusions. Just as an human expert is only an expert in a particular field, an expert system has a narrow field of application. Expert systems are most often written for a group of users who though not experts in the domain they are none the less familiar with some aspects of the domain.

A formal definition of expert systems has been given by Brachman et al (1983):

> " An expert system is one that has expert rules and avoids blind search, reasons by manipulating symbols, grasps fundamental domain principles, and has complete weaker reasoning methods to fall back on when expert rules fail and to use in producing explanations. It deals with difficult problems in a complex domain, can take a problem description in lay terms and convert it to an internal representation appropriate for processing with its expert rules, and it can reason about its own knowledge (or lack thereof), especially to reconstruct inference paths rationale for explanation and self-justification. "

Brachman clarifies this definition by identifying the fundamental properties that an expert system should exhibit. The seven features identified are:

1. Expertise; The system simulates an expert in some domain oriented task either as a consultant or a colleague.

2. Symbol Manipulation; The system incorporates the knowledge about the domain by representing it in some symbolic form which it can then reason about.

3. General Problem-Solving Ability in a Domain; The system can solve problems within the domain of expertise.

4. Complexity and Difficulty; The system is concerned with problems that normally require the input of a human expert. Solving problems in the domain may involve: one or more tortuous search paths, numerous different tasks, or assessing a large number of decisions.

5. Reformulation; The system requires the real world information to be reconstructed so that it can be manipulated. The domain knowledge is analyzed and refined into some symbolic form. Data describing a specific problem is converted into an abstract representation that can be matched using heuristics to the abstract solutions within the knowledge base.

6. Abilities Requiring Reasoning about Self; The system should be able to explain what it is doing and why.

7. Type of Task; The system can be classed as carrying out one or more generic expert system tasks. Precisely what these generic tasks are could be disputed. Hayes-Roth (1983) identifies six generic classes of task while Clancey (1985) recognizes only two basic generic forms.

Weiss and Kulikowski (1984) use the following definition of an expert system:

> " An expert system is one that handles real-world complex problems requiring an expert's interpretation and solves these problems using a computer model of expert human reasoning, reaching the same conclusions that the human expert would reach if faced with a comparable problem. "

Although less concise this definition is more understandable and highlights two key points definitions of expert systems generally have in common:

1. There must be at least one human expert in the domain field.

2. The computer system attempts to model the expert's knowledge about the domain.

Scott et al (1977) point out that an expert system is not a psychological model of a human expert. More specifically in the context of a consultative production system, they identify the main goal of the modelling process as developing an expert system where:

> " the system and a human expert use the same (or similar) knowledge about the domain to arrive at the same answer to a given problem."

2.2 Structure of an Expert System

An expert system has two key facilities which form the core of the system (Davis, R, 1978; Buchanan and Duda, 1983; Hayes-Roth, 1985; Brakto, 1986; Harmon and King, 1986). These are:

- the knowledge base
- the inference engine

The type and complexity of the domain of application for a specific system determines the complexity of the key components, and the additional features the system may require. When a knowledge based system is being used as a consultant, or a colleague, two further components are essential to any system acting as an expert system:

- a communication facility
- an explanation facility

The communication facility is necessary as it provides an interface between the human users and the computer system they are trying to consult. For a consultative expert system the explanation facility is also basic requirement because the system is taking on the role of a consultant and interacting with the human clients. When a non expert client asks an human expert for advice the client needs to be able to evaluate the reliability of that advice. Generally the non expert can ask the expert how a particular conclusion was reached, or why some aspect is more significant than another. If a computer program is going to take on the role of an expert it, too, must be able to explain how and why it reached a particular conclusion.

A fifth facility is closely associated with the explanation facility. A knowledge acquisition facility is commonly included in an expert system, although it is not an essential component. This facility is used by the builder and/or maintainer of the knowledge base to add, delete and revise the static elements of the knowledge base.

To allow for incremental development of expert systems the different facilities are purposefully perceived as independent modules. Figure 2.1 shows how these different modules interact in a small scale expert system. The user, consults the system via the communications module which interacts with both the inference engine and the explanation subsystem. The inference engine is the reasoning centre of the system and determines how the system reaches a conclusion. Finally the knowledge base is where the expertise of the human expert is stored.

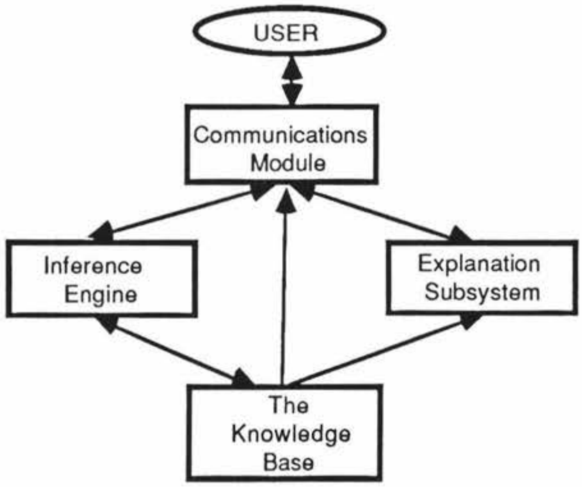


Figure 2.01 The four primary modules of an Consultation System showing the main communication channels.

2.2.1 The knowledge Base

The knowledge base is the module that gives expert systems their power. The type of knowledge stored in a knowledge base is dependent on the nature of the problem domain. In the simplest case the domain knowledge may comprise a finite number of solutions plus the information required to select an appropriate solution based on the data for a specific problem. Alternatively the stored knowledge may be that required to construct a tentative solution which is then evaluated for its relevance, correctness and completeness as a solution to a particular problem.

The knowledge base contains all the stored information required by the other modules so they can perform their functions. Depending on the complexity of the system, the knowledge base may need to be partitioned. Each module of an expert system would have access to knowledge that is specific to its needs as well as access to common areas of knowledge.

Further partitioning may be based on the type of knowledge in the knowledge base. Two main types of domain knowledge can be identified:

- Heuristic knowledge
- Factual knowledge

Heuristic knowledge is usually represented in the form of rules or default values. Factual knowledge can be stored in a standard relational data base format.

Another level of division of the knowledge base can be made between the knowledge that is static from one consultation to the next, and the dynamic knowledge which is created in response to a specific problem during a consultation. This division becomes blurred in a sophisticated system that involves self learning, because the knowledge in the nominally static area of the knowledge base will change as the system acquires new knowledge about the domain.

The division and identification of the knowledge found in an expert system has been discussed by numerous authors including Davis (1977), Davis (1978), Duda (1978), Clancey (1984), and Neches (1985). All or some of the following kinds of knowledge can be identified in current expert systems:

1. Surface Knowledge; heuristics based on experimental knowledge.
2. Deep Knowledge; underlying basic principles.
3. Compiled Knowledge; heuristics that can be explained in terms of basic principles.
4. Support Knowledge; information required for justification of decisions to the user.
5. Strategic Knowledge; strategies that control the reasoning process during a consultation.
6. Descriptive Knowledge; characteristics and definitions of the domain objects.
7. Dynamic Knowledge; values assigned during a specific consultation
8. Meta Knowledge; characteristics of the structure, composition and function of the domain knowledge.

When items in the domain knowledge have a degree of uncertainty associated with them then some method of indicating this uncertainty is required. The types of metrics used require the quality of the separate chunks of knowledge to be indicated by some form of weighting. This weighting indicates to the inferencing mechanism the reliability of each chunk of knowledge. The different metric methods used each provide a method for incrementally gathering evidence for and against a hypothesis. In some domains the use of a single value is insufficient to indicate the weighting both for or against the hypothesis currently being evaluated. In these cases the system requires extending to allow for more flexibility and accuracy by assigning two values to each independent chunk of knowledge. These values indicate, respectively, necessity and sufficiency.

2.2.2 The inference engine

The inference engine is the reasoning module of an expert system. It uses the data about a specific situation and the information in the knowledge base to develop and evaluate possible problem solutions. Unlike the processing carried out in traditional data processing systems, the inference engine of an expert system solves problems in domains where there are no clear algorithmic solutions. Instead, expert systems use inference methods that involve a substantial amount of plausible reasoning to enable them to come to a conclusion. Stefik et al (1983) identified three generic situations where plausible reasoning, or guessing, is used by expert systems.

1. Incomplete knowledge: When knowledge is incomplete it is necessary for the system to make an educated guess to fill in the gaps so that reasoning can continue.

2. Information Overload: When there is too much information and a large number of possible solutions , it becomes too inefficient to use exhaustive processing so guessing is necessary to simplify the search tree into more manageable proportions.

3. Heuristics: When the use of heuristics is the most effective way of arriving at a solution.

There are two main features of an inference engine:

- the control mechanism
- the inference mechanism

The control mechanism determines the order in which operations are performed. The inference mechanism provides the reasoning methods used. The analysis of the problem domain will probably indicate that one type of inference mechanism is more appropriate than another. It is possible that the best approach is to use a number of different inferencing mechanisms in an expert system to match the different stages in the problem solving cycle.

2.2.2.1 The control mechanism

The control mechanism, or scheduler, determines where each search or subsearch is to start, and resolves any conflicts when there are two or more equally valid lines of reasoning that can be pursued. In a rule based system this means determining the order in which the rules are to be traversed. In a structure orientated system the scheduler determines the order in which structures are invoked.

Conflict resolution is required to decide which rule or object to use when more than one becomes operable. The simplest strategy is to take the first operative item found and if it is successful continue with its line of reasoning. Alternatively each item can be listed in order of some priority strategy and the item on the top of the list is activated next.

The type of reasoning the system uses can affect the role of the controller. If the system uses monotonic reasoning then once an assumption has been made it can not be retracted. If nonmonotonic reasoning is used then the controller has to decide whether to replace an existing assertion by a contending one or to remove it altogether.

Three basic control methods are used in expert systems:

- backward chaining
- forward chaining
- the blackboard agenda

Backward chaining involves choosing a goal and then trying to prove it using the available data. Forward chaining is data-directed reasoning where the system loops through the knowledge base trying to limit the number of possible goals in a recognize-act cycle. The blackboard architecture is used when there is a very large knowledge base and a number of knowledge bases are required to solve the different parts of the problem. The system is divided into a number of parallel modules which cooperate via a common data structure called a blackboard.

2.2.2.2 The inference mechanism

The inference mechanism chosen for a specific expert system is based on characteristics of the application knowledge. Expert systems tend to use a combination of several different inferencing methods. Commonly *modus ponens*, fuzzy logic and the certainty factor model are used together. If the knowledge has precise characteristics then a straight-forward "true" or "false" decision is usually sufficient to determine the value of an assertion. Alternatively, if the domain knowledge is inexact or uncertain and/or the data supplied by the user is incomplete then some method of dealing with this unreliable information must be incorporated into the inferencing module.

Modus ponens is the simplest and most widely used form of inference. *Modus ponens* simply means that if the premise of a rule are considered to be true it is reasonable to infer that the conclusions of the rule are also true.

Uncertainty has to be quantified in some way so that the system can make judgements about differing degrees of truth. The two most widely reported methods (Buchanan and Duda, 1983; Forsyth, 1984; Harmon and King, 1986) of dealing with uncertainty are:

- Bayesian Logic
- The Certainty Factor Model

Bayesian logic requires a large database of facts about the domain that can be used to calculate conditional probabilities. If this information is not available, the domain expert is asked to supply the most likely values indicating the *a priori* probability for each chunk of knowledge.

The certainty factor model is a model of inexact reasoning that was developed by Shortliffe (Buchanan and Shortliffe, 1984) specifically to provide a workable alternative to the Bayesian Model. The main emphasis for developing the certainty factor model was to create a model that was understandable to domain experts collaborating in building the medical diagnostic system MYCIN.

When the certainty factor model is implemented, each rule is assigned a numeric value which indicates to the inferencing mechanism the weighting the expert gave to the truth of the consequence of a rule if the antecedent is true. The evidence from a number of rules all asserting the truth of the same consequence is accumulated, as is the evidence against the consequence. The accumulated degrees of belief and disbelief are then combined to determine the value of certainty for the hypothesis.

When the inference mechanism uses *modus ponens*, and a numeric indicator of uncertainty, a mechanism has to be found to quantify the truth of the antecedent when there is more than one premise. Most systems use possibility theory, referred to as fuzzy logic. Fuzzy logic tackles the problem of imprecision by extending boolean logic to include real numbers. This permits quantification of such descriptors as MOST and USUALLY indicating set membership in instances where there is no sharp boundary between membership and non-membership. It is important to determine exactly which population is being referred to when indicating set membership as membership is determined on a sliding scale with cut off points that may be purely subjective.

The fuzzy logic model also redefines the boolean operators AND, OR and NOT and most of the probability functions. Those most relevant to Expert systems are:

- p1 AND p2 = MIN(p1,p2)
- p1 OR p2 = MAX(p1,p2)
- NOT p1 = 1 - p1

MIN and MAX are used for combining pieces of evidence that are not unconditionally independent while NOT can be used for negative evidence.

2.2.3 The user interface

The user interface, or communications module, is fundamental to any program that interacts with the real world. In a consultation system the interface is between a human user and the system. Some expert systems have real world interfaces that are machine to machine as in the case of a control system gathering data from separate environmental sensing systems.

Communication with human beings is a complicated process. To be effective, it is desirable for an expert system to be able to communicate via a natural language. Current systems attempting to use natural language rely heavily on the keyboard skills of the user. Most users have poorly developed key board skills and therefore natural language interface modules will not become a cost effective communications method for expert systems, until reliable and usable speech synthesizers and voice recognizers have been developed. More powerful natural language processors are also required.

User friendly WIMPS based interfaces appear to be the most practical interfaces for present day expert systems (Kemp and Boorman, 1987). Whatever philosophy a developer may have about user interfaces, an expert system requires one through which the user, normally a human being, and the program can establish an effective man-machine dialogue.

2.2.4 The explanation module

The explanation facility provides feed-back to the user. Explanation is the method by which the expert system describes its chain of reasoning and the evidence that supports any solutions or conclusions. The two basic functions (Scott et al, 1984) of an explanation subsystem are to provide information on the status of:

1. The reasoning process

2. The knowledge base

The basic terms for explaining reasoning are HOW and WHY. HOW explains the decision path by which any particular conclusion, either goal or subgoal, has been reached. WHY explains the reason for the system asking the user a specific question.

Explanation involves describing the chain of reasoning the system is pursuing to prove a specific goal.

As well as the HOW and WHY facilities an explanation facility commonly includes features that enable the user to browse through the contents of the knowledge base. The browsing facility usually provide access to a glossary of technical terms defining the meaning of words and phrases used by the system in describing the domain knowledge.

The help facility is another aspect of an explanation module and is linked to the browsing facility. It provides the the user with explanations on different aspects of the domain knowledge and descriptions on how the system works i.e. knowledge about the systems functions and structures. The user can activate the help facility when unsure about how to respond to a system prompt.

The most basic explanation system simply displays an ordered trace back of the rules or the executed objects from a simple stack on which the session has been stored. An advance on this is to display a free-format text description of the knowledge from a pre-defined version supplied by the knowledge engineer or the expert. More sophisticated systems construct the explanation from the basic knowledge representation and tailor the explanation to a model of the user which the expert system builds up during the consultation. This type of explanation module (Rich, 1986) becomes an expert system in its own right.

When the system is to be used for training, or is to be used by people unfamiliar with the domain, there is a need for what has been termed deep knowledge. This is knowledge about the basic principles (laws, theorems and processes) on which the domain knowledge is based (Clancey, 1979). To be able to provide such information the knowledge base has to be augmented with the additional knowledge and more complex data structures than that which is necessary to reach satisfactory solutions within the domain.

2.2.5 Knowledge Acquisition Facility

The knowledge acquisition module provides the tools that enable the transfer of domain knowledge from an human expert source into the knowledge base. This module should ideally perform any transformations required so that the system can use this task specific knowledge. Buchanan and Shortliffe (1984) identify the role of acquisition as:

> " initially one of helping the expert conceptualize and structure the domain knowledge for use in problem solving"

The process of extracting expert knowledge is still poorly understood and consequently ill-defined. There are as yet no formal procedures or techniques available to help the knowledge engineer in the initial stages (Cooke and McDonald, 1986). The facilities that are available require that the basic concepts within the domain have been identified, a conceptual model established (albeit incomplete and informal) and an implementation representation chosen.

When the knowledge representation is stored as a text file the most basic type of acquisition tool is a text editor the knowledge engineer can use to edit the file. The simplest task specific facility is an editor that displays on-line a template of the basic knowledge structure which can be filled in as a form. Ultimately, the knowledge acquisition facility should be usable by the domain experts to build a domain specific knowledge base with a minimum of help (Boose, 1986; Shaw, 1984). Pragmatic researchers are investigating tools to assist the expert and knowledge engineer partnership in building the knowledge base. Tools identified as being required by the knowledge engineer to build and maintain the knowledge base include:

1. A human understandable version of the knowledge representation consisting of:
    - a functional description language (i.e. rules)
    - a data definition language (i.e. objects)
2. A context sensitive editor
3. A knowledge representation checker
4. An explanation facility
5. A debugging facility
6. A facility for creating and maintaining a library of test cases.
7. A batching facility for testing the knowledge base after each revision.

2.3 Knowledge Representation

Most writers on the subject of expert systems agree that the success of such systems lies in the content and design of the knowledge base. The knowledge base is the most important part of an expert system which is indicated in the statement "Knowledge is Power" (Hayes-Roth et al, 1983). Knowledge is more than the entity, attribute and relationships stored in a ordinary data base. Knowledge has been defined by Hayes-Roth as:

> " consisting of descriptions, relationships and procedures in some domain of interest"

From this description the three main kinds of domain specific knowledge required by an expert system can be identified:

1. The symbolic descriptors. These are the relationships that describe taxonomic, definitional and empirical associations between items within the domain.

2. The procedures for manipulating these descriptions. These are generally in the form of rules.

3. Item identification. These are the objects which can be identified within the domain and include the descriptions of entities, their attributes and the generic groupings of the items.

In addition to the knowledge about the domain, a knowledge representation should be able to provide facilities for representing meta level knowledge. Meta knowledge is knowledge that describes how the domain knowledge is represented and used. Davis and Buchanan (1977) have identified four major groupings of meta knowledge relevant to expert systems. These contain knowledge about the representation of:

- objects
- functions
- inference rules
- reasoning strategies

The content of the knowledge base has to be represented in some computable form but this form must be easily translatable to make it accessible to the users of the system. Buchanan and Duda (1983) identify three basic requirements for the knowledge representation used for an expert system:

1. Extendability - so the system can be easily modified in the light of experience and/or new knowledge.

2. Simplicity - the representation should be easily understood by both the users and the computing processes.

3. Explicitness - the knowledge used by the system should be easy to inspect.

A fourth requirement (Todd, 1987) is that the knowledge representation used should be able to be mapped onto the conceptual models within the domain. This additional requirement is supported in the list of criteria for deciding on a knowledge representation outlined by Clancey (1985) and Ramsey, et al (1986). Clancey identifies the need to examine the naturally occurring knowledge structures in the domain of expertise, viewing them as systems occurring in the real world. The knowledge representation used in an expert system is a model of real-world systems, and their inter-relationships.

Implementation representations can be divided into three generic groups (Jackson, 1986; Friedland, 1985). These are :

- production rule systems
- structured objects
- logic systems

Although each of these representations is described separately, elements of all three can be found in maturing expert systems. All rule based expert systems require a number of data structures in which to store information associated with the rules. Many of these data structures, for example those found in MYCIN, are basically object descriptions. In discussing the different ways of storing knowledge in the computer Jackson (1986) points out that regardless of the representation used

> " non-determinism . . . inconsistent and semantically anomalous descriptions of the world can still be composed."

It follows that any interaction with the knowledge base reflects the structures and content of this artificial model of the domain, which in the worst case may not actually have any meaning in the real world.

2.3.1 Rules

A production system uses rules to store independent modules of knowledge. The form of a rule is:

IF antecedent THEN consequence

The production rule is the knowledge representation used in most introductory text books and general papers about expert systems. Along with being the most widely documented representation, the production rule has the attraction that it is perceived to be an easily understood way of expressing knowledge. Hayes-Roth (1985) states that:

" Experts tend to express most of their problem-solving techniques in terms of a set of situation-action rules, and this suggests that Rule Based Systems should be the method of choice for building knowledge intensive expert systems. "

An alternative form of the rule is

IF situation THEN action

In both forms, the premises of the IF part are tested to determine their truth. But in the former, the rule is purely descriptive, meaning that it can be used either in a goal or a data directed mode. The antecedent being true implies that the consequence is also true. In the latter, when the situation is true then the action part is to be carried out. In this case the rule is an instruction being applied to achieve some procedural objective in a similar way to "if" statements in procedural languages.

Rules are normally understood as storing the heuristic knowledge associated with the domain, but they have been classified (Davis & Buchanan, 1977; Davis, 1978; Clancey, 1983 & 1985) using a variety of criteria. The two main types of rule are:

- Object level rules
- Meta rules

Object level rules are the rules that store the domain specific knowledge and can be further subdivided into two main groups:

1. Definitional rules; Three types of definitional rules are:
    a. identification rules, which are based on the properties of an object the system is interested in
    b. world fact rules, which are common sense rules that are relevant to the domain
    c. domain fact rules, which relate domain features within the problem area

    Clancey (1985) includes soft definitional rules within this group.

2. Heuristic rules; At least four types have been recognized as being relevant (Clancey, 1985) for matching data systems to solution systems:
    a. agent or experiencer rules.
    b. causal rules which range from rules that could be termed empirical to those termed complex.
        - empirical rules recognize a correlation or co-presence between features.
        - complex rules describe processes where at least the direction of the association is understood but not the whole model.

c. model rules that describe well understood processes.

d. preference rules that describe either known preferences or advantages.

Meta level rules store knowledge about knowledge. The divisions recognized for meta rules are:

1. Typical rules. These are rules that provide an abstract description of a subset of rules with the same typical structure. This allows for more compact storage of the knowledge and because it is a generalization of the knowledge base leads to better comprehension by users.

2. Control rules. The inference engine provides the global control for the expert system, that is in a sense a general set of primary control rules but meta rules can provide a second level of control or local control that is domain dependent. Control rules are one of the following types:

   a. strategic rules that:
      - make conclusions about other rules
      - give information on the best route when alternative choices are considered

   b. self referencing rules. A self referencing rule checks on the state of a specific feature before trying to prove a condition related to that state.

The main advantages of using rules as a method of knowledge representation are:

1. Compared to alternative representations rules appear to be easily understood by both domain experts and system users.

2. The structure of rules is particularly suited to symbolic expressions.

3. Each rule is an independent unit of knowledge.

4. Rules communicate via the active database. The data determines the sequence in which the rules are applied.

5. The modularity of each rule facilitates incremental development.

6. Simple explanation is easily generated by replaying the rules utilized so far.

2.3.2 Objects

A structured object is defined by Jackson (1986) as

> " any representational scheme whose fundamental building blocks are analogical to the nodes and arcs of graph theory or slots and fillers of record structures . . . and are essentially ways of grouping information in a more of less 'natural' way that allows it to be applied for a particular purpose."

Computer literature on data structures abounds with different kinds of structural objects including:

- trees
- networks
- graphs
- semantic nets
- frames
- abstract data objects

As yet the exact meanings of a number of these terms have not been standardized. The term semantic net can simply be a graph on which both the nodes and the arcs are labeled or can be a much more sophisticated model involving different types of links, organizations and inheritance. Tsichritzis and Lochovsky (1982) identify four aspects of a generalized semantic net model:

1. Generalization - determines class membership of objects. A member inherits attributes from its superclasses.

2. Aggregation - identifies parts or attributes of objects.

3. Classification - relates instances of objects to types.

4. Partitioning - groups objects on basis of the value of specified attributes.

The structured object that is associated with expert systems is the frame. A frame (Minsky, 1974) is a chunk of information that describes both the features and functions of some object. A frame system tries to build up a pattern of the problem situation and match it to those known by the system. As pointed out by Kemp (1986)

> " a frame or frame system can be matched with a pattern with only part of the information known, and the remainder surmised or computed as required"

A frame describes a complex structural node in a network and provides for inheritance in one direction via the linking arcs. Inheritance is a weak form of inference allowing a frame to acquire information from a more general parent in the hierarchy. Inheritance can be used to provide default values for attributes which can be overridden by values in specific frames. Procedural knowledge can be associated with the data defined in each of the slots in a frame. This procedural knowledge is often in the form of rules in frame based expert systems . The rules provide information for data collection and validation as well as strategic knowledge about matching frames.

Object orientated programming is a further refinement of frames. The main difference is that in an object orientated environment there is a distinction between the instances of an object and the generalized hierarchy of classes that define its type. Also in an object orientated environment prototypical objects can be set up that become templates for instances of the object.

2.3.3 Logic

In a logic system there is no differentiation between the facts and the rules, all knowledge commonly being stored as Horn clauses. The clauses define the concepts identified in the domain and the states that can occur. A logic program is a declaration statement of the problem, which can often be transformed into a runable program specification. The best known logic programming language is Prolog, which is based on predicate calculus. Logic programming can also be viewed as a generalized form of a rule based system.

The most important feature of logic programs is that they use "an application-independent inference procedure" (Genesereth & Ginsberg, 1985). The programmer only has to develop the knowledge base. Although, it is possible to write meta logic statements to tailor a system for a specific problem.

The main limitations to using logic based programming for expert system development is that the inferencing systems are unable to draw conclusions if the data is uncertain unless they are modified or some form of data laundering takes place. Secondly, the interfaces are as yet unsuitable for ordinary application users. For a logic program to succeed the knowledge base must be correct as the power of logic cannot compensate for an incomplete model.

Logics, such as first order logic, can also be used as a specification language for providing a logical description of the domain. Many formal specification languages are based on one of the formal mathematical logics. This use of logic is gaining acceptance for specifying critical areas of specialized software because it is possible to prove mathematically the correctness of a specification. As with the logic programming discussed above, using a first order logic to completely describe a knowledge base is probably impractical due to the inherent uncertainty associated with problem domains suited to expert system solutions.

2.4 Developing an Expert System

Typically an Expert System is developed by building a series of prototypes. A prototype, according to the Oxford Dictionary of Current English, is

> " a trial model, or preliminary version".

This definition implies that the prototype is effectively an experimental and expendable item. A prototype in computing terms may be a throw away program but alternatively it may evolve to become the final installation system. The meaning of the term prototype when used in reference to expert systems development depends on the current state of development of the project. In general the final prototyping exercise results in the production system.

The process of prototype development is referred to as prototyping. The use of prototyping for developing commercial systems has required the adaption of traditional data processing system development methodologies and development of new strategies that enable project managers to make decisions about system deliverability.

2.4.1 Prototyping

There is no one universal definition of prototyping. The cynical may prefer Hayward's (1986) definition

> " 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"

A more useful definition is that of Huffaker (1986) who defines prototyping as:

> " A technique of discovery and a tool for effective communication"

Prototyping is often referred to as experimentation and it can be used very effectively in this mode. Huffaker discusses two ways in which prototyping can be used. They are:

1. Expendable prototyping which is akin to the experimental mode where a program is written to explore a specific aspect of the finished system. It is disposable and may not be written in the language of the final system.

2. Evolutionary prototyping which as the name suggests is the development of a system that will evolve by being modified and extended after being evaluated by the users. When the users are satisfied with the system, the prototype is declared to be the final system.

Although prototyping has been widely used effectively in the development of conventional data processing systems, such software systems normally have well defined sets of rules and practices. These can be identified and taken account of to develop rigorously defined requirements and specifications. The traditional method of developing such a system is by working through the steps of feasibility, requirements, specification, design, coding, testing and maintenance. This methodology is inappropriate where uncertainty surrounds the specification of the problem domain or when it is uncertain that a system can be developed to address the problems within the domain.

Prototyping is one method that has been used successfully where the development of a complete well defined specification was initially untenable. In the case of domains suitable for expert systems development a specification as defined in terms of the traditional life cycle may actually be impractical even after the system is in production. Uncertainty implies an inability to develop a rigorous set of specifications. This can occur because:

1. the domain has not been computerized before and there is no experience on which to base the technical aspects of the design

2. the intended users are not sure as to what they really require.

3. due to context-sensitive components of the domain or system.

All these conditions normally apply to the development of expert systems.

2.4.2 Stages in Prototype Development

Hekmatpour and Ince (1986) discuss prototyping in a commercial data processing environment and outline a four stage prototyping development methodology. These are:

1. Establishing Objectives

2. Function Selection

3. Prototype Construction

4. Evaluation

The most important phase of prototyping is its evaluation with respect to the established objectives. It is important to clearly define these objectives when building a prototype so that when the objectives have been met no further development is carried out before a full evaluation is made. If the cut off point can not be identified then the purpose of the prototype maybe lost.

Function selection is closely related to the establishment of objectives. The functions identified as requiring implementation in the prototype are those that are necessary for the objectives to be met, but selected functions will to some extent determine the objectives of the prototype. Selection of functions will include deciding on which functions need to be simplified and alternatively which functions need to be developed in full.

The main aim during the construction of the prototype is to develop a system that can be used experimentally as quickly as possible. Finally the system is evaluated against the stated objectives and in terms of what has been learned.

These stages may be used either for developing an evolutionary system or for expendable prototyping where the prototyping activity is embedded within some of the steps of a traditional development methodology. In particular prototyping can be used effectively as a tool to determine requirements and to aid in developing a set of rigorously defined specifications. Within this scenario the 'throw-away' prototype may be built to explore the implications of

1. a critical module within a system, with the purpose of identifying problem areas that require strict specifications.

2. a central module, to determine interface requirements with other modules.

3. the interface for a specific user group, so the users and developer can determine system requirements and specifications cooperatively.

4. a limited model of the complete system, using a high level language which becomes in effect an executable specification that is then implemented in a more efficient language.

5. the functional aspects of the system, with minimal interface facilities so that functional specification can be more precise.

6. several systems or modules for the same function, using different data structures etc. to determine which is the most appropriate for meeting system requirements.

2.4.3 Prototyping and Expert System Development

Prototyping is widely accepted as the standard methodology for developing expert systems because the nature of expertise and knowledge is not clearly defined and is poorly understood. Experience in building expert systems is sparse. As well as not being able to define the system clearly at the start, implementing an expert system makes use of symbolic programming which requires different skills to those used in conventional data processing environments. These factors all contribute to the evidence that prototyping is a suitable methodology to apply to expert system development.

One of the most authoritative books on the building of Expert Systems (Hayes-Roth et al, 1983) discusses the use of prototyping in the context of a cyclical set of stages:

- Identification
- Conceptualization
- Formalization
- Implementation
- Testing

Although these authors initially describe these steps as leading one into another they acknowledge that the steps are not well-defined or even independent. They discuss the development of a prototype expert system in terms that imply that it is generally an evolutionary process.

The steps proposed by Hayes-Roth for the development of an expert system can be loosely related to the traditional life cycle, figure 2.1. In the traditional life cycle each step has been rigorously defined whereas the definitions of the steps for developing an expert system could more appropriate be described as heuristic.

Expert Systems Life Cycle		Traditional Life Cycle
Identification	------>	Feasibility
Conceptualization	------>	Specification
Formalization	------>	Design
Implementation	------>	Coding
Testing	------>	Testing
		Maintenance

Figure 2.02 Comparison of Traditional Life Cycle with Hayes-Roth's Expert System Development Cycle.

The stages for prototype development proposed by Hekmatpour and Ince (1986) are equally applicable to the development of expert system prototypes within the stages outlined above. At each stage the different aspects of the system can be explored using the four stages of the prototyping cycle. When using evolutionary prototyping the four stages outlined are used for developing each new facility as they are added to the main prototype. In this situation objectives of the main system are extended and modified as each major facility is incorporated into the prototype.

When developing an expert system the first stage of the prototyping cycle is the building of a prototype to determine whether the proposed system is feasible. The evaluation of this first prototype should result in the identification of the problem to be solved, the scope of the problem domain, the constraints on the systems development, as well as the choice of knowledge representation and data structures required. If these initial decisions prove to be sufficient for the final system, then the prototyping process used to develop an expert system could be classified as evolutionary. If assessment of the prototype at any stage indicates that the initial decisions will not lead to a successful final system then the prototype should be discarded and a cycle of expendable prototyping adopted until a satisfactory representation of the domain is created.

Expendable prototyping should also be used to experiment with different aspects of the system before being incorporated into the main prototype. A prototype might be developed to explore different ways of presenting information to the user, or to try out different inferencing methods. Regardless of whether the prototype is for the main system or for exploring just one aspect of the system when any prototype is built a clear set of objectives should first be defined and these should be used.to determine the cut off point in the development of that prototype.

2.5 Summary

In the literature reviewed there is no clear definition of an expert system. Brachman's check list of the features an expert system should exhibit, is the most comprehensive of the definitions cited. Basically, consultative expert systems attempt to replicate the role of a human expert. Currently successful expert systems can only carry out such a role consistently within the narrow application domain for which they have reliable knowledge.

Five modules have been identified as important components of a consultative expert system. The inference engine, the knowledge base, the communications module and the explanation facility are all considered to be of primary importance for a consultative expert system. The acquisition module is a desirable extra. Of these five modules the knowledge-base is the most important component. The design of the knowledge base and the representation used for the knowledge is crucial to the development and operation of the other modules in the expert system.

In the penultimate section, the importance of prototyping and the role of prototyping techniques for designing and implementing an expert system were examined. Methodologies for controlling the use of prototyping have been reviewed with specific reference to the expert system development methodology proposed by Hayes-Roth (1983). Finally this method was compared to the traditional life cycle for systems development.

Chapter 3: The Feasibility of SES

The identification stage in the development of an expert system is primarily concerned with finding out if there is a feasible solution to the problem. Secondly it should determine whether the solution is best implemented using expert system technology. The type of problem where expert system technology is applicable is one that appears to be difficult to solve using conventional software techniques and requires heuristic methods to make the problem more tractable. Buchanan et al (1983) suggest that if

> " the knowledge is firm, fixed and formalized, algorithmic computer programs that solve problems in the domain are more appropriate than heuristic ones. "

A reliable method for deciding on the applicability of an expert system solution has not yet been developed but many authors have published guide-lines and check lists.

Once the feasibility of the project has been determined the next major step is to decide how the problem can be subdivided into manageable partitions and the order in which these will be investigated. The decisions about subdivision and order are closely related to the type of prototyping method that will be used.

The identification stage includes a definition of the system requirements. At this stage the goals and objectives of the project are decided and project management decisions made. Decisions include staff organization, responsibility, lines of communication and plans on how the following stages in the development cycle are to be executed. Sources of information to complement that provided by the Expert should be identified.

The feasibility of a soils expert system is established by first analysing the overall problem and determining whether it meets the criteria suggested in the literature as appropriate to an expert system solution domain. To test whether an expert system solution is actually possible, two simple expert systems were implemented and the resulting prototypes are discussed. Finally the methodology for further development of the system is outlined, sources of knowledge indicated and the subdivision of the problem for further development is summarized.

3.1 Deciding on the Feasibility of a Project

The guide lines for deciding on the suitability of a problem for an expert system solution usually concentrate on the characteristics of the problem. Prerau (1985) has published guide-lines as a list of fifty-four rules on different aspects of the problem domain. These rules when applied to a specific domain should indicate whether it is possible to capture the knowledge in an expert system and whether development of such a system would be commercially viable. Prerau divides the desired properties of the domain into six main groupings.

1. Basic Requirements - concentrating on project desirability versus risk of failure.

2. Class of problem.

3. The Expert - in particular the degree of commitment the expert is able to give the project.

4. Problem bounds - explores size and complexity .

5. Domain area personnel.

6. Other desirable features.

Hayes-Roth et al (1983) have tried to categorize expert systems into different types by identifying the sort of problems that can be solved using this technology. The authors identify ten types of expert systems, shown in figure 3.01. If a specific domain falls into one of these categories then it is possible that an expert system can be successfully implemented.

Category	Problem Addressed
Interpretation	Inferring situation descriptions from sensor data
Prediction	Inferring likely consequences of given situations
Diagnosis	Inferring system malfunctions from observables
Design	Configuring objects under constraints
Planning	Designing actions
Monitoring	Comparing observables to plan vulnerabilities
Debugging	Prescribing remedies for malfunctions
Repair	Executing a plan to administer a prescribed remedy
Instruction	Diagnosing, debugging, repairing, and monitoring system behaviours

Figure 3.01 Generic categories of knowledge engineering applications.
(from Hayes-Roth et al, 1983)

Stefik et al (1983) have identified eleven different classes of problem that can be solved using different aspects of expert system methods The requirements of the simplest class of problem are

- small solution space
- data reliable and fixed
- reliable knowledge

The type of solution they prescribe for such a problem area is one where a single line of reasoning is pursued using monotonic logic and exhaustive search techniques. By relaxing each of these requirements in turn the ten other classes are characterized.

The classification system proposed by Clancey (1985) divides expert systems on the basis of the method used for solving the problem. The domain is defined in terms of systems. Two different problem solving methods have been identified- interpretation and construction. These methods can be further subdivided into a hierarchy of generic groups, figure 3 02.

The developer first describes the problem in terms of a sequence of operations relating the identified systems. From this study a set of significant points arise that should then be considered:

1. Is the problem concerned with perception ?

2. Is the primary task numerical analysis?

3. Can the system be practically enumerated ?

4. Is there a hierarchical description for generating solutions ?

5. Is it possible to use incremental search ?

6. How many uncertain choices need to be made ?

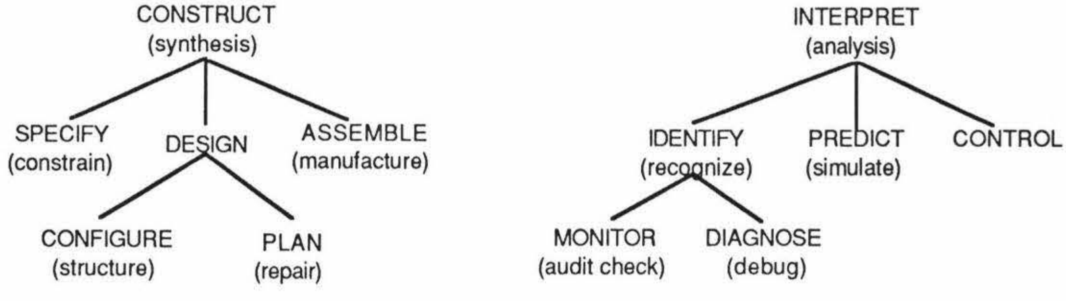


Figure 3.02 Generic operations for synthesizing and analyzing a system. (Clancey, 1985)

The features of a problem area that have been identified are matched with the different expert system problem solving methods. In complicated domains more than one method may have to be used at different stages during the consultation process.

Although the guide-lines discussed will identify some domains as unsuitable, eventually the only way to determine whether the technology can be applied to a domain of expertise is to try and implement a subset of the problem. This results in the implementation of a prototype where the main objective is to ascertain whether it is feasible to build an expert system that is useful in this problem area.

3.1.1 A Soils Identification Expert System

The first stage in developing any system, as outlined above, is to determine whether the project is feasible. This requires a statement of the main intent of the project. The main objective of the soils identification expert systems project is:

> " to develop an expert system to identify a soil from incomplete field data "

The practical purpose of the system is to assist people with limited training in soil science to identify soils in a specific region of New Zealand. Two kinds of user are anticipated: undergraduate soil science students and agricultural advisers.

The knowledge base will contain information about soil site descriptions, soil type location profiles and associated heuristic rules. The inputs will be incomplete field description data including soil forming factors such as topography, climate, soil parent material and profile information. The output will be the soils classification at either Series or Type level. Information about agriculture/horticulture potential of such soils could be.associated with the classification.

The system was recognized to be primarily one of classification and this type of problem has been shown to be suitable for expert system development (Hayes-Roth et al, 1983; Weiss and Kulikowski, 1984). Using Hayes-Roth's categorization the soil problem is to interpret the field data and infer which classification class the soil best falls into. The problem falls into Stefik's (1983) second class of problem as the data can be unreliable and the limits on the individual chunks of knowledge may overlap. This category of problem suggests that a possible prescription for a solution could include all or some of the following approaches

- combining evidence from multiple sources
- using a probability model either Bayesian logic or the certainty factor model
- incorporating the fuzzy set model into the inferencing process.
- using an exact model such as predicate calculus with some form of error correction for cleaning up the data

Although the soils domain appeared to meet those conditions outlined as necessary for expert system development, to confirm that the development of SES was feasible a prototype was built. This step was carried out before continuing with any in-depth analysis of the problem domain.

Prior to coding the prototype, a choice of sub-problem had to be made and the appropriate limits defined under the following headings: goals, domain, representation, language choice and interface design.

3.2 Development of the Initial Prototype

The initial prototype was built with two main aims:

1. To answer the question "Was an expert system practical for the soils domain ?" .

2. To produce an expert system that could be demonstrated to the domain experts so they could decide on their level of participation in the project.

Specific objectives for the design of the first prototype were:

- to keep the code as modular as possible.
- to make the individual modules as readable as possible.
- to create a knowledge representation structure that would be relatively easy to extend.
- to make the knowledge in the knowledge base easily understandable by a domain expert.

This first prototype was built with no direct input from a human domain expert. Therefore it was necessary to choose a subdomain that was simple enough to implement from published sources. The source of expertise used was Gibbs (1980).

It was decided to limit the implementation to a system for determining whether a soil was either a brown grey earth or a yellow grey earth. Both these soils are Zonal soils the simplest class of generic soils in New Zealand. Both soil groups have a set of well documented features that make identification comparatively straight forward. This sub problem not only limited the domain to two soil descriptions but it was also limited to the top two levels in the New Zealand soil classification hierarchy. Another limitation was to assume that the field data was accurate when known so that questions on facts could be answered by a simple "yes" or "no".

Finally the number of site description and profile description features allowed was limited. The choice of features used to describe the soils was determined by those that could be considered diagnostic and those that are listed as the main criteria in Soil Survey Method (Taylor and Pohlen, 1970).

The production rule has been considered the natural form of knowledge representation (Hayes-Roth, 1985) and was chosen as the basic structure for building the knowledge base in the first prototype. The rule is a modular easily understood representation that in simple form is relatively easy to implement. The language used was Prolog. In general it is easier for a non computing person to comprehend Prolog when compared to Lisp, a factor which supported the objective that the knowledge base be easily understood by a domain expert.

The code for this first prototype was developed in 3 modules: the inference engine, the interface and the knowledge base. A simple inference engine utilizing the Prolog logic engine was implemented. Therefore the rules were written using Prolog's basic rule structure.

The interface was a simple question and answer dialogue with two types of question:

1. " What is the . . .?" and

2. "Is the . . . ?"

to which the user could answer, either YES or NO. This type of interface was used as it is quick to implement and adequate for producing a usable system that reflects the basic ideas of an expert system.

In the knowledge base the choices of names used in the rules were particularly important as the module must be understandable to the domain experts. Soil science has a reasonably well defined and well documented vocabulary for most of the features described in the field descriptions. This vocabulary was used in the rules in the context normally used by a pedologist. Careful choice of names for terms and the use of comments made both the interface module and the inference engines comparatively easy to follow.

The initial iteration of the prototyping cycle partially satisfied the goals set out for the prototyping exercise. Development of an expert system was confirmed to be possible and the expert approached expressed an interest in participating in the project. The expert found the rules easy to read and understand. Two deficiencies were immediately apparent when the expert used the system.

1. If the field data was accurate then a way of volunteering the data was required.

2. If field data was going to be incomplete or possible inaccurate then a method of dealing with uncertainty needed to be chosen.

3.3 Developing the Second Prototype

The goals of the initial prototype were expanded to address the problems identified in the evaluation phase. The primary aims of the second iteration were to :

1. Allow the user to volunteer soil profile data.

2. Explore the use of the Certainty Factor Model as a method of dealing with incomplete field data and weighting the rules so the hypothesis can be given a certainty rating.

3. Demonstrate to the domain expert a system that used a method of determining the strength of accumulated data to identify a possible solution.

The specific design objectives extend the four outlined for the initial prototype to include:

- extend the interface to include a profile description collection module.
- extend the interface to include an on-line help module.
- extend the inference module to process uncertainty

The help module was considered necessary because users volunteering information need to know what vocabulary the system expects them to use and the different options the system knows about. As well the development of the help module would give a useful indication of the type and scope of explanation that may be required in a full soils identification system.

3.2.1 The Interface

Soil descriptions tend to be rather lengthy. A menu system was developed to enable the user to volunteer information about their field data. The menu mode is particularly suited to the input of soil features as each feature has a well defined set of descriptors or attributes. These are the same as the vocabulary used in the knowledge base. The descriptor sets were used to develop the menus for recording profile information.

Two types of menu were designed:

1. Type 1 menus require the user to select the appropriate criteria for a specific feature by keying the associated number, figure 3.03.

2. Type 2 menus require the user to type in the criteria using only the descriptors that are displayed. This was necessary for some soil features as the meaning of a descriptor depends on the context in which it is used, figure 3.04.

```
                         soil_classification
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
     Describe the TEXTURE for the A_horizon

     1 -sand                 9 -fine sandy loam    16 -silt
     2 -very coarse sand    10 -sandy loam         17 -silty clay
     3 -coarse sand         11 -medium sandy loam  18 -silty clay loam
     4 -medium sand         12 -clay loam          19 -silt loam
     5 -fine sand           13 -sandy clay loam    20 -loamy
     6 -loamy sand          14 -sandy clay
     7 -coarse loamy sand   15 - clay
     8 -fine loamy sand

     Enter choice:
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                 Type the OPTION NUMBER for the chosen criteria
                      PRESS RETURN - if you have no data
```

Figure 3.03 A Type 1 Menu

```
                        soil_classification
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
     Type in STRUCTURE for the A_horizon

          GRADE OF DEVELOPMENT                          SIZE OF AGGREGATES
          structureless                              very-fine(thin)
          moderately_developed                       fine(thin)
          weakly_developed                           medium
          strongly_developed                         coarse
                                                     very_coarse
                             SHAPE OF AGGREGATES
          massive              single_grained        platy
          prismatic              columnar            granular
          blocky                   nutty             crumb

     Enter description:
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
           Type description. Use UNDERSCORES between words(No Spaces)
                      PRESS RETURN - if you have no data
```

Figure 3.04 A Type 2 Menu

The help module did not require much extra information to be stored because the display of sets of descriptors used in the profile description menus were used as the basis of the help screens. Help screens were to inform the user of the possible values a specific soil feature may take.

As the entry module for volunteering soil profile field data was menu driven, for consistency the simple natural language questioning interface of the first prototype was modified to have a similar menu structure. Text templates for each kind of soil feature were designed and these are stored in the interface module. Each soil feature has associated with it a specific question. Two types of question are asked. Type 1 questions required a "yes/no" response, figure 3.05. Type 2 questions require specific numeric values to be entered, figure 3.06.

```
                         brown_grey_earth               A_horizon
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


        Is the texture of the A_horizon a sandy_loam ?


        Enter choice:
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                    YES - if true           NO - if false
                    PRESS RETURN - if you do not know
                               H - for help
```

Figure 3.05 A Yes/No Type 1 Question

```
                       brown_grey_earth               soil_forming_factor
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


        What is the average rainfall in mms ?


        Enter value:
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                      Enter the appropriate value
                   PRESS RETURN - if you have no data
```

Figure 3.06 A Type 2 Question

3.3.2 Uncertainty

The two methods of dealing with uncertainty that were considered in this project are Bayesian Logic and the Certainty Factor Model. The Certainty Factor Model was used in the enhancements to the prototype because at this stage there is no suitable database available for calculating the *a priori* values required by the Bayesian model. As Shortliffe and Buchanan (1984) have observed most experts find the task of supplying *a priori* values very difficult.

It has been noted that although an expert will supply a numerical weighting to indicate their belief in a set of evidence supporting an hypothesis (i.e. $p(h|e) \approx 0.8$) they are uncomfortable with the corollary that the evidence also supports the case against the hypothesis (i.e. $p(\sim h|e) \approx 0.2$) . Shortliffe and Buchanan state :

> " that the numbers the expert has given should not be construed as probabilities at all, that they are judgemental measures that reflect a level of *belief* "

To incorporate uncertainty into the prototype both the inference engine and the structure of the rules in the knowledge base had to be extended and the user interface modified. Each rule in the knowledge base had to have associated with it the reliability factor indicating the expert's belief in that specific rule's weight to support the hypothesis. Each of the rules supporting the hypothesis represents an independent piece of evidence.

With the simple Yes/No rule system only when all premises were true was the hypothesis valid. One of the purposes of using an uncertainty model is to allow the user to indicate either that the value of a specific fact is not fully reliable or to indicate that the value for the fact is unknown. To calculate the belief factor for a rule the minimum of the belief values for each of the individual premises is multiplied by the reliability factor for the rule. Therefore each of the premise in the antecedent have equal weight in determining the evidence for or against the current hypothesis.

When trying to assign a belief value to the rules used in version 1 of the prototype it was found that the premises in the soils classification rules did not have equal weight. The rule in figure 3.07 highlights this problem.

```
if
        the location is Central Otago                              &
        the texture of the A and B horizons is silty or sandy      &
        the overall colour of the soil is grey             &
        the A horizon has a weakly developed platy structure       &
        the profile contains an accumulation of soluble salts
then
        there is strong evidence that the soil is a brown grey earth (0.95)
```

Figure 3.07 Rule showing premises with unequal weights.

The third premise about soil colour is of less significance than the first three. If a users belief in the values of the four premises were 1, 2, 0.8, 0.2 then the certainty factor for the overall rule would be only 0.19. This vastly underestimates the evidence of the first three premises and the certainty should be at least 0.9.

To overcome this problem the method of representation, the domain knowledge had to be re-evaluated. This re-evaluation indicated that the rules would have to be rewritten to reflect the differences in the relative weights of the evidence of the individual premises.

3.3.3 Assessment

When the system was used by the expert the new interface was found to be straight-forward to use and no more inflexible than the forms that the N.Z. Soil Bureau used for recording field data for entry into their computerized data bank. The questions were still rather stilted and the expert indicated he would like to be able to select answers or responses from the set of possible values a feature may take rather than having to answer a query about each in turn.

The module for volunteering soil profile data was acceptable. Having to indicate the number of horizons that were to be described at the beginning of the soil description did not cause expected resistance because as the expert explained on first observing a profile the identification of the different horizons was often the first step before recording more detailed observations.

The experts were enthusiastic about the ability of the system to rate the hypothesis. They were not quite so eager to provide the reliability ratings for each rule.

3.4 Criteria for further development of SES

After developing this identification stage prototype a full scale development of a soils expert system was considered feasible. Therefore it was decided to utilize the remaining four stages in the expert development methodology as outlined by Hayes-Roth to guide the further development of the soils expert system referred to as SES. At each stage prototyping would be used as the main experimental tool for:

1. Clarifying the requirements of the different modules and user needs at the Conceptualization stage.

2. Exploring possible structures for storing knowledge at the Formalization stage.

3. Implementing the main system modules for the final system.

The information for developing the knowledge base for SES would come from a number of different sources including published texts, teaching material developed by Massey University Soil Science Department, published material from the DSIR Soil Bureaux and unpublished field notes and reports. These are outlined in appendix C.

The main modules of an expert system provide one level of suitable subdivision for incremental development of the system. The knowledge base, inference engine and human interface form the core of the expert system. The most important module being the knowledge base. Therefore the main thrust of the conceptualization and formalization stages is directed towards the design of the knowledge base. As the users provide the main purpose for a system the design of the user interface will also be given extra emphasis.

The second level of subdivision is in the implementation of the domain knowledge. The hierarchical structure of the soils classification system allows for the possibility of implementing the different levels of the classification in stages. Early prototypes have concentrated on the soil class and soil group, implementing the breadth of the classification structure. Later prototypes will examine problems associated with implementing a segment using the depth of the classification structure. The initial versions of the system will develop the most readily available domain knowledge while later prototypes will incorporate the more obscure and less common soils domain knowledge.

3.5 Summary

Although the check lists discussed in section 3.1 indicated that a soils expert system was possible, the feasibility of SES was clearly established by the development of the prototype. Evaluation of the prototype indicated the directions for further development of the expert system. Two important points arose from this stage:

1. A clear objective of stating why the project is being undertaken is required.

2. Good communication between expert and developer is essential.

Setting aims and objectives is necessary before any process or activity is to be computerized. This is particularly necessary when developing an expert system because of the absence of clearly defined methodologies for guiding expert system development.

Dryer (1988), in discussing the application of A.I. techniques to physical and biological science, states that

> " Many otherwise successful projects have been discredited because expectations grew as the work progressed. Always remind yourself of your original intentions. "

The expert has the knowledge necessary for developing the knowledge base for an expert system. To enable the knowledge engineer to structure and record the knowledge for computerization a meaningful dialogue must be maintained between the expert source and the knowledge engineer. This dialogue not only transfers the expert's domain knowledge but should also indicates how the knowledge can be presented in a meaningful manner. Many expert systems have been developed that adequately solve the required problems but these systems are not used. Even if all other factors for developing an expert system are favourable, good communication between the experts and the developers is imperative for the development of a usable system.

Chapter 4: The Conceptualization Stage

Knowledge is the key to expert systems. The design of the knowledge base is a major objective of the knowledge elicitation process. The knowledge engineer works with the domain expert to identify the domain concepts and systems, and to identify the form of the knowledge within each system. Identification of the form of the knowledge becomes the main purpose of prototyping after the feasibility of the project has been established. From the evaluation carried out during the first stages of the prototyping process it should be possible to characterize the domain descriptors: the objects, the systems, the inference structures and their interrelationships. The specification describing the domain becomes a conceptual model for the knowledge base.

The composition of the user group with respect to their knowledge about the domain will influence the level of detail and quantity of the domain specific knowledge included. The role of the expert system when interacting with the users will also impact on the type and detail of knowledge required. Determining the users requirements is an important step in the development of the knowledge specification representing the domain.

The specification of an expert system should include a description of the requirements of the different facilities that the expert system is to provide. Regardless of implementation details these facilities can be viewed as separate modules each of which will impose limitations on the final conceptual design. The functions of these basic modules were outlined in chapter 2. This chapter outlines the requirements of these modules along with the users' requirements.

4.1 Steps in the Conceptualization Stage

Hayes-Roth et al (1983) interprets the process of conceptualization as the stage at which consideration is given to the meaning of

> " the key concepts, relations and information flow characteristics needed to describe the problem-solving process in the given domain."

A detailed account of the proposed system should be developed using words, diagrams and examples. This description should be independent of specific design implementation details since, the knowledge base is the most important component of an expert system.

The methodology for specifying an expert system at the conceptualization stage is divided into two main steps:

1. Outlining the requirements.

2. Building a conceptual model.to represent the domain knowledge.

This chapter describes the requirements for SES while the next chapter discusses the development of the conceptual model. Outlining the requirements for an expert system is also divided into two steps:

1. Outline the Users requirements:

2. Outline the types of knowledge the different modules in an expert system need to meet the users' requirements

4.2 Specifying Requirements from the User's Perspective

A requirements specification indicates "what the system will do". It is not concerned with how the tasks are performed. Because of the nature of expert systems in general, as indicated by Partridge's incomplete specification model, detailed requirements are impractical. The specification of requirements should indicate the general goals of the system as seen from the users point of view because a consultative expert system is interacting with the users. The needs of the users impose a particular set of constraints on the conceptual description of the domain.

The description of the knowledge base which stores the domain knowledge must take into account the requirements of the users. The users requirements will also impose limits on the design of the different facilities and functions that an expert system provides. This in turn will affect the kinds of knowledge required in the knowledge base.

Three user groups for SES were considered:

- the soils science expert
- the student
- the general agriculture adviser

Each of these groups of users has different needs and different roles. Therefore each group's needs have to be examined separately. The main emphasis of SES is initially towards the student user group as end users of SES. Secondly, the experts are considered as they are the source of the knowledge that goes into SES. The needs of the third group are seen as coincidental at this stage of development, due to the time constraints and limitations on access to a suitable group of users.

4.2.1 The Domain Expert's Requirements

The methodology that the student is expected to use when identifying a soil is similar to that used by the pedologist when he is identifying a soil from a verbal discussion with a client. When the pedologist is asked to identify a specific soil the usual strategy followed is:

1. Identify the position on a soil map and use the classification information to create a set of possible classification units.
2. Use soil forming factors to refine the hypothesis list:
    a. Identify the major topographical features, i.e where the soil lies within the landscape.
    b. Determine the drainage characteristics of the soil.
    c. Check any vegetation or climatic data that may be relevant.
3. Use major profile characteristics of series and types to narrow the hypothesis set to one or two candidates.
4. Confirm the hypothesis by checking further profile characteristics:
    a. Check that further profile characteristics are consistent with hypothesis.
    b. Compare with similar soils.
    c. If characteristics indicate two classification units are equally good solutions consider the possibility of an intergrade.

At each step the number of possible soil series is reduced. One important point that often leads to mis-identification of soils by a novice is the assumption that because the map indicates a soil is of a particular type that it must be so. Soil maps are usually published at a scale of 1:63360. This does not allow for the detailed mosaic of soils in individual paddocks and the maps are therefore only a generalized guide to the soil type.

The pedologist's use of the system will be limited to the knowledge acquisition phase and testing that the knowledge base is valid. The system should be able to present the pedologist with a series of instructions and questions that enable knowledge about soils in a surveyed area[1] to be entered into the system.

---

[1] Usually on a county basis.

4.2.2 The Student User's Requirements

The student user population will use the system to help them complete their farm project reports which are submitted after each practical farm experience. These reports are about the individual farms where the agricultural science or horticultural science student carried out practical work. They are completed independently of any explicit direction from the lecturering staff, other than an outline specifying the general contents of the report. Although each report must include details about the soils on the farm, the students will not necessarily have had any experience in describing or classifying a soil. The system therefore has to be self-explanatory and easy to use.

The student user population has a wide range of skill levels in both the domain and computing. The range of domain knowledge is from the novice to those students who have had over two years experience in soil science. The experience with computers ranges from limited to students with considerable experience with a number of different computer packages and some programming skills. This wide range of experience means that the proposed system will have to be able to present a variety of levels of interface.

The novice student will require questions couched in terms that are familiar to most layman with a farming background. When technical terms cannot be avoided, access to a comprehensive dictionary will be necessary. An aim of the system is to reinforce knowledge about soil development processes so the student will need access to information about basic principles of soil development associated with the terminology. As one of the main aims of the system is to reinforce a systematic strategy for identifying a soil, the system should present the student with a problem solving pattern that reflects that strategy.

The experienced student will have completed a formal soil description and will want to enter this information in a form appropriate to the description, even though a full description of the soil may be unnecessary to the system for identification purposes. Ideally the user interface should present the student with a form similar to that used by the DSIR Soil Bureau for recording field data. Entering this data is time consuming and the facility to obtain hard copy of a session will be useful. Printouts maybe useful as part of student assessment. The facility to save a partially completed session so that it can be continued at a later time is essential.

The explanation system will be an important feature for the student user and should be able to describe not only how the system arrived at a particular state but also the reasoning that underlies the specific knowledge used to make decisions. This includes both deep knowledge and compiled knowledge which may not necessarily all be required for the system to draw a conclusion. It is envisaged that in many cases the system may be unable to make a clear decision on the classification of a soil due to incomplete data or incomplete knowledge in the knowledge base. The system should therefore be able to give a list of the possible soil types the soil could belong to, or at the very least a list of the possible soil series. In these cases the system should be able to direct the student as to what further data is required to determine whether a soil is of a particular type. The explanation module should be able to explain why the particular data is required with regard to comparing different soil series or types, and, where appropriate, explain any significant underlying principles.

4.3 Specifying the Knowledge Requirements of the Modules

One of the strengths of Expert Systems is their modularity. To maintain this modularity it is necessary for the form and function of each of the system's modules to be kept as general as possible. The knowledge that is specific to the domain should be stored in the knowledge base. In early expert systems, domain knowledge such as strategic knowledge was often imbedded within the module in which it was used. Not until the requirements for good explanation, acquisition and tutoring were examined (Clancey, 1984; Davis, 1978; Neches, 1985; Aikins, 1983) did it become apparent that much of this implicit knowledge should be stored explicitly within the knowledge base.

The following sections discuss the knowledge requirements of the basic modules of an expert system. Prototyping has been used where appropriate to explore specific aspects of the requirements for the different modules. An analysis of each prototyping exercise helps to refine the requirements. The requirements are not always precise and as fully detailed as those that result from a formal system analysis, but they attempt to describe features that will make the system meet the users requirements.

4.3.1 Inference Module

The inference engine requires access to sufficient knowledge about the domain to reach a conclusion about a specific problem with a similar success rate to a human expert. Implicit within the code of the inference engine is knowledge about the structures and relationships of the objects it is manipulating. Strategic knowledge may be part of the inferencing mechanism or it may be stored in the knowledge base and used as required. The method of representing strategic knowledge depends on the level of sophistication or generality of the system or the characteristics of the problem domain.

From the discussion about user requirements it is clear that the inferencing process should closely model the strategy used by the domain experts. It is not sufficient just to make the interface module simulate the steps because the explanation to the student about how conclusions are reached must reflect this strategy. One of the characteristics of the strategy is that although the steps appear to be general, for example "determine major topographic feature" for the system to correctly reflect the expert's method it is inappropriate just to check sequentially through a set of all the values that the system knows about. The features to be checked depend on which soil units are appropriate, as identified on the soil map. If the soil is located in the Manawatu coastal sand country then the strategy should concentrate on critical aspects of sand country topography.

The strategy for checking profile characteristics for one soil unit may be to look at one or more overall characteristics (grainsize distribution) while for another, unit-specific features are most significant. Therefore strategic knowledge for the soils system needs to be associated with the context in which it is being used.

4.3.2 Explanation Module

Many of the requirements of a good explanation system are similar to those of an on-line help system. The help system is a facility that should provide the person who uses the system as a tool, with enough information to feel comfortable about using the system. The explanation system for an expert system has a similar but extended role. The explanation module must also be able to explain the decisions made by the system during the consultation to the satisfaction of the user.

All on-line help systems are built around a model of the user. The simplest help facility will be based on the needs of the mythical average user as defined subjectively by the system designer. At the other end of the scale is the intelligent help system (Erlandsen and Holm, 1987) that builds up an individual model of the user as the consultation progresses. Rich (1986) has identified three dimensions for characterizing user models. The model dimensions are:

1. Canonical versus individual models

2. Explicit versus implicit models

3. Long-term versus short-term models

Differing levels of explanation or help will be required by the projected users of SES. The levels are dependent on the user's degree of expertise in both the problem domain and their level of computer literacy. As well, differing kinds of explanation will be required at different stages within a consultation. The most advanced explanation systems build individualized implicit models of the user using both short and long term models as appropriate. Such an intelligent explanation system is beyond the scope of this project. A practical alternative is to build explicit, long term models of different levels of user and allow the user to move between the levels at will.

To determine what level or levels of explanation were required by the users a simple prototype was built to explore the users' requirements for explanation. From the discussions about classification in the soils domain required for the main prototype discussed in chapter 3 a simple Yes/No decision tree for determining the soil group was developed. The specific objectives for this prototype were to:

1. Identify what explanation facilities were required.

2. Identify the types of information required to generate explanations.

Only the simplest of the explanation facilities, the usual HOW and WHY options were implemented. The explanation option WHY allows the user to query the system "Why are you asking this particular question?". The example rules (figure 4.01) form an inference chain used by the system to establish that a soil is a Yellow Brown Pumice. During the consultation the system will use rule 26 to try to prove that the soil is volcanic. If the user asks WHY when the system requests information about the "high water table" then the system will output rule 26. The system is telling the user what it is trying to prove. On being asked WHY again the system will output, in turn, all the rules that lead from its present position to the hypothesis. In this case rule 28 is the next rule in the inference chain so the system will indicate that the consequence of this rule (YELLOW BROWN PUMICE) is the current hypothesis.

```
rule(1,if (neg 'B horizon distinct') then class('AZONAL') ).

rule(11,if class('AZONAL') or (neg 'rock type quartzofeldspathic') or 'parent material aeolian
sand' then (neg class('ZONAL')) ).

rule(20,if (neg class('AZONAL')) and (neg class('ZONAL')) then class(INTRAZONAL) ).

rule(26,if class('INTRAZONAL') and (neg 'ground-water table high') and 'rock type volcanic'
then 'VOLCANIC SOIL' ).

rule(28,if 'VOLCANIC SOIL' and 'parent material a pumice coarser than sandy loam' then
'YELLOW BROWN PUMICE'

1 »»» 11 »»» 20 »»» 26 »»» 28
```

Figure 4.01 Rules from the Explanation Prototype.with the associated Inference chain.

If instead the user asks HOW then the system indicates to the user what has already been proved in pursuit of the current hypothesis. Therefore if the user repeatedly asks HOW from rule 26, rules 20, 11 and 1 would be displayed in turn. When all the currently proven rules have been displayed the system indicates what the current hypothesis is and what hypotheses have been rejected during the current session.

A number of points relevant to the user's requirements of an explanation system were identified when the system was demonstrated

1. The users were confused about what WHY and HOW meant. A HELP option is required to explain how they work.

2. The HOW would be more relevant in most circumstances if it was associated with a particular premise rather than a rule as a whole.

3. As well as HOW and WHY there was a definite need for a third option EXPLAIN which would make clear the significance of a rule to the inference structure used in the system. For instance an explanation for rule 26 would include reference to the features that differentiate intrazonal soils from other soil classes and that by establishing the fact that a soil was volcanic the system could limit the number of possible hypothesis.

4. An EXPLAIN option can also be associated with the individual premises. The option could be used to make clear the relevance of a premise to either soil identification or soil classification. Rule 11 tries to establish whether a soil can be classed as zonal. An explanation of zonal would be

   " Soils groups which given adequate time for formation and on a normal relief and ordinary siliceous parent material, show distinguishing characteristics due to the processes controlled by the climate and associated vegetation of each zone throughout New Zealand."

5. Pedology uses a large number of technical terms and potential users of the system may not be sufficiently familiar with these terms to answer questions clearly. Rules using technical terms can either be reworded using simple terms or broken down into smaller steps. A rule asking whether the climate is subhumid could be reworded to ask about mean annual rainfall and mean annual temperature. This simplification is used to CLARIFY the question.

Even though the premises for the rules in this system were in the form of short phrases the repetition of the text for the rule was not always easy to understand. In a more comprehensive system the structure of the premises will have to be more formal making them even more unreadable to the user. From informal observations of the users interacting with the system it was obvious that the wording of a question also played an important part in the users understanding of the system. From these observations two points emerge:

1. Unless a reasonably sophisticated natural language processor is available each rule needed a textual version that could be presented to the user when required.

2. The wording suitable for a question was not necessarily suitable to use as a statement in part of an explanation.

To be able to provide a reasonable level of explanation to the user the system will have to store a large quantity of text information

1. for each different type of premise:
   - a text version
   - an optional question version
   - a reasonably detailed account of its implications in soil science
   - help information outlining the possible values a feature may take, and their significance.

2. for each rule type:
   - a structure for a WHY explanation
   - a structure for a HOW explanation
   - an explanation of the rules use in the inference structure
   - a list of CLARIFY rules

3. for each system option available to the user:
   - at least one help screen to make clear the use of that option

4. for each stage in the consultation processes:
   - help facilities explaining what the system is trying to do at this point in the consultation.

4.3.3 Communications Module

A number of different interfaces have been used successfully with expert systems. The communication module for an expert system may become expert system in its own right as TEIRESIAS (Davis 1977) in the MYCIN project. The expert system HEARSAYII (Hayes-Roth, 1985) is a natural language system designed to interpret data base queries. HEARSAYII has been developed to interpret a subset of spoken language.

Menus are the usual interface for the simpler expert systems shells examples being ES/P ADVISOR and EXPERT-EASE (Harmon and King, 1986). The more sophisticated expert system development environments supply tools to develop interfaces using WIMPS[2] and graphics. The interfaces to OPS5, KEE and LOOPS (Harmon and King, 1986) all make use of windows when run on the appropriate hardware. KEE allows the developer to create graphical windows to demonstrate functions from the domain to the user. The user can interact with the screen, with the aid of a mouse, to add to the data during a consultation. A newer technology that has not as yet been used as part of an interface to an expert system is interactive video stored on optical disks This has exciting possibilities

As with the explanation facility, it was decided to use prototyping to experiment with the types of interface that would be most appropriate for the users. The straight forward menus for the interface of the main prototype were discussed in chapter 2 . Although this method is successful on the hardware used, the availability of more sophisticated interface features such as those on the Apple Macintosh enable more appropriate alternatives to be considered.

The use of WIMPS and graphics interfaces was explored by implementing two prototypes. The first interface prototype was implemented using LPA PROLOG on the Macintosh to investigate the use of dialogue boxes. The second prototype used the graphical data base FILEVISION to experiment with selecting soil features from a specialized type of menu built on graphical representation of those features.

In the first instance, the three types of interface used in the main prototype, were implemented using dialogue boxes in the prolog interface prototype. These types shown in figure 4.02 are:

- the choice from a number of alternatives
- the yes/no question
- and the request for a numeric value

The experts found selecting choices by pointing with the mouse much easier than entering the same option via the keyboard.

Each soil feature has associated with it a subset of possible values which can be viewed as a fuzzy set around the central value of the generalized description. In the main prototype the system often had to query the user repeatedly asking about each value in the subset.

2 WIMPS : W = windows, I = icons, M = mouse, P = pull down menus

This process required the user to respond for each member in the subset. The dialogue boxes were implemented so that the full subset of values were displayed on the screen thereby giving the user more information on which to base a response and cutting down the number of interactions between the user and the system. The user simply selects one of the values listed (or none where appropriate).

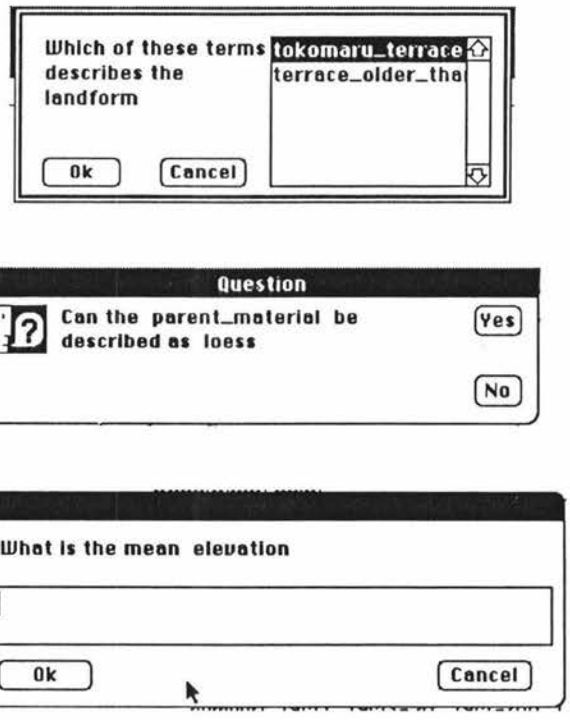


Figure 4.02 Examples of dialogue boxes.

Many of the characteristics of soils information are better presented in graphical form. The graphical database interface provided by FILEVISION has facilities for drawing objects and associating stored text records with these objects. The most important strategy for identifying soils in N.Z. is the position of the soil in the landscape. Most soil bulletins include a table and a generalized block diagram of the morphology of the region. These associate the different soil units with the main geomorphic features. This information is most important to an expert system on soils as the geomorphology of the site limits the number of possible soils.

In the menu based prototypes the expert system queries the user about the specific landforms that occur within the domains knowledge, for example:

"Does the soil occur on the Tokomaru Terrace ?"

The main problem with this question is that the user has to understand the location and nomenclature of the local terrace system of which the Tokomaru terrace is a part. With a graphics interface the user can be presented with the block diagram of the district (figure 4.03). The user can then point to the landscape element on which the soil in question is found. Using a mouse or other pointing device this is a very simple task and the system can highlight the selection so the user knows the location and extent of the landscape feature that has been selected. A user will usually be able to visualize where a soil is in the landscape without knowing the technical term for the geomorphic feature.

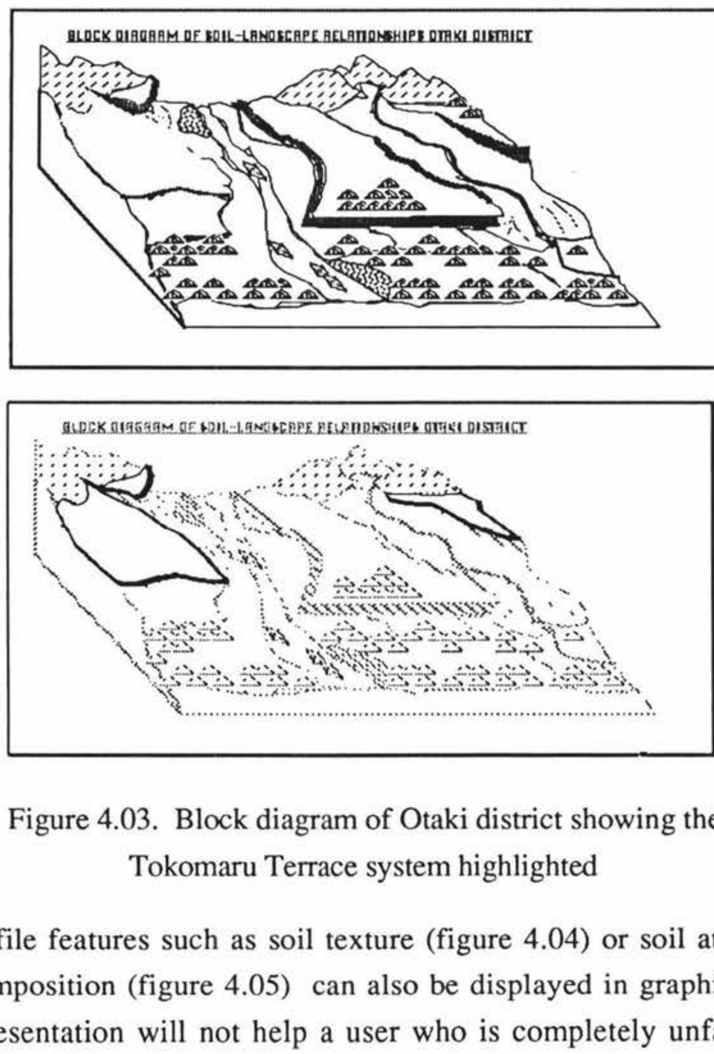


Figure 4.03. Block diagram of Otaki district showing the Tokomaru Terrace system highlighted

Many soil profile features such as soil texture (figure 4.04) or soil attributes such as percentage composition (figure 4.05) can also be displayed in graphical form. Such graphical representation will not help a user who is completely unfamiliar with the technical terms. However it is very useful when this type of information is presented to students or agriculture advisers who are partially familiar with soil technology as similar diagrams are standard in both textbooks and handbooks on New Zealand soils.

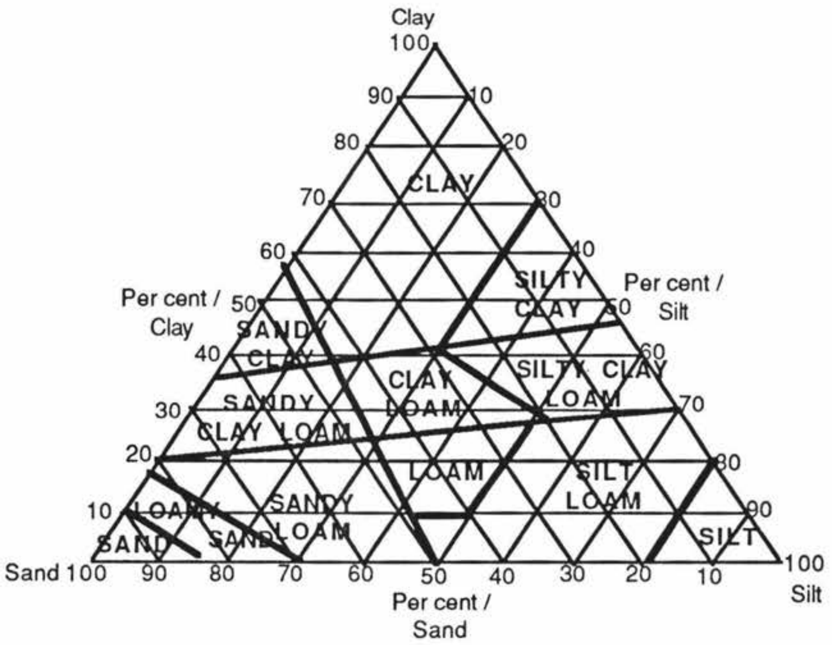


Figure 4.04 Diagram for soil texture assessment.

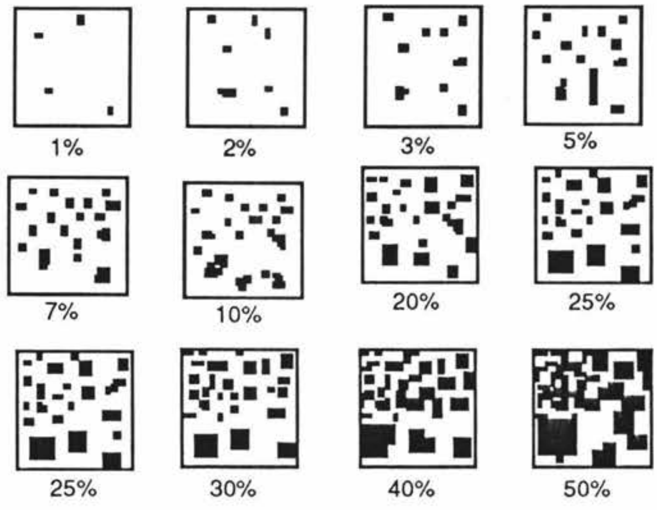


Figure 4.05 Percentage diagrams.

The experts were most enthusiastic about this interface. Their experience as consultants to naive users indicated that this type of interface would convey to the user the types of information they were trying to obtain A use that was not envisaged but suggested by the expert was that such a tool may be a practical way of supplementing consultation interviews. With such a tool the expert could display in graphical form the information required from the user.

For students this type of interface was also considered ideal as collecting quality data about soils in the field depends on good visual discrimination skills. One of the aims of a course in soil science is to improve the student's powers of observation. By using a graphical interface these skills of observation are reinforced.

4.3.4 Acquisition Module

The main prototype expert system is based on classifying soils from the Horowhenua County but for the system to be useful to a wider audience information about soils in other counties needs to be either added to the existing knowledge base. An alternative solution would be the creation of a set of parallel knowledge bases that are linked together. This early prototype had a simple module for entering field data into the dynamic data base that could be modified to allow entry of much of the factual knowledge in the knowledge base. The module provided facilities for directly entering field data and although not very flexible was satisfactory to use. The expert had no difficulty with the menus but the module does not allow the user to change direction and once information was entered it could not be edited. Ideally much of the static factual knowledge required in a district knowledge base should come directly from the databases each DSIR Soil Bureau district office maintains.

Regardless of the source of the knowledge the acquisition module requires a schema (Davis, 1978) documenting all of the structures in the knowledge base. The schema is used to structure new knowledge input by the user. It can also be used to query a user about apparent gaps in new knowledge based on the patterns of knowledge currently in the knowledge base. The ability to define subschemas would enable the definition of structures that more accurately reflect the nature of some of the atypical soil classes. Schemata should be used not only to record the factual knowledge, as in data bases, but also to record the structure of all the forms of knowledge in the knowledge base.

One of the functions of the knowledge acquisition module is to validate, verify and check for consistency all new knowledge as it is added to the knowledge base. Therefore knowledge about possible value ranges, suitable comparisons operators and patterns of similar types of knowledge should be available in the knowledge base. Once knowledge has been added and checked in isolation it should then be checked with the current knowledge to ensure the consistency and integrity of the entire knowledge base is maintained.

The knowledge acquisition module needs to be closely associated with the explanation module. this would permit existing knowledge to be browsed and also used to provide the user with examples of how similar knowledge has been recorded. The explanation system can also provide information about terminology usage and of course information on the type of knowledge required to provide explanations about the new knowledge.

Due to the project limitations only a simple knowledge acquisition system is possible for the current soils system. The highly structured way in which pedologists describe soils means a simple module should be adequate. The system should present templates that are similar to that already in use by pedologists to store soils knowledge.

The second identification stage prototype explored the use of using a simple menu template interface allowing users to volunteer soils information. Evaluation of this interface by the experts was favourable and indicated that a similar approach could be used in a knowledge acquisition module. For knowledge of a less familiar type the system should provide a suitable dialogue to guide the experts in recording their knowledge.

4.4 Summary

The different kinds of knowledge have been outlined in section 2.2.1. Figure 4.06 summarizes the kinds of knowledge required by each of the main modules comprising an expert system.

Type of Knowledge	Inference	Communication	Explanation	Acquisition
Surface	√		√	
Deep			√	
Compiled	√		√	
Support			√	
Strategic	√		√	√
Descriptive		√	√	√
Dynamic	√		√	
Meta	√	√	√	√

Figure 4.06 Summary of the kinds of knowledge required by the different modules

Meta knowledge is required by all the modules. Descriptive knowledge is required by all but the inference module. Although the inference module. uses the domain objects this use is via the other forms of knowledge. The inference module does not require knowledge about the characteristics and definitions of the objects. The explanation module has to be able to explain all aspects of the domain knowledge used by the system. Therefore it requires access to all forms of knowledge.

Chapter 5: The Conceptual Model of SES

The knowledge base module forms the core of the system. The four other modules: inference engine, communications, explanation and knowledge acquisition, all require a basic range of knowledge to function effectively. The knowledge base should contain all the knowledge that is required by the other modules in the system. This includes both the knowledge that is common to one or more modules and any knowledge that is specific to individual modules. The specification of the contents of the knowledge base must take into account the different requirements of each of the modules.

The knowledge base representation chosen should be flexible enough to enable all the different types of knowledge to be specified. However as pointed out by Mylopoulos and Levesque (1986)

> " The basic problem of knowledge representation is the development of a sufficiently precise notation with which to represent knowledge. "

This chapter examines some of the different ways in which other authors have specified expert based systems. Three aspects are identified as important for specifying an expert system. These are: the inference structures, the objects and the functions. The discussion for the specification of SES in the second half of the chapter is structured around these three aspects.

5.1 Specification of an Expert System

The specification of the content of the knowledge base begins with an analysis of the domain when the problem is subdivided. Important terms and their relationships are recognized during the identification stage. The main systems within the domain have to be identified and their component parts recorded. These records form the core of the specifications of an expert system. The specification of the knowledge base is a conceptual model of the domain that can be mapped into a logical model. Buchanan et al (1983) suggests that

> " The knowledge engineer may find it useful to diagram these concepts and relations to make permanent the conceptual base for the prototyping system. "

Ideally the conceptual model should fully document the objects, relationships and functions representing the domain problem. The conceptual model can then be mapped onto an appropriate logical model using the formal representations available, such as rules, semantic nets, frames etc. This methodology is frequently not practical for problems requiring an expert system solution due to the ill defined components of the problem.

Expert system technology was developed to solve problems that appear to be inherently unstructured. Clancey (1985) showed that the expert system knowledge bases studied contained a great deal of structure regardless of the knowledge representation used. Although it is not possible to develop a well-defined specification for many expert system (Partridge, 1986) the forms etc that can be identified within the domain should be reflected in the design of the final system.

In discussing his program development method Partridge (1986), emphasizes the fact that the first specification of an expert system system is only an approximation of what is required. He terms his methodology the run-debug-edit cycle. The edit step refers to the editing of the specification before yet another version of the system is implemented.

The type of specification developed for an expert system program is described by Partridge as an Incomplete Specification Function or ISF. An expert system is deemed to be completed when an adequate approximation of the desired system has been developed (see figure 5.01). An ISF is made up of three components:

1. A well-defined component

2. A context sensitive component.

3. An unknown component.

The author proposes the thesis that all specifications are made up of these components but the specification for a traditional system consist of mainly the well-defined component with the other two components being insignificant. Alternatively, an AI program such as an expert system specification, the context sensitive component and the unknown component are a significant proportion of the intended specification. It should be possible to rigorously specify the well defined component of any system but the other components can probably only be determined by experimentation. An example of a context sensitive component of an expert system is the explanation facility.

Partridge considers the implementation code to be the actual specification of the system. The purpose of the run-debug-edit cycle is to develop a system that adequately meets the users immediate needs at which stage the specification is considered complete.

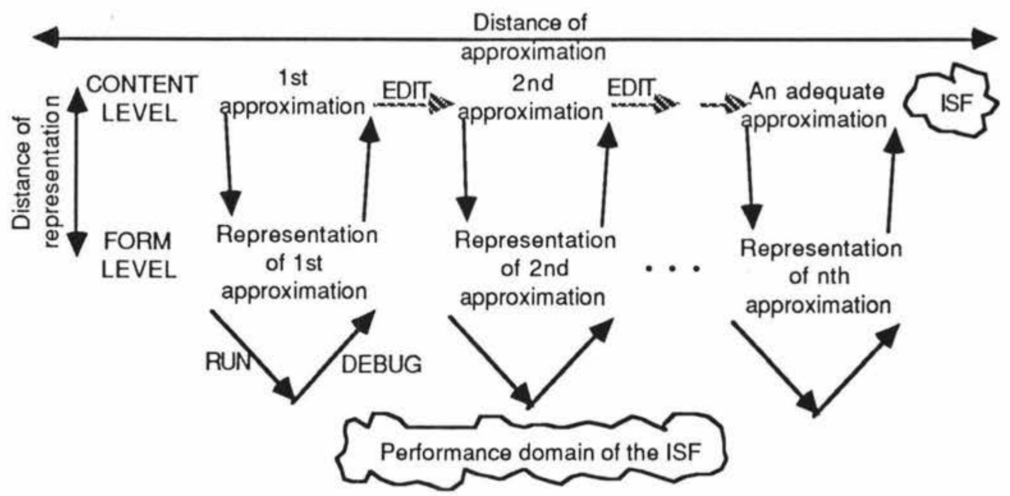


Figure 5.01 The RUN-DEBUG-EDIT cycle related to the ISF.
(Partridge, 1986)

Partridge's run-debug-edit cycle appears to be similar to the four stage methodology outlined by Hekmatpour and Ince (1986), with stages one and two combined into the Edit step. The content of the edit step evolves into the specification.

Clancey's (1985) method for examining a problem domain is closely related to his model of heuristic classification. An outline of the method is:

1. Describe the problem in terms of a sequence of operations that relate the systems which have been identified.

2. Identify all the alternative solutions in the solution space.

3. Determine the relative number of uncertain choices.

Ramsey, et al (1986) use a more general method in their list of steps for examining the problem domain. Again, the emphasis is on the need to identify which models or systems occur within the domain. They identify three important questions that the developer should address:

1. What is the pre-existing format of the domain?

2. What types of classification are involved?

3. How context-dependent does the inference process need to be?

The importance of Partridge's thesis is in the identification of the different aspects of the specification. When comparing his methodology with that of Hayes-Roth et al, the well defined component of the specification can be defined by the end of the identification stage and during the early phases of the conceptualization stage. Prototyping can be used to discover sufficient details of the unknown and context-sensitive components of the domain to permit an adequate conceptual model of the domain to be defined. The conceptual model may not be complete but should be adequately defined so that the design stage can be initiated with a high degree of certainty that the basic form of the knowledge for the system will be correct.

Clancey has provided a specification method that allows for the analysis of the nature of the problem domain. Inference diagrams (figure 5.02) provide a tool that enable this analysis to be documented in a form independent of the implementation, and fits into Hayes-Roth's conceptualization stage. The inference structures do not describe the objects that make up the components within the domain systems but provide a high level specification tool. Prototyping may be used in conjunction with the development of the diagrams. Documenting the specific inference paths identified in a domain can highlight where generalization or further discrimination is required to adequately describe the domain.

The questions raised by Ramsey (1986) highlight two further aspects of knowledge engineering. Firstly, to be able to specify the domain sufficiently the knowledge engineer will have to work closely with the domain experts to define the objects and the relationships that the expert uses to describe the domain. Secondly the ways in which a domain expert classifies the subject area will be both formal and informal. The formal methods are probably well described in text books. However, the knowledge engineer must elicit how the expert uses experience to identify those parts of the classification that are relevant to a specific problem. The more difficult task is to elicit how the expert modifies a formal classification to fit real world problems and to determine the informal classifications used by the expert to quickly identify possible conclusions.

5.1.1 High Level Specification using Inference Structures

Clancey's inference-structure diagrams are a form of knowledge specification at the conceptual level. They are independent of the actual representation used in a specific implementation for domains that can use the Heuristic problem solving method. The inference structure diagram (figure 5.02) can be used at two levels:

1. To show the specific inference paths between specific instances of data to a solution.
2. To generalize the domain

The purpose of these diagrams is similar to that of the data analysis methods used in database design. Figure 5.02, shows that the basic characteristics of an expert system are: data systems, solution systems and their inter-relationships.

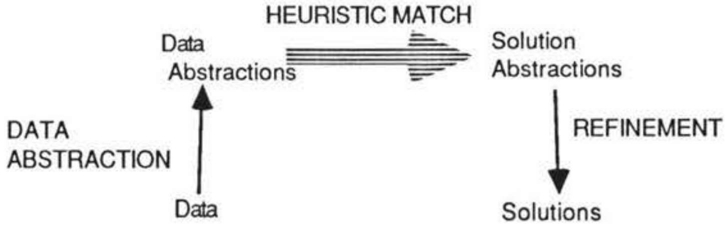


Figure 5.02 Inference structure of heuristic classification.
(From Clancey (1985), figure 2.3, p 296)

Clancey stresses that two conditions must be met for the match between systems to be considered as heuristic:

- the match is a "direct, non-hierarchical association".
- the match is between two distinct classes.

The term non-hierarchical implies that the specific rule is not one that is used to determine membership between different classes within a hierarchy. Heuristic matching is defined (Clancey, 85) as developing links that

> " capture incidental associations between a solution and available data, usually concrete concepts"

Heuristic links can be classified into four different types of association:

- agent
- causal or correlation
- preference
- physical model

Heuristic links are not always at the same level when mapping the data systems onto the solution systems. Also they are not the only kind of links between the data and solution systems. These non heuristic have also been identified by Clancey (1985). There are three basic relationships:

- Definitional - (including soft definitions)
- Qualitative
- Generalization

5.1.2 Specifying Domain Objects

The specification of an expert system includes the specification of the knowledge about the domain objects and relationships in addition to the associated facilities for manipulating the knowledge. The specification defines the conceptual design of the domain and provides the main frame-work for building a model of the domain specific knowledge. The representation used to describe the model should enable both the knowledge engineer and the domain expert to more easily comprehend the underlying conceptual model of the the domain. The representation technique should enable different views of the domain to be developed that reflect the context of that view.

Mylopoulos and Levesque (1986) have examined domain modelling as used in expert systems and presented a taxonomy of knowledge representation schemes. The taxonomy is based on the main aspects of each representation and recognizes four types of representation scheme:

1. Logical
2. Network
3. Procedural
4. Frame-based

The aspects considered important when devising this taxonomy were how the representation dealt with:

- individuals
- relationships
- states
- transformations

Knowledge representation has been a major concern of both artificial intelligence and database research. The models developed for database design tend to be better formulated. This is probably due to commercial influences. Brodie (1986), recognizes four generations of database models:

1. Primitive data models - files of records.
2. Classical data models - hierarchical, network and relational.

3. Semantic data models - entity relationship models through to semantic hierarchy models.

4. Special purpose data models.

Early data base design models were concerned mainly with the objects in the domain and identification of the relationships between objects in terms of ,one-to-one, one-to-many and many-to-many. Traditional stand alone applications each have their own files of data, duplicating much common data. By storing the facts in a data base a consistent view of the data is provided across the many different applications that use it. Early knowledge base representations concentrated on the semantic aspects of a domain, especially heuristic knowledge. The items stored in a knowledge base may have a number of different roles in the problem solving process. The role depends on the context in which the knowledge is being used. When a data base is set up not all the applications which will use a specific piece of data are known. Similarly, when specifying a knowledge base all the roles a specific item may take is probably unknown.

Although from the early studies the two disciplines appear to have little in common, comparison of the knowledge representation taxonomy and the generations of database models shows that both deal with similar aspects of the domain. As research has proceeded, both disciplines have recognized that objects, relationships, semantics and meta data, each have an important role in developing usable models of a real world domain. Models such as dynamic semantic nets occur in both taxa described above. This drawing together of two different research areas provides the knowledge engineer with a wider base of tools and models which can be adapted to represent specific domain knowledge.

The third generation of data base models developed from the need to be able to create a conceptual model of a data base, independent of the type of software that would be used to implement the resulting schema. A commonly used method of conceptual data modelling uses the Entity-Relationship model (Chen, 1976). This modelling method can be used for developing parts of the static knowledge base and for designing the structures required for parts of the dynamic data base in an expert system. The main deficiencies with the Entity-Relationship model for expert system design are:

1. No distinction between the different dimensions of the object-relation hierarchies.

2. No facility for composite objects.

3. No facility for ranking objects.

4. No facility for grouping objects.

5. Inability to show semantics.

6. No facility for defining variables

Semantic hierarchy models developed from a need to provide such facilities. These models also belong to network representation schema. The modelling techniques used in semantic hierarchies is also flexible enough to allow some functional aspects to be defined such as integrity checking, control structures and meta-data definition.

Semantic network models are based on collections of objects and how they are associated. They can be represented diagrammatically by representing the objects using nodes that are linked by associations represented by directed labeled edges, figure 5.03.

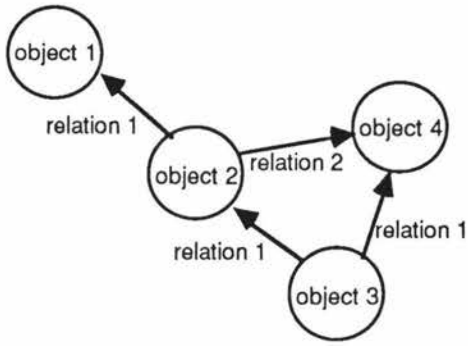


Figure 5.03 A Semantic Network Diagram.

The diagrammatic representation of the networks provides a method for a knowledge engineer to explore the structure of the domain knowledge. Networks can be created to represent different views of the domain model. The most widely recognized views are termed planes or dimensions. These are:

1. Classification - A type-to-token relationship which classifies an instance of an object with its generic groupings.

2. Aggregation - The common relationship "part-of" which defines the essential components of an object.

3. Generalization - A type to type "is-a" relationship that provides a classification hierarchy.

4. Partitions - A grouping of objects that have a defined association.

The use of semantic nets as a design tool has not been tightly defined therefore it is possible for the knowledge engineer to modify the methodology to suit the specific aspects of each problem domain.

5.1.3 Specifying the Functional Aspects

The description of the conceptual model of a domain should also include an outline of the main procedural and control aspects for problem solving in the domain. The main functional areas need to be identified and a description of the problem solving strategies developed. There are many tools for specifying the functional aspects of traditional data processing systems. These methodologies, such as Structured Systems Analysis (Gane and Sarson, 1979) are too detailed to be appropriate for specifying an expert system. This is because the execution path of a specific consultation session with an expert system, depends dynamically on the data representing the specified problem. Adapting the design tools used by these methodologies may provide acceptable tools for specifying the main strategies the domain expert uses. Data flow diagrams could be modified to show the main information flows in an expert system.

Johnson, Zualkernan and Garber (1987) have developed a methodology for specifying expertise. The methodology makes use of protocol analysis which is used by cognitive psychologists to investigate human problem solving. The main emphasis is on the behaviour of the domain expert, in this case interpreted as the functions the expert carries out. Johnson's model is based on the assumption that

> " the problem space is a fundamental construct of problem solving. We define a problem space as a set of problem states, and assume that problems are solved by moving from one state to another. "

The methodology requires two types of analysis to be carried out:

1. Syntax analysis.

2. Semantic analysis.

Syntax analysis identifies three categories of behaviour:

1.	Operations -	these are termed the primitive activities used to solve problems in the domain and are indicated by the verbs used by the domain expert.
2.	Episodes -	repeated patterns of operations.
3.	Data cues -	the different types of data that are used and are associated with each of the operations.

The semantic analysis of the domain expert solving problems in the domain results in the identification of six semantic categories which can be grouped into four main types:

1. The problem States - goals and solutions.
2. Functions - actions and abilities.
3. Conditions
4. Strategies

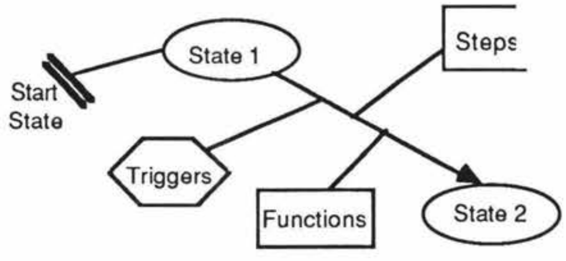


Figure 5.04 A State Specification Diagram.
(adapted from Johnson, Zualkernan and Garber, 1987)

Johnson, Zualkernan and Garber (1987) also provide a detailed methodology for identifying these categories and a diagrammatic tool for representing the analysis. The representation tries to show all aspects of the analysis on a single diagram, termed semantic specification diagrams. Due to the functional content of these diagrams and to avoid confusion with semantic nets a better name is 'State Specification Diagrams' (figure 5.04). Although the authors indicate that the methodology was developed to specify certain types of design and diagnostic expert systems, state specification diagrams can be adapted to specify some of the functional aspects of classification expert systems.

The two most important aspects brought out by Johnson's analysis are the identification of:

1. The different states in the problem solving process the domain expert uses
2. The different strategies used to move from state to state.

Strategies are defined by the data, the primitive functions and the steps in the path relating the two states. Specific data are viewed as the triggers required to enable each step in the strategy to be realized. A special start state is also defined.

5.2 Developing a Conceptual Model

The need for a more disciplined approach to developing a diagrammatic representation of the conceptual model of the domain specific knowledge was stressed in section 5.1.2. This need for discipline must be tempered by the need to find and use diagrammatical representations that are both flexible enough to represent new and possibly unexpected relationships, classifications and procedures that occur in a specific domain. The diagrams used must be easily understood by both the domain expert and the knowledge engineer and should form the basis for much of the discussion about the domain during the knowledge elicitation process.

By using such tools as semantic nets, state specification diagrams and inference-structure diagrams it is possible to represent domain knowledge at a conceptual level. The resulting representation of the conceptual model can then be used to select and design a representation for the production system that more closely models the form of the knowledge that exist in the real world. This view of the domain-specific world model will still be subject to the domain expert's and the knowledge engineer's opinions of the best way to represent the conceptual model. The knowledge representation requirements of extendability, simplicity and explicitness cannot be achieved without developing a conceptual design that in some way reflects the nature of the problem domain regardless of any bias in that representation.

The three steps identified as necessary for developing the conceptual model of SES are:

1. Develop a high-level conceptual model of the domain system.
2. Develop a detailed model of the main objects within the domain.
3. Develop a model of the important functional aspects of problem solving in the domain.

5.3 The High-level Specification of SES

In theory the rules in a traditional production rule based expert system are independent objects and the rule base is supposedly unstructured. Evaluation of the function of the rules used in the early prototypes for SES showed that in SES there was a great deal of structure in the knowledge base. Some structure is explicitly imposed by the use of meta rules that control the order in which the rules are executed. Structures are also implicit in the ordering of the rules, and the use of object names. Further, structure is implied by the patterns of groups of rules. Using groups of related production rules to model objects may be effective, but systems developers should be aware of the underlying objects and the interrelationships which are being represented. It is essential that the specification identify these items explicitly even if they are redundant to the resulting implementation of the system.

Specific Inference diagrams can be used to describe a domain by mapping out the relationships between features and identifying the specific inference sequences used by the expert. They often help to identify the types of abstraction and matchings that occur, as well as indicating objects and characteristic relations within the domain. Specific inference diagrams can then be generalized to show heuristic matches between different classes and used to identify the data and solution systems.

5.3.1 Specific Inference Paths in the Soils Domain

A simple inference in the soils domain states

> " if kauri trees are growing on the site then the soil will probably show signs of podzolisation"

This inference matches the vegetation class to soil morphology. Figure 5.05 shows similar parallels can be made between climate and profile characteristics, parent material and profile characteristics etc. All of the parameters in a site descriptor can be used to infer possible soil morphological features.

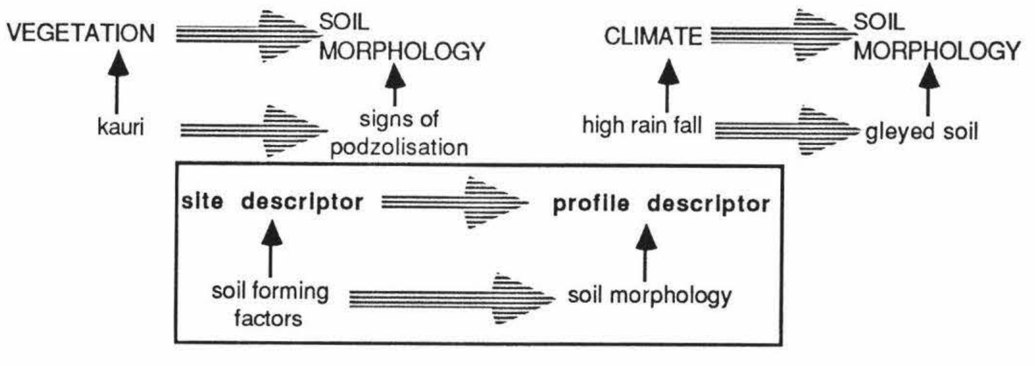


Figure 5.05 Generalizations from specific inferences

These simple diagrams do not show how the soil is going to be classified, which is the aim of SES.

A more substantial chain of reasoning leading to classification of the soil is shown in Figure 5.06. The inferences link the field observations about the soil to its generic classification, by identifying the features that characterize the classification of the soil. Generalizing this diagram shows the profile description made from field observations matching the typical classification description of the Paraha stony silt loam.

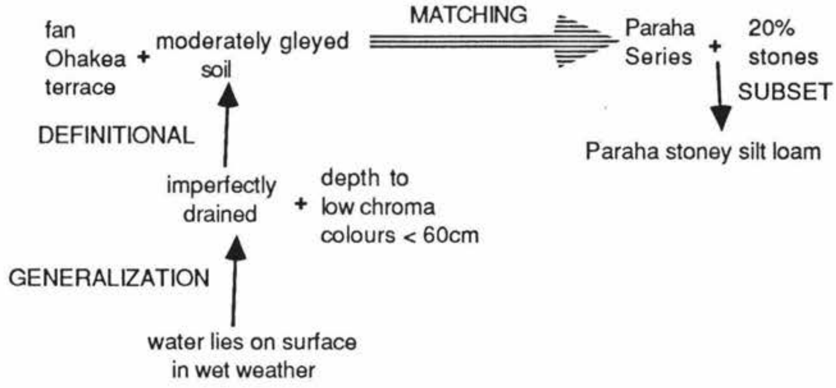


Figure 5.06 Specific inference diagram.

To aid in identifying the main systems of a domain it is important to identify the types of link leading to a solution. The heuristic link in figure 5.06 is partly causal and partly associated with the physical model for soils in the Horowhenua County. The other links can be classified as:

1. Definitional - if the soil is imperfectly drained and has low chroma colours higher than 60cm in the profile then it is described as moderately gleyed.

2. Qualitative - if the soil is a member of the Paraha series and contains 20% by volume of stones then it is a Paraha stony silt loam.

3. Generalization - if the water lies on the surface in wet weather then the soil is imperfectly drained.

As each item of field data is added on the left hand side of the diagram, the character of the soil is more precisely described. On the right hand side the series descriptions contain sets of values and it is necessary to use further knowledge from the data descriptions, to identify the soil type.

5.3.2 A Generalized Inference Structure for SES

From the generalization of the specific inference paths three levels of heuristic links between systems have been identified. One linking individual systems that reflect the soil forming processes such as vegetation to soil morphology. In this case there are a series of parallel systems (figure 5.07). Another linking soil forming factors to soil morphology. And the third linking field observations directly to the typical soil descriptors for each of the units in the classification system .

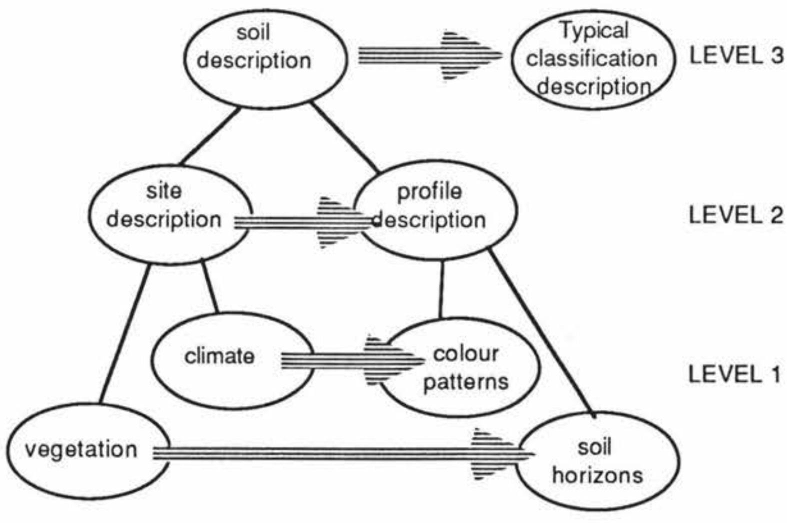


Figure 5.07 Levels of heuristics matching in the soils domain.

The links identified in the first two levels are both reflected in the soil description The first level is a break down of the parts of the site descriptor which reflect a range of different aspects of the profile descriptor, e.g. climate and vegetation. The simple soil-function equation discussed in section 1.2 formalizes the causal and incidental relationships between the soil forming factors and soil morphology. The typical soil descriptions not only record the expected values for features of a specific classification unit they also indicate the processes that influence the development of such soils.

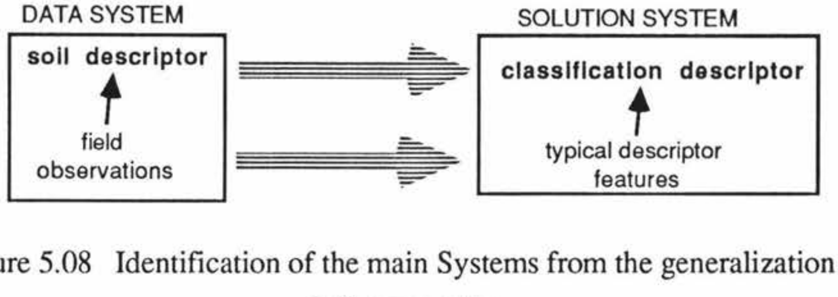


Figure 5.08 Identification of the main Systems from the generalization of inference paths.

The process of generalization is to identify a level of inference that generalizes the whole of the domain. Only the third level of match includes all the systems of interest in the soils domain, (figure 5.08). On the left hand side data is being abstracted to form a soil description. On the right hand side the data is used to refine the classification unit to which the soil belongs. There is a difference between the rules dealing with data abstraction and those dealing with solution refinement. Each abstraction step narrows the number of possible classification units that the observed soil can belong to, at a high level in the classification hierarchy. The rules applying to solution refinement identify the subclasses in the hierarchy that more closely match the actual observations.

The soils knowledge can be broadly characterized as in figure 5.09, which shows that the problem solving method in the soils domain is one of simple heuristic classification. There are a manageable number of predefined alternative solutions for the system to choose from, once the input data has been abstracted.

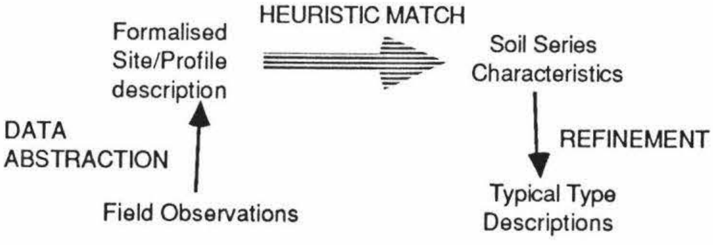


Figure 5.09 Generalized inference structure of knowledge about Soil identification.

The refinement of the solution down the units of the classification hierarchy uses the typical descriptions of the dominant soil forming factors for each unit. These typical descriptions highlight features that are definitional in nature for that particular unit. A soil's field description may almost match a typical description for a classification unit but for the inclusion of an exception. Where a normally insignificant factor is critical to the definition for a soil unit such an exception may place the soil into another classification unit.

5.4 The Domain Objects

In the prototype of SES the field data the user has provided is stored as objects in a dynamic data base. These objects are in the form of object-attribute-value triplets. An inspection of these structures shows that there is a hierarchical relationship between the objects, figure 5.10. This hierarchy is represented in the prototype by the names used for the objects, e.g. [a-horizon, consistency, friable].

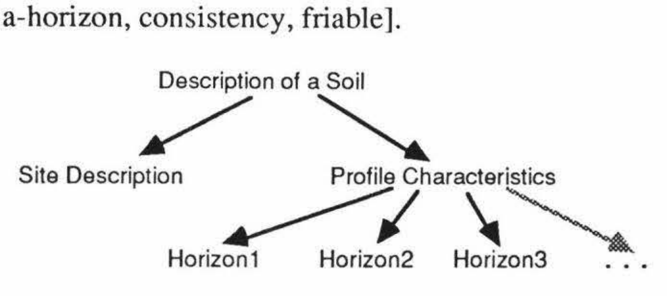


Figure 5.10 Hierarchy for soils data.

The early prototypes used rules for describing the domain knowledge but much of that knowledge reflects similar hierarchies to the field data. The form of the domain knowledge was not obvious in the rules. If a system is to be well defined, objects and implicit relationships between objects need to be represented explicitly, both within the identified systems and between them. As discussed in section 5.1.2 semantic nets provide a useful tool for developing a conceptual model of the objects and structures within the soils domain. The term conceptual model is used loosely here to mean the diagrammatical representation of the objects and relationships, that reflect a view of the underlying conceptual model of the domain.

The conceptual model for SES has been developed using modified forms of semantic net diagrams. The semantic nets have been used to describe both:

1. The object level schema.

2. Specific instances of the schema.

The features described in formal descriptions by pedologists can be characterized by a two dimensional semantic net structure. These dimensions represent the generalization and aggregation planes. The generalization plane describes the subset relationships between the classification units such as between soil series and soil types. The aggregation plane forms the description of a soil. As well as these two major dimensions there are informal hierarchies that can also be described using semantic nets and could be viewed as partial parallel systems. Partitions can be used for grouping specific objects and for linking different structures together.

5.4.1 The Generalization Plane

The classification system for mapping units is an is-a hierarchy and forms the basis of the generalization plane, figure 5.11. Unlike most hierarchical classifications where an object is a member of only one classification sub tree (a mammal can't be a fish) a specific soil may have dominant features that indicate that it is a member of more than one classification unit. Where these factors both contribute significantly to the description, then the soil is classed as an intergrade between the competing units. In most is-a hierarchies there is a one-to-many relationship between parent and child objects. In the soils domain the existence of intergrades means relationship between series and groups is many-to-many.

To be able to display the significance of the intergrade relationship in the diagram the semantic net notation used to describe the generalization plane has been modified. This modification is based on the common use of arrows in data base conceptual modelling where the arrows are used to indicate the ordinal value of a relationship. A one to one relationship is represented by a line while a many to many relationship is represented by a double headed line. A single arrow indicates a one to many relationship.

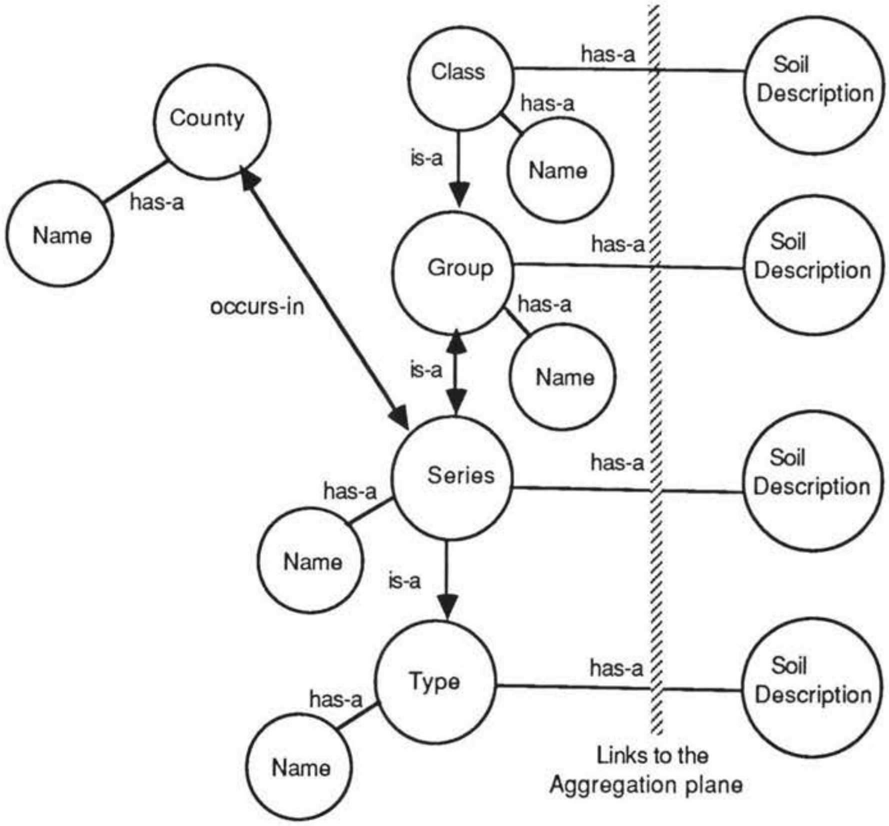


Figure 5.11 Semantic Net of the Schema for the Generalization Plane.

The occurs-in relationship is also important in the generalization hierarchy and records the geographical locations of soil series. This relation is not part of the formal classification of a soil but New Zealand soil maps are generally based on county boundaries. The general user perceives soil divisions by county. Counties also provide a convenient division for individual soils domain knowledge bases. The Counties only have a subset of soil series associated with them but specific series occur in more than one county. A further complicating factor is that some soil series that have very similar characteristics have different names depending on geographic location or climatic conditions. This information can not be shown in the schema diagrams but would be significant for a New Zealand wide soils expert system.

The links to the aggregation plane are has-a relationships. Each soil classification unit: class, groups, series and types are described by a typical soil description. In the generalization plane the soil description can be viewed as a single object which like the unit's name, can be viewed as an attribute of the main entity.

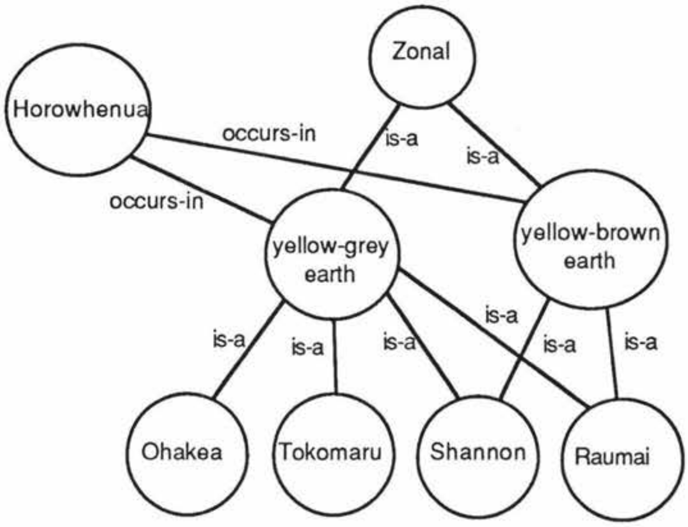


Figure 5.12 Specific Semantic Net showing part of the Soils Classification Network.

In figure 5.12 the Shannon Series and the Raumai Series of soils are shown as being members of both yellow-grey earths and yellow-brown earths. Diagrams of specific instances of any of the classification units can be drawn, but lists of the membership for each classification unit may be more useful in specifying the domain knowledge.

5.4.2 The Aggregation Plane

The aggregation plane describes the different parts of a soil description. Many of the objects in this hierarchy can be termed entities which are associated with a set of objects that can be viewed as attributes. These in turn can be regarded as entities by other parts of the system. The attributes are analogous to slots in a frame system. The aggregation plane is therefore represented by a number of diagrams at different levels within the hierarchy. The root diagram, figure 5.13, details the top level of a soil description and is linked to the generalization plane via the has-a relationship.

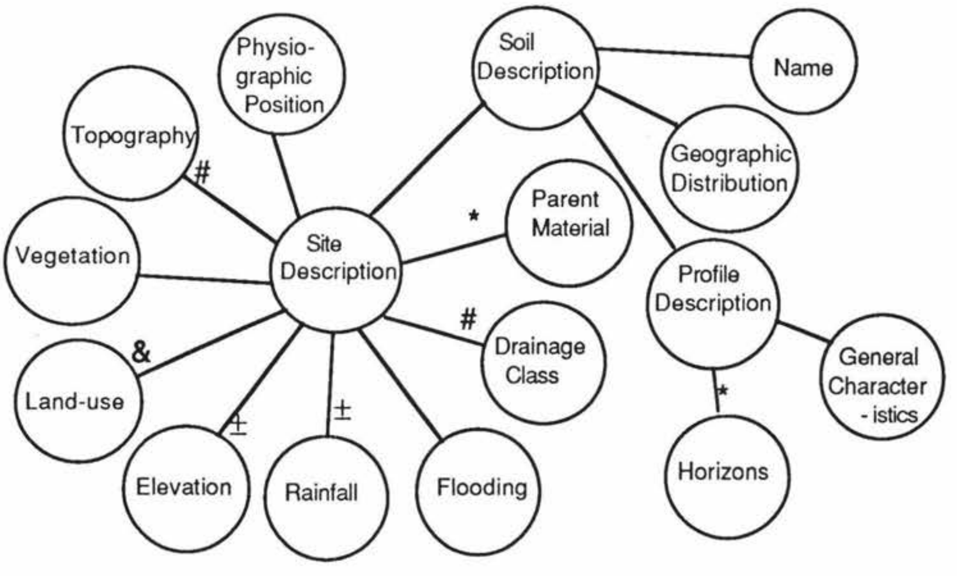


Figure 5.13 Semantic Net of the Schema for the Aggregation Plane.

Most of the relationships used in the representation of the aggregation plane are simple part-of relationships but there are a number of modifications to this relationship which are designated by specific symbols, figure 5.14.

Relation	Symbol	Description
RANGE	#	A list of values that represent points on a continuum defining a set of values an object may take.
VALUE RANGE	±	The values that the object may take are numerical units of measurement. The range defines a set of values the object may take.
CONDITIONAL	§	This defines a conditional dependency. The value taken by the object depends on the value of the object being pointed to.
MULTIPLE	*	One or more repeats of the object.
LINK	@	A link structure can be created with a dominant descriptor linked to a subordinate descriptor with the same structure.
ALTERNATIVE	&	More than one version of the object can describe the object.

Figure 5.14 Key to symbols used to designate specific relationships used in the aggregation plane diagrams.

These symbols can be divided into two kinds:

1. Those that partially define the type of values an instance of the object may take - range, range value and conditional.

2. Relationships that indicate different types of multiple copies of an object - multiple, link and alternative.

The soil description has two major parts identified when specifying the inference structure. These are:

1. The site descriptor which identifies the soil forming factors.

2. The profile description, which is a description of the morphology of the soil.

The main object for describing soil morphology is the horizon. Horizons are the layers that can be seen in profile, a vertical section through the soil. The lay person can usually recognize at least two main divisions, the top soil and the sub soil. Any soil has a minimum of two horizons and more commonly at least three. Each horizon can be described by up to twelve common features, figure 5.15.

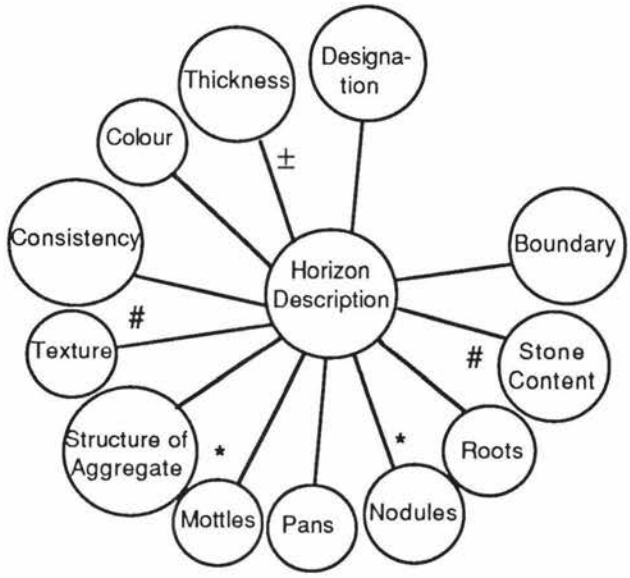


Figure 5.15 Semantic Net of the Schema for the Horizon Descriptor.

Each of these features may in turn be described by another level of semantic networks. Figure 5.16 shows both the schema and a specific instance for the structure of a soil aggregate. The schema diagram shows a link object. The link relationship is necessary for descriptions where there is more than one type of aggregate. The example shows the structure corresponding to the following written description:

> " a moderately developed fine subangular blocky structure breaking to a moderate granular structure "

The conditional relationship between size and shape is necessary as the qualifier fine has different meanings depending on the shape class of the aggregate. A fine block structure is between 5 and 10 mm in diameter while a fine granular structure is between 1 and 2 mm in diameter. In the sub-structure description the link object and relationship have been left out because they do not apply in this case. Any specific instance of a schema may exclude some objects, conversely some of the objects are mandatory. This is not shown in the schema diagrams as it is considered an unnecessary detail.

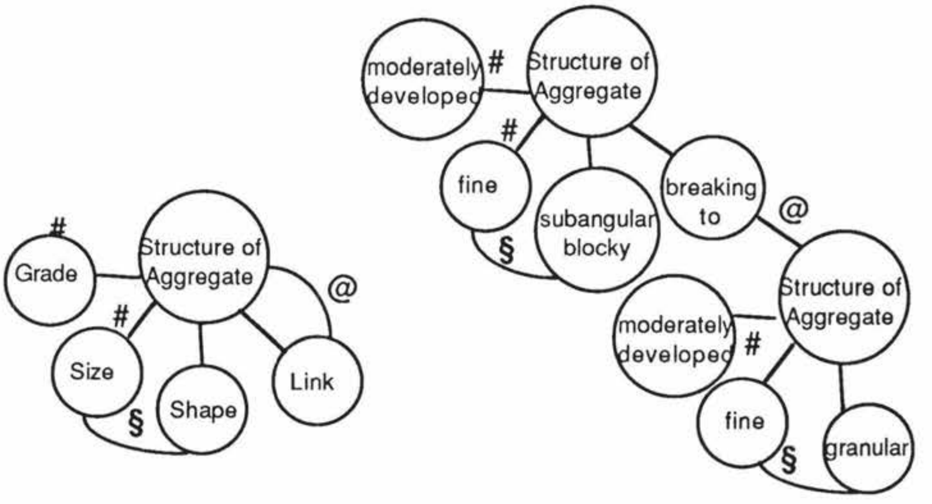


Figure 5.16 Semantic Nets showing the schema and a specific instance of the object describing the Structure of an Aggregate.

The soil description hierarchy, as already mentioned, can be used not only to describe the incoming field data but also to describe each of the classification units from the generalization plane. Only the main associations between objects are represented in the diagrams of the aggregation plane, as in the generalization plane . Soil descriptions have a large number of variables but only a subset of these are relevant to any specific soil description.

The representation of the schema for a soil description will contain all the possible descriptors required to fully describe any one of the classes occurring in the knowledge base or that may be used to represent the field data. The main differences between the descriptions of the different classification units forming the solution systems are:

1. in the types of attribute that are significant

2. the range of values defined as acceptable for an attribute

The descriptions for soil units in the higher levels of the hierarchy will only include the salient features and will usually define a set of valid values for each feature. A soil type is a subset of a soil series: e.g. the Paraha silt loam is a member of the Paraha series. The soil series descriptions are a general description which defines the sets of permissible values for the essential attributes of the member soil types. The values defined in the series descriptions may be inherited if a value has not been defined specifically for any soil type within the soil series.

5.4.3 Informal Classifications used in the Soils Domain

The descriptions of the generalization plane and the aggregation plane are based on the formal methods used by the pedologist to describe and classify soils. A number of semi-formal classifications are also used by pedologists. These classification systems are used for determining which subset of soil series that a specific soil may may be provisionally classed in before more detailed evaluation. They form alternative levels to the generalization plane. These alternative networks represent important divisions in New Zealand soils. By specifically representing them in the conceptual model the importance of these soil features is highlighted.

Partitions are a way of regrouping objects across the normal structures in the semantic net and allow links into the alternative hierarchies to be set up. New Zealand soils can be quickly grouped, based on the geomorphic features on which they occur. This information is recorded in the soil descriptors but by defining partitions the groups of soils with similar characteristics are more readily identified.

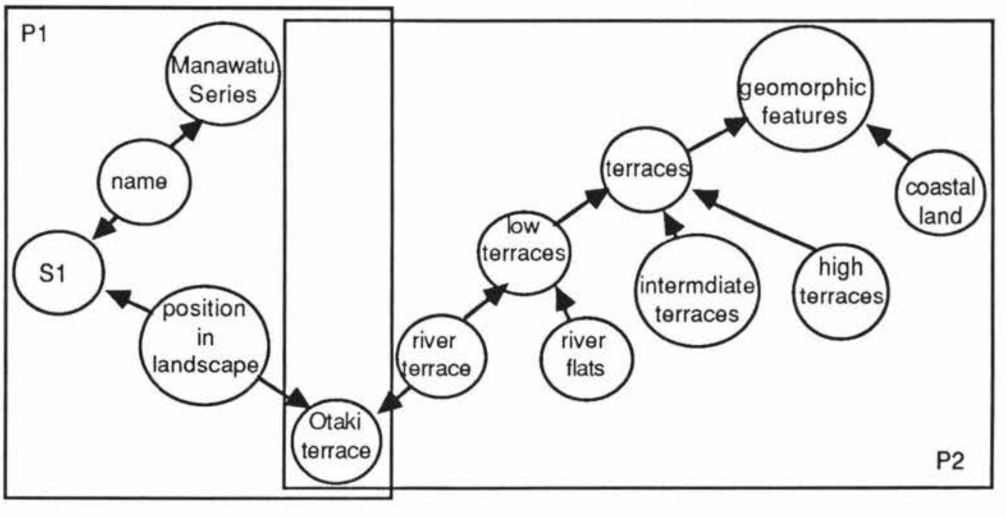


Figure 5.17 Semantic net showing two of the partitions the node "Otaki terrace" participates in.

Partitions can provide a focussing space (Rich, 1985) for viewing the data from different perspectives. In figure 5.17 the node "Otaki terrace" participates in two partitions, the left-hand partition representing soil series descriptions and the right-hand partition representing geomorphic features. If the geomorphic feature is established a subset of possible candidate solutions for classification of a soil is defined.

Partitioning can also be used by the explanation module and in providing information for clarification when communicating with the user. If for instance the user did not know which specific terrace the data came from the system could use the geomorphic hierarchy to pose the question "Is it on a river terrace ?" or moving further up the tree. "Is it on one of the lower terraces?". A partition linking "river terrace" into information about the requirement for a river or stream to be nearby would produce a better series of question.

Not all soil features need to participate in partitions. Only features significant for a group of soils in a region, such as the drainage characteristics of soils in Horowhenua County. Partitions can be used to provide links based on specific values of such features when there is no hierarchical network associated with the feature.

Another important use of partitions is focussing on special relationships such as soil associations and soil complexes. The association relationship, figure 5.18, refers to soils that occur in the same geographic locations.

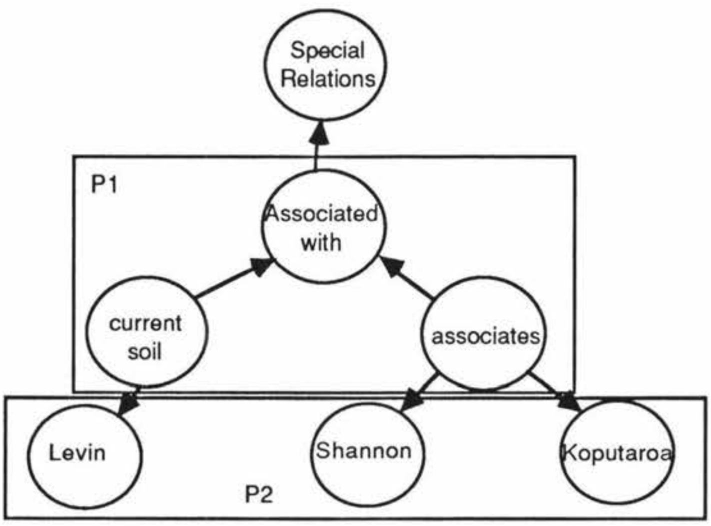


Figure 5.18 Semantic net with partitions for the special association relationship

5.5 Functional Aspects

Ready acceptance of the prototype expert system as a possible tool for helping to identify soils, occurred because the interface and the pattern of questioning was familiar to the expert. The functional aspects of the specification should reflect the methodology and vocabulary used by the expert. During the knowledge acquisition process the knowledge engineer and the domain expert should be able to outline the strategies used during the problem solving process.

The soils domain expert's main strategy was outlined in section 4.2.1. This outline was used as the basis for preparing a series of simplified state specification diagrams. The first diagram, figure 5.19, outlined a straight forward simple classification session.

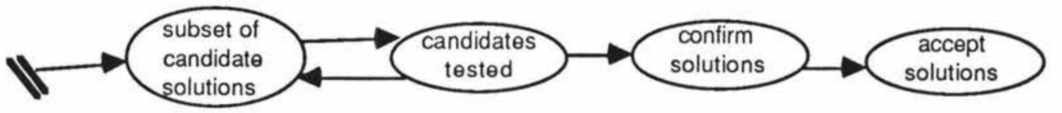


Figure 5.19 Initial Simplified State Specification Diagram for SES

Using the initial diagram as a basis for further discussions, the diagram was modified, figure 5.20, to show the states required to deal with situations where:

1. The field data was insufficient to determine a set of possible solutions.

2. The initial data did not clearly indicate a suitable solution set.

3. Conflicting solutions are compared and contrasted with the original data and each other, to determine more clearly the solution set.

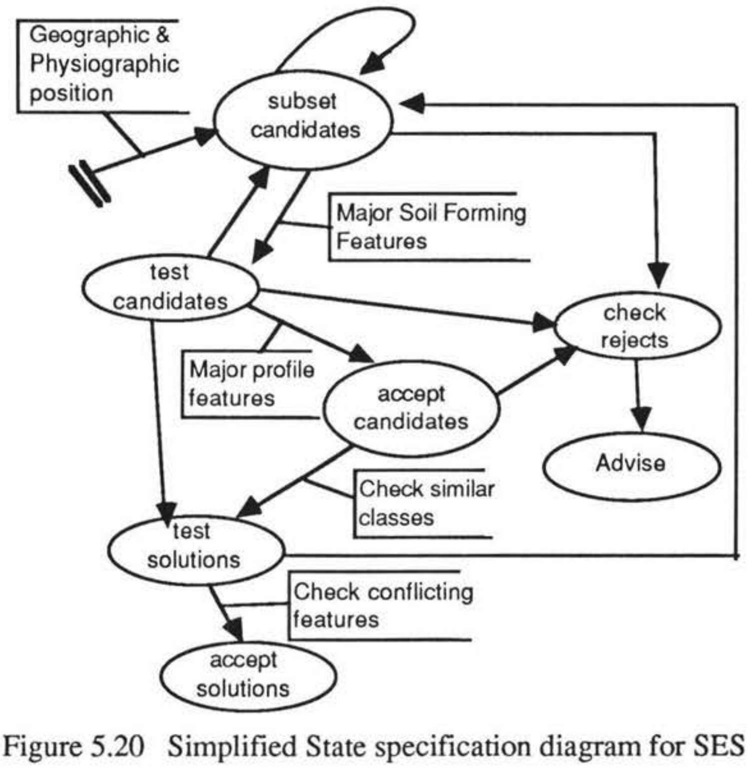


Figure 5.20 Simplified State specification diagram for SES

At the same time as the discussions that led to the simplified state specification diagram were taking place the taxonomic unit sheets which summarize the field data from the soil survey of Horowhenua County were also examined. From these two sources, notes on the strategies used to move between states were made with the purpose of identifying:

1. The intermediate steps between states.

2. The conditions necessary to both initiate and complete the steps.

3. The operations required.

When developing an expert system in a commercial environment the descriptions should be carefully documented and filed, but even rough notes and jottings are sufficient if not ideal. It is the resulting state specification diagrams that become part of the specification.

A description should be written for each strategy line on the simplified diagram. These textural descriptions are analysed and the completed State Specification Diagram developed (Appendix F). This diagram is too detailed to provide a base for discussion with the domain expert but provides a quick overview for the knowledge engineer. On this diagram a shorthand notation has been used for the benefit of the knowledge engineer. The domain expert does not think in terms of 'certainty values less than two' but rather that 'there is not significant evidence', indicating the current solution unit is not a possible solution in this case.

The functional analysis proceeded in parallel with determining user and module requirements. Therefore some of the ideas under consideration were incorporated into the prototypes being developed. The prototype exploring the user interface used the pattern of the most significant features of Horowhenua County soils to cut the initial search tree. The main access into the graphics data base was also based on this work. The other significant result was the development by the experts, of a simple decision tree for determining soil group classification. This decision tree was used to teach students how to quickly decide which group, a soil could be classed in. This decision tree was used as the base for the development for the knowledge base for the explanation prototype.

5.6 Summary

Knowledge acquisition is not a simple task. Specifying the knowledge acquired in a form that is understandable, useful and complete is correspondingly difficult, if not impossible. In section 5.1 different aspects and different methods of specifying this knowledge were discussed. One point that needs to be emphasized is that for the resulting system to be acceptable to the expert and the users, it is essential that the questions and the terminology be familiar. This must be reflected in the specification tools used.

Recently a number of authors (Johnson, Zualkernan and Garber, 1987; Hayward, Wielinga and Breuker, 1987; LeFrance, 1987) have presented methodologies for specifying expert based systems. These papers all acknowledge the role of cognitive science in the knowledge acquisition phase of expert system development. The methodologies discussed have adapted one or more models used by cognitive psychologists for describing how the human brain works.

The methodology outlined by Johnson et al (1987) is useful not only for identifying the expert's problem solving methodologies but also the objects their relationships and the inference structures. Ideally every interview should be taped and then transcribed. It is important that the knowledge engineer takes notes during taped discussions as relevant information is often visual. These notes should be amalgamated into the transcripts which can then be analysed. A more detailed discussion of the role of interviewing techniques for knowledge acquisition is given by Hoffman (1987).

Three different aspects, or levels, were identified as necessary to specify the knowledge base.

1. The Inference Structures
2. The Object/Relationship Structures
3. The Functionality

All three aspect of a knowledge base need to be described to provide an adequate specification for developing an expert system. These aspects are similar to those identified by Hayward (1987) who suggests that expertise be viewed from a number of different perspectives. His four level model of expertise specification can be matched to the three levels outlined in this chapter.

1. The inference level matches the inference structures
2. The domain level matches the object/relationship structures
3. The task and the strategic levels matches the functionality specification.

Each of the three levels identified models a different view of the domain expertise. The solution systems as identified by the inference structures in the soils domain are modeled using semantic nets. Whereas the inference diagrams were analysing the patterns of heuristic matching the part-of semantic nets are describing the structures in the different solution systems. The state specification diagram details how the expert uses the inferences and the objects to come to a conclusion during a consultation.

Chapter 6: The Formalization Stage

The formalization stage is similar to the traditional system design phase. As in the conceptualization phase, formalization concentrates on knowledge base design. At this stage the conceptual design is mapped onto a specific formal representation. The representation used (rules,semantic nets, frames etc) may be dictated by the expert system tools that are to be used during implementation. Ideally, the representation chosen should be the one that most closely matches the structures naturally occurring in the domain.

The first step in the formalization phase is to identify the kinds of domain-specific knowledge the expert system will require and the types of structure necessary to represent this knowledge. The development of CENTAUR (Aikins, 1983), which used knowledge from an existing expert system, highlighted the kinds of knowledge which should be explicitly stated in the design of a system. Examining the models of existing systems is one method which will identify the types of knowledge and suitable knowledge structures required in an expert system.

In early expert systems, such as MYCIN and PROSPECTOR, the formalization stage of matching the conceptual model to the chosen tool or framework was embedded in the implementation phase. Implicit conceptual models evolved as the appropriate data structures were chosen or developed to meet the system's requirements. An examination of these structures, as described in Davis (1978), will reveal the conceptual models underlying the systems thereby identifying the way the conceptual knowledge has been implemented.

Details of the different types of structure used to represent knowledge in existing knowledge bases are surveyed in this chapter. The three knowledge bases examined are those used in MYCIN, PROSPECTOR and CENTAUR. The knowledge base structures used in each of these three systems are explained by using examples from the soils domain. This approach was taken so that the suitability of the structures for representing knowledge about soils could be better evaluated.

6.1 Knowledge Structures in MYCIN

There are five basic structures for storing knowledge in the MYCIN system. They are :

- rules
- object-attribute-value triplets
- the context tree
- lists
- tables
- functions

Rules are used to represent the static heuristic domain knowledge. Object-attribute-value triples are used to store the dynamic data. This information is either entered by the user or deduced by the system. The context tree is a control structure of the system that enables the rules to refer to the correct piece of information, i.e. to function in the correct context. Lists and tables are used to store the factual knowledge about the domain.

```
(   OBJECT       ATTRIBUTE    VALUE          CERTAINTY       )
(   'A horizon'  texture      'silt loam'      0.8     )
(   series       identity     [ (Manawatu      0.6 )
                                (Rangitikei    0.5 )
                                (Kairanga      0.3 ) ]   )
```

Figure 6.01 Object-Attribute-Value triples

An object-attribute-value triple (figure 6.01), has three positions for storing information. The object indicates an identifiable physical or conceptual entity within the domain such as a soil horizon or a site descriptor. The attribute is a property or feature of the object such as texture, colour or topography. The value position stores the specific value of the attribute for the named object. Although this structure is referred to as a triple, when uncertainty is dealt with by the system a fourth position is used to store the certainty value associated with the triple. The structure can be modified to store lists of values with their corresponding certainty factors. This modification is normally used when the identity of some entity is being pursued as shown in figure 6.01.

Rules in MYCIN use the object-attribute-value triple as the basis for both the premise and the action part of the rule. Conditions that make up the premise are evaluated using the functions such as known,same, similar etc. The format of a possible rule is shown in figure 6.02. Each clause in the premise part of the rule begins with a function name indicating how the system should evaluate the truth or otherwise of the clause. The evaluation uses the value of the certainty factors stored in the dynamic database. The function in the action part of the rule tells the system what to do when the premise is true. Associated with each rule is a reliability factor indicating the degree of belief the expert has in the hypothesis being true given the evidence.

```
RULE ##

Premise: (and   (known( profile,'A horizon')
                known( profile,'C horizon')
                thoughtnot( profile,'B horizon')
                or  (same('C horizon','parent material','non calcareous alluvium' )
                        same('C horizon','parent material','volcanic ash' ))))
Action:   conclude(classification,'soil group',recent, Tally = 0.7)
-----------------------------------------------------------------------------
IF:       1)    the 'A horizon' has been identified in the profile, and
          2)    the 'C horizon' has been identified in the profile, and
          3)    the 'B horizon' is not present in the profile, and
          4)    either the 'parent material' of the 'C horizon' is 'non calcareous
alluvium'
                or      the 'parent material' of the 'C horizon' is 'volcanic ash'
THEN:     There is suggestive evidence that
                the soil can be classified as a Recent soil.
```

Figure 6.02 Mycin like rule showing functions and O-A-V triplets with English like translation.

As well as the rules using the simple object-attribute-value structure used with the dynamic knowledge store, other rule structures make use of the static factual domain knowledge. These rules have corresponding special functions for comparisons between the case data and the factual knowledge in the knowledge base. The rules make use of both lists and tables. The tables are used to store information that is common to a group of entities. Clancey (1985) states that:

> " A single, general rule uses a table to identify the unknown organisms. These tables are also called 'grids'; we were unaware at the time (1974-1977) that we were recording the same kind of information AI programmers were storing in frame hierarchies. "

The objects referred to in the triples are the nodes in the context tree. A set of attributes is associated with each node type that can occur in the context tree thereby defining the allowable attributes that can be associated with each object in a triple.

During a consultation the system develops a number of subtrees that all link to a single head node. The system has a template for each of the node types that can occur in the context tree. These templates, figure 6.03, are discussed under data structures in Shortliffe (1984) and provide structures for creating new nodes in the context tree. These structures are a form of meta data. The descriptions of these structures provide a template for knowledge about the objects within the domain, associated procedural knowledge and the relationships between structures.

Instances of these structures are instantiated during a consultation and this includes specifying the link into the current context tree. Each main node or object referenced in the tree is termed a context. The attributes specific to the domain knowledge associated with each context in MYCIN are termed clinical parameters. Using data definition terminology a context is an entity and the clinical parameters can be viewed as attributes.

SNAME:	system defined name for this entity.
ANAME:	the name identified by the system from the act of diagnosis. This may be known like the patients name or may be deduced as in the case of a bacterium.
ASSOCWITH:	defines the entity's parent nodes.
TYPE:	identity of the type of entity
MAINPROPS:	List of the main values that must be asked for when the attribute is created. These values are not necessarily required by the system but they are the values a user would expect to supply when asked about the entity.
PROPTYPE:	list of the attributes associated with this entity type
SUBJECT:	lists the appropriate categories of rules to be used with this type of entity
PROMPT:	the question template used when requesting the creation of an instance of this type of entity.
SYN:	indicates how to reference the context in which questions are asked
TRANS:	the English translation used when an entity instance is referenced in a rule

Figure 6.03 Context Template

There are over 65 clinical parameters. Each of these parameters is represented by a structure or set of properties (figure 6.04). These structures are very like frames. There are at least 10 different properties of which only a subset is used for any one parameter type. The MYCIN workers identify three different types of parameter of which the multi-valued parameter is the most general. Five of the properties appear to be mandatary for all parameters: type, expect, lookahead, trans and prompt.

TYPE:	ident
EXPECT:	oneof (list of organisms known to the system)
LOOKAHEAD:	list of rules identified by rule number where the parameter occurs in the premise
TRANS:	the natural language translation to be used when this parameter is referenced in a statement
PROMPT:	the question to be put to the user when the value of the parameter is sort by asking
CONTAINED_IN:	list of rules identified by rule number where the parameter occurs in the conclusion
UPDATED_BY:	list of rules which the conclusion infers a value for this parameter
CONDITION:	one or more rules for deciding whether the user should be queried about this parameter

Figure 6.04 Parameter Template.

The context is identified by a system-defined name, such as "object-1". Although it is not an explicit property of the context the name or names used to identify the specific context are an implied property. Each context has associated with it a specific set of parameters which together represent an object within the domain of expertise. As well as these relationships between entities and attributes that define the domain model, further attributes are associated with both contexts and parameters to create a frame-like structure.

This knowledge can be viewed as a kind of meta knowledge because it is information about the domain knowledge.

An examination of the content of these structures reveals that they contain information or knowledge required by different modules within the expert system. Slots such as TRANS and PROMPT are used by the interface module and the slot SUBJECT by the inference module. Other slots contain information used by a number of modules. The EXPECT slot can be used by all four major modules:

1. The inference engine uses it to check the validity of an action or response.

2. The interface module can use the information for presenting the user with a list of alternatives in a menu.

3. The explanation module can use it for indicating the possible range of values a parameter can take and for locating further knowledge relevant to explanation but not for inferring a result.

4. The acquisition module will use the information to check whether a rule is using a known value or whether a new value must be added to the system.

The CONDITION slot behaves like a demon[1], (Rich, 1983). Whenever an object containing this slot is referenced the actions of the slot must be carried out first. Knowledge such as stored in MAINPROPS is unnecessary for the inferencing process but makes the system more acceptable to users.

6.2 Knowledge Structures in PROSPECTOR

The geological domain of Prospector is not only concerned with judgemental knowledge but also the relationships between objects. As pointed out by Duda et al (1978)

> " it might be natural to represent judgemental knowledge by a set of production rules, but unnatural to use the same mechanism to represent other relevant knowledge, such as taxonomic (subset/element) relations among objects in the domain. "

To represent both the judgemental and the object relationships applicable in the domain the developers of Prospector adapted the semantic network representation to

1 Rich defines a demon as "conceptually, a procedure that watches for some condition to becomes true and then activates an associated process" .

" retain the desirable modularity of a rule-based approach while permitting an explicit, structured description of the semantics of the problem domain. "

Duda et al (1978) outline the structures used in the knowledge base of Prospector. The main structures are:

- hierarchies of objects
- relationships linking objects
    - common relationships
    - relationship families
- partitions
- rules
- the inference network

Relationship families link specific entities with values. Common relationships form the links in the taxonomic structures used for classifying the domain objects. Partitions group together objects and relationships. Heuristic knowledge is represented using rules which can be linked together into an inference network.

The relationship structures are composed of the two main types of nodes used in semantic networks, subject nodes representing domain objects and relation nodes representing the relationships between different nodes. The statement "the A horizon overlies the B horizon" has two subject nodes with overlies being the connecting relational node, figure 6.05.

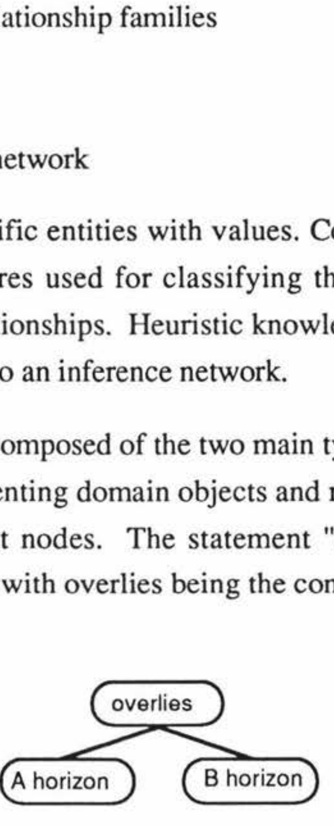


Figure 6.05 Semantic network showing the representation of the "overlies" relationship

The common relationships are represented by special arcs rather than nodes. The networks created by these common relationships form the taxonomic structures within the domain where the set/subset relationship is used. The element relationship links specific values into the network. Other common relationships used in semantic networks are: has-a , part-of, and is-a. The subset and element relationships are examples of the commonly used is-a relationship. A soils taxonomic network (figure 6.06) shows the relationship between soils in Horowhenua County based on major physical features.

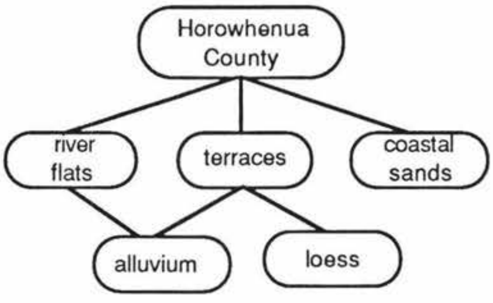


Figure 6.06 Network showing subset relationships based on a physiographic division of soils.

As well as the common relationships there are a number of more specific relationships that have been termed relational families. These relations are mainly used to link objects identified during a consultation with the static facts in the knowledge base building up the dynamic database. The special relationships more closely reflect the special nature of the domain knowledge.

Figure 6.07 shows the special relationship "composition" being used to define the possible values an entity may take. It can be used to represent the fact that the parent material of a soil has been identified as volcanic tuff. This specific instance of the special relationship composition becomes an assertion in the dynamic data base. Another use of special relations is to show important dependencies between instances of entities. The valid descriptions used to describe the size of a soil aggregate depend on the type of aggregate that has been identified. Here the relation is being used as a validation check.

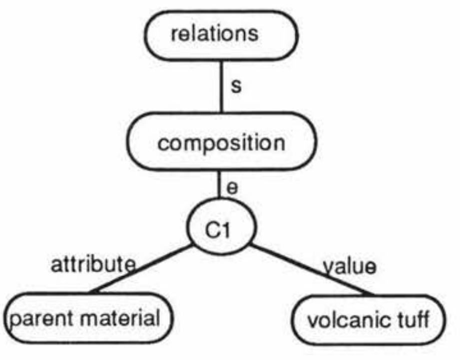


Figure 6.07 Net showing that composition is a relation that has an instance linking parent material to volcanic tuff.

Partitions are used to define the members of the sets of relational families. The relationship between a named relation and instances of the relation is called a delineation. The partition provides a reference for specific instances of a relation separating member nodes from their associated network structures. This allows for some of the nodes to act as variables (figure 6.08) that can be instantiated during a consultation.

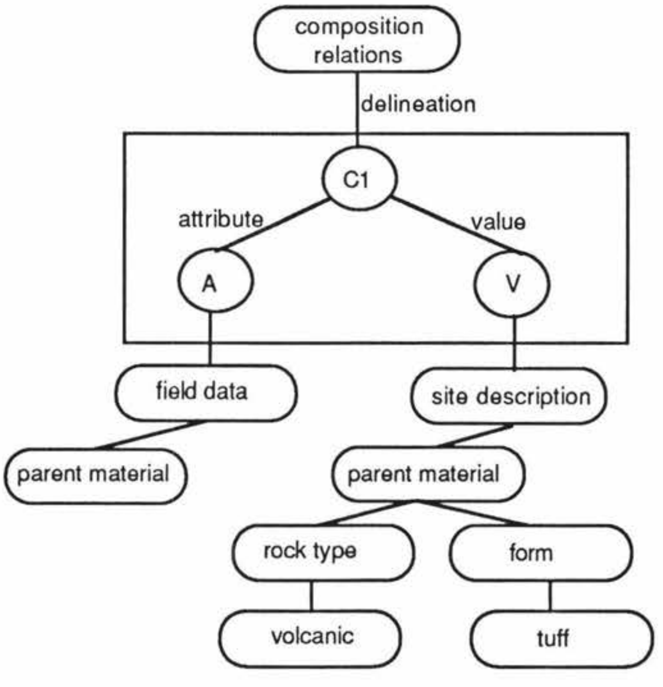


Figure 6.08 The nodes within the box form the partition for the delineation of the composition relations.

Rules are also defined by using partitions. The partitions define the limits of the antecedent and consequent parts of a rule. The partitions contain the network structures that represent each rule. The rules themselves can be classified using a further network. The premises in the antecedent partition of a rule consist of inter-related instances of relations.

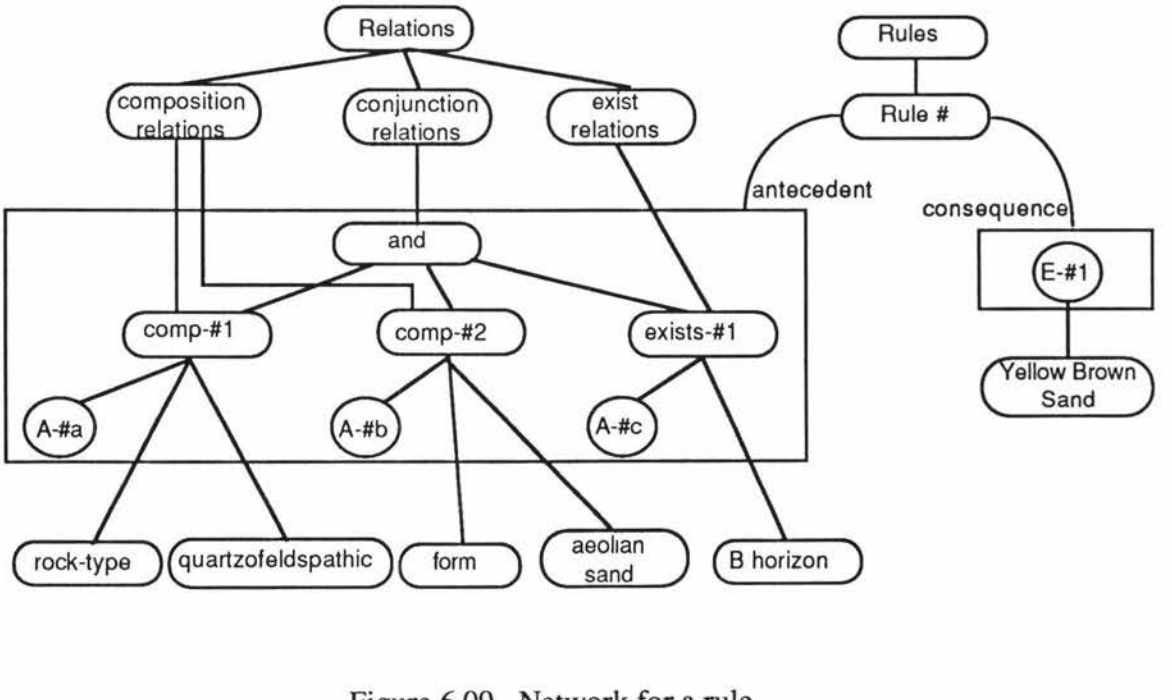


Figure 6.09 Network for a rule.

Associated with each rule node is a property list including the measures of necessity and sufficiency indicating the strength of the rule. The metrics used in Prospector for dealing with uncertainty are based on Bayesian Logic. Property lists are also associated with each instance of a consequence partition recording the prior probability and slots to maintain posterior probabilities as the evidence is accumulated.

The rules form inference networks of assertions that lead to the main domain hypotheses. Duda and Gaschnig (1981) identify three different kinds of relations that effect the probable truth of the hypothesis and interrelated assertions. These are:

- Logical Relations
- Plausible Relations
- Contextual Relations

Each of these relations has to have associated with it a procedure that is used by the inference module to calculate the strength of the hypothesis under consideration. Logical relations connect assertions using the primitive boolean operations AND, OR and NOT. The following rule depicted in figure 6.09,

> " if the parent material is quartzofeldspathic aeolian sands and there is a distinct B horizon then consider the possibility that the soil type is a yellow brown sand"

has three special relational assertions linked by the logical relation AND. Plausible relations result when more than one rule contributes to the truth of the same hypothesis. Contextual relations impose order on the evaluation of assertions where the possible truth of one assertion is dependent on the truth of a related assertion. The contextual relationships are unnecessary for the system to reach a conclusion but they ensure that the user is queried in a sensible order.

6.3 Knowledge Structures in CENTAUR

The CENTAUR system (Aikins, 1983) is related to the MYCIN experiments. The design of CENTAUR is based on the premise that

> " the chosen structure(s) must be expressive enough to represent a variety of types of knowledge explicitly; that is the system should have direct manipulatory access to the knowledge as opposed to having knowledge 'built-in' "

This system uses a hybrid of knowledge representations that allows object knowledge to be explicitly stated. The main structures are:

- frame like structures:
    - prototypes
    - components
- tables
- control structures
- rules
- a contextual hypothesis network
- an agenda

The frames are used to represent the expected patterns of data that occur within the domain and the values of the individual components of the data. Tables are used as an efficient method for storing condition-action pairs. The rules are divided into different types depending on their function and are associated with individual slots within the frames. The hypothesis hierarchy provides a contextual structure which explicitly identifies the linked frames relevant to the current consultation. The dynamic data is stored within the frames as they are referenced and the agenda provides a stack structure storing the current state of the inference system during a consultation.

Frames and slots model the instances of entities and attributes identified within the domain. Two kinds of frames are used in CENTAUR, the prototype and the component. The prototype frame provides a stereotypical description of a domain object. The prototype frames, model entities that are important within the taxonomic hierarchies of the domain, such as the hierarchy grouping different kinds of soil based on generic classification of soils (figure 6.10).

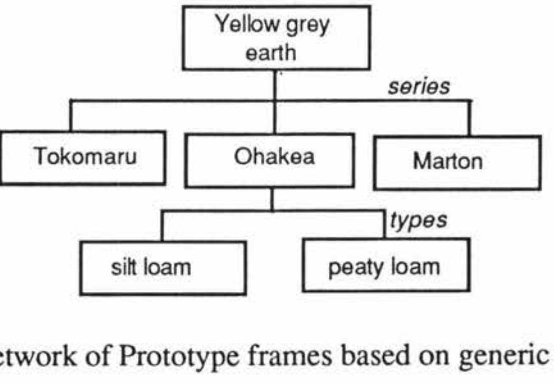


Figure 6.10 Network of Prototype frames based on generic classification.

The slots in a prototype (figure 6.11) contain object-level knowledge or attributes as well as meta level knowledge. The meta-level knowledge includes specific control knowledge relevant to the particular prototype frame and task dependent knowledge in the form of rules. The third type of meta-level knowledge is general knowledge including book-keeping information and the information about the network hierarchy links.

Frame:	PROTOTYPE
Name:	YELLOW GREY EARTH
Hypothesis:	Soil belongs to the Yellow Grey Earths
More General:	(County MANAWATU)
More Specific:	(Series TOKOMARU)
	(Series OHAKEA)
	(Series MARTON)
Too fill in:	physiographic position
	parent material
	B horizon general description
	drainage
If confirmed:	clay content
	pans
	colour
	mottles
	gleying
	if hypothesis certainty > 4 then target SPECIFIC Frames immediately
If disconfirmed:	if physiographic position OK then target INTERGRADES
Components:	physiographic position
	parent material
	B horizon general description
	drainage
	. . . all other attributes referenced in earlier slots. . .

Figure 6.11 Prototype frame showing conditions to be met for a soil to be a Yellow Grey Earth.

The object-level knowledge associated with each prototype frame is stored in individual component frames (figure 6.#12). The components associated with each prototype frame represent the features that characterize that prototype frame i.e. the main attributes. The type of knowledge stored in these component frames is related to the possible range of values the object may have. Each component frame has at least one slot to record its Importance Measure. This metric ranks the attributes represented by the component frames in relative importance as a characterizing feature of the entity represented by the prototype frame.

Tables are used to define condition action pairs when a value for a component is required. They outline the specific types or kinds of association the known value can have with the recorded prototypical value and the action to take when a match occurs. Typically associations defined in these tables are comparative.

The control structures occur in every prototype frame and direct the system on how to validate each prototype frame (figure 6.11). They are the action part of a rule. Four kinds of control rule are identified in CENTAUR:

1. to-fill-in : actions to be taken to instantiate the prototype frame

2. if-confirmed : actions to be taken after the instantiation of the prototype frame

3. if-disconfirmed : actions to be taken if instantiation fails

4. action-slot : actions after final conclusions have been derived

The control structures enable the implementer to specify appropriate strategies for instantiating each individual prototype frame. The to-fill-in slot can indicate which other slots must be instanciated and which rules must be applied indicating that a sufficient condition has been reached for a match to be recorded.

Frame:	COMPONENT
Name:	physiographic position
Pointers-General:	(Group YELLOW GREY EARTH)
Importance Measure:	4
Plausible Values:	OHAKEA terrace
	TOKOMARU terrace
	MARTON terrace
Possible Error Values:	River terrace
	if river terrace then target RECENT SOILS
Default Value:	TOKOMARU Terrace
Inference Rules:	if on a terrace and too high to be flooded and close to ranges then suggests TOKOMARU Terrace
	. . . further rules for confirming terrace location. . .

Figure 6.12 Component frame defining physiographic features characteristic of a Yellow Grey Earth.

Rules other than control rules fall into three main groups depending on their function in the system.

1. The inference rules are used to infer the values of specific components. These rules are found in component frames.

2. The trigger rules are used to determine what further prototype frames should be invoked.

3. The last group of rules are associated with specific stages in the consultation process. They are invoked after the active prototype frame network has been established and are use to confirm and summarize initial tentative conclusions. These rules appear to check the necessary conditions for a match to be confirmed.

One type of rule that appears to be missing is the comparison rule. The type of rule that compares the values of two or more slots. This type of heuristic may consider

"if the texture of the A and B horizons is similar then"

and is often important in the soils domain for quickly identifying whether a prototype frame match can be made.

The prototype frames are linked together in a hierarchical network that allows for inheritance and also defines the context in which the system is working. As the consultation proceeds an active network is built up of the prototype frames that have been confirmed. This active network defines the tentative conclusions of the first stage of the consultation process. The tasks to be carried out for the current prototype frame are placed on an agenda. The agenda is a LIFO structure used for keeping a record of the current state of the consultation.

6.4 Summary

All three systems surveyed can be described as simple heuristic classification models (Clancey, 1985). SES can also be described in terms of the simple heuristic classification model. Consequently, similar structures should be appropriate for storing soils domain knowledge. Each of the three knowledge representation methods examined has advantages and disadvantages for organizing soils domain knowledge.

The main method for knowledge representation in MYCIN is the rule. The rules identifying organisms are re-used for each of the organisms in each of the cultures associated with a consultation. MYCIN registers which organism in which culture a rule is refering to. The context tree controls the execution of the consultation. It records the relationships between the main objects of interest thereby identifying the organism and culture a rule is referencing.

The way the context tree is used in MYCIN is inappropriate for the soils domain because the structure of the MYCIN context tree does not match the structures used to describe the soils domain. In the soils system a profile may have many horizons but exact identification of an horizon is not usually critical. The identity of an horizon can normally be inferred from its relative position in the profile. Soils rules make direct reference to the specific horizons as required.

A soils system still requires explicit knowledge about the relationships between domain objects although the identity of the context is known. The concept of entities and attributes being represented by frame-like structures or templates (figures 5.03 and 5.04) is applicable to the soils domain. In the medical domain the context tree represents the aggregation plane. In the soils domain both the aggregation plane and the generalization planes are required. Therefore the single hierarchical structure represented by the context tree is insufficient.

The main knowledge representation for PROSPECTOR is the semantic net. The use of semantic nets to explicitly represent objects in PROSPECTOR is directly applicable to representing many of the kinds of knowledge structures that have been identified in the soils domain. For PROSPECTOR, the networks shown in known published examples are taxonomic structures; networks in the generalization plane (is-a type relations). In the soils domain much of the important domain knowledge is in the aggregation plane (part-of type relations). In the soils domain both the generalization plane and the aggregation plane would require representative network structures with links between the two planes.

A major advantage of the semantic net notation used in PROSPECTOR is the ease with which the other system modules can use the domain knowledge stored in the semantic nets developed for the inference engine. The flexibility of the semantic net notation permits extra knowledge required by other modules to also be associated with appropriate parts of the domain structure without interfering with the inference process. Language dependent information such as tense usage can be associated with the appropriate entities or attributes as required. The semantic nets developed for inferences can be used by:

1. The communications module - could use the names of the objects and the relationships to create the required dialogue, figure 6.05

2. The acquisition module - could use the definitions of partitions for both relations and rules to form rudimentary acquisition templates, figure 6.08.

3. The explanation module - could use the taxonomic hierarchies to help explain decisions and reasoning, figure 6.06.

The main knowledge representation for CENTAUR is the prototype. The use of prototypes as a knowledge representation appears to match many aspects of the conceptual model of SES outlined in chapter 4. The use of structures representing prototypical knowledge is particularly relevant because type descriptions for a soil classification unit describe a typical soil member. Another advantage of the representation used in CENTAUR is the ability to define strategic knowledge in context. This could be important in the soils domain as much of the knowledge is context dependent. For example, different features of a soil have different levels of significance for different soil types.

The two levels of frames, prototypes (figure 6.11) and components (figure 6.12), used for CENTAUR are not sufficient to adequately describe the soils domain. A component slot in a soil type prototype may refer to a soil horizon such as the B horizon. An horizon is an attribute of a classification unit description but to describe an horizon (figure 5.15) it must be viewed as an entity with a large number of attributes including structure, colour and texture. The structure of this aggregation plane requires a complete new hierarchy of frames. The component slots in CENTAUR function more like specialized type definitions.

The main disadvantages of the knowledge representation used by CENTAUR are:

1. Increased complexity, compared to PROSPECTOR's semantic nets

2. Relative inflexibility, compared to MYCIN's rule based system.

Although the three knowledge bases examined each use a different primary knowledge representation method, a set of similar facilities can be identified. Figure 6.13 compares the representations used for these facilities. This comparison shows that:

1. Explicit specification of some facilities occurs in the knowledge structures, such as the individual semantic nets in PROSPECTOR representing taxonomic hierarchies.

2. Implicit specification of some facilities occurs in the knowledge structures, such as in MYCIN's self referencing rules which define possible values an object may have in a specific situation.

3. Major knowledge structures in one system will be of marginal importance in another system, such as "Patterns of Common Knowledge" which are central to the CENTAUR system while being insignificant in the MYCIN system.

4. Metrics for indicating knowledge reliability are a major factor in all three systems and include measures which indicate both the uncertainty of the knowledge and uncertainty about the data in the dynamic database.

All three systems use rules successfully to represent both heuristic and compiled knowledge, as well as some strategic knowledge. That domain knowledge best represented using objects and relationships is represented using a variety of methods: tables, templates, semantic nets and prototypes.

Figure 6.13 Kinds of knowledge from the three knowledge bases - MYCIN, PROSPECTOR, CENTAUR.

Knowledge Structure	MYCIN	PROSPECTOR	CENTAUR
Main Knowledge Structure	Rules	Semantic Nets	Prototype frames
Objects	- context templates - parameter templates - in rule premises	- nodes in semantic nets - individual semantic nets	- prototype frames - component frames - slots within frames
Relationships between Objects	- links between context and parameter templates - links defining multiple objects - by rules referencing the same hypothesis - by specific rule premises setting context for u se of rule - self-referencing rules	- common relationships of set and subset - special relationships - partitions	- objects representing attributes grouped together within named prototype frames - prototype network shows explicit links between prototype frames - explicit links between prototype and component frames
Values Instances of Objects may take	- tables of factual knowledge - "Expect "slot in parameter templates - specific rules	- terminal nodes in the taxonomic semantic networks	- Plausible values slot in component frames
Heuristics	Rules	Rules	Rules
Relationships between Heuristics	- via the lists of rules associated with each context and parameter template - by the use of Meta rules - rule order	- semantic net of rules - inference network	- related rules stored in named prototype frame - grouped by named type - interlinked inference rules
Kinds of Associations between Objects and Structures	- functions used in premises of rules - "part-of" relationships between contexts and parameters	- special relations - taxonomic semantic nets	- defined in condition action tables associated with component frames
Patterns of Common Knowledge	- Some implied by the structure of tables	Typical patterns of knowledge stored in the taxonomic semantic nets	Prototype frames represent typical patterns of knowledge
Metrics for Indicating Reliability of Knowledge	The Certainty Factor Model - degrees of belief and disbelief - certainty factors	Modified Bayesian Logic - necessity and sufficiency - subjective probabilities	Modified Certainty Factor Model - degrees of belief and disbelief - certainty factors - importance measures
Dynamic Data Base of Assertions	The Context Tree	instantiate instances of special relationships	invocation record slots
Inference Structures	- rules linked by backward chaining via the context tree - slots such as MAINPROPS in context templates control which groups of will be invoked rules - order of rules and premises impose control on the inference pattern	Special relations linked together via the i nference network - logical relations - plausible relations - contextual relations - general strategic and control knowledge	- via the links connecting the prototype frames on the current hypothesis list - prototype context determines control and strategic procedures

Figure 6.13 Kinds of knowledge from the three knowledge bases - MYCIN, PROSPECTOR, CENTAUR.

The main knowledge representations for both PROSPECTOR and CENTAUR appear to be more appropriate for storing soils domain knowledge than those used for MYCIN. The knowledge base structures used to represent the soils domain knowledge should make clear the main relationships and functions of the knowledge. Therefore the representation chosen for SES must address two key points:

1. What significance should the different features of the domain knowledge be given in the expert system?

2. Which structures will enable this knowledge to be represented clearly and explicitly?

Chapter 7: The Design Phase of SES

The knowledge base structures chosen to represent the domain knowledge are crucial if an expert system is to meet the needs of the users. These needs can directly influence the design of the structures in the knowledge base. The result of the systems analysis at the conceptualization stage should identify these needs which will be reflected in the conceptual model of the domain knowledge. The conceptual model should be referred to during the design of the knowledge base structures.

The preceding chapter surveyed three expert systems each using a different knowledge structure as its main representation. The information gained from this survey was used to develop a prototype experimenting with those structures that appeared to be most suitable for the knowledge base of SES. The evaluation of this prototype in turn indicated the kind of structures that would enable the knowledge for SES to be represented clearly and explicitly. The last section of this chapter outlines the detailed design of these structures.

7.1 Using Prototyping to Experiment with Different Knowledge Structures for Representing the Soils Domain

Before deciding on the best way to represent the soils knowledge base structures described in the Conceptual Stage the rule structure, used for the earlier soils expert system prototypes, needs to be evaluated. Also alternative knowledge base representation structures needed to be investigated and evaluated.

As a first step, expanding the existing rule base of the second prototype was attempted by rewriting the rules to take into account relative weights of evidence for and against each premise. The rule form used was found to be inappropriate for depicting all necessary aspects of a typical soil description for a classification unit. Two major problems occurred with trying to extend the rule structure.

Firstly, the rules used in the second expert system prototype concentrated on the main diagnostic characteristics of the soil classification unit and only used the main value a soil feature may have for that classification unit. To fully represent the unit all the possible values that were possible for a feature of a unit member had to be represented. Each soil feature often has a range of possible values that could be considered correct. To complicate matters the degree of correctness varies depending on the value. For a yellow-grey earth a sandy loam texture is more likely to occur than a silt loam so a rule using the silt loam parameter should have a lower certainty rating, figure 7.01. Thus one existing rule spawned twelve new rules.

if the texture of the A horizon is a sandy loam and
the structure of the A horizon is a weakly developed nut and
the consistency of the A horizon is friable
then the soil is a yellow grey earth, <0.35>

if the texture of the A horizon is a silt loam and
the structure of the A horizon is a weakly developed nut and
the consistency of the A horizon is friable
then the soil is a yellow grey earth, <0.25>

Figure 7.01 Two of the rules created from one parent rule in the second prototype

Secondly, if the diagnostic horizons or features are not well developed in a specific soil example then it is still possible to categorize the soil from the accumulated evidence of features of lesser importance. The group of features identified as major features contributing to membership of a classification unit need not all be present for membership. As indicated in section 3.3.2, within the set of clauses in the premise of a rule, a subset of clauses may be of greater importance in indicating the truth or otherwise of the conclusion. Figure 7.02 shows such a situation where parent material and thickness of the A-horizon have greater importance than the rest of the clauses. The reliability of the first rule in figure 7.02 is <0.95> but the second rule, with a reliability of <0.9>, is also relevant and uses only the subsets of clauses identified as the most important.

if
the parent material is peat and
the topography is a swamp and
the drainage is very poor and
the thickness of the A horizon is greater than 60 cms
then
the soil is classified as organic, <0.95>

if
the parent material is peat and
the thickness of the A horizon is greater than 60 cms
then
the soil is classified as organic <0.90>

Figure 7.02 Two rules with a subset relationship.

If both rules are allowed in the knowledge base then the clauses that occur in both rules will contribute more than their share to the degree of belief when the value of all four features covered by the clauses are known. The rules are not logically independent. The first rule has to be split into two or more rules, but even with only two or three clauses per premise there is still a problem of relative importance among the clauses. The certainty factor model doesn't provide any facility for indicating relative importance to the different clauses in a rule

The rules in figure 7.02 highlighted another problem with the rule representation for representing knowledge about the soil's domain. These rules include both soil forming factors - parent material, topography and drainage, as well as profile features - thickness of the A horizon. The conceptual difference between these two kinds of soil features is not apparent in the rule representation. It is important to differentiate between the different kinds of factors because of the causal relationships that occur between soil forming factors and soil morphology.

Not all the existing rules exhibited these types of problems. The premises of this second group of rules included general references to overall soil characteristics and comparative statements about profile horizons. Figure 7.03 illustrates a case where the first two premises deal with comparisons while the last premise is a generalization about the overall texture of the profile.

```
if
        A and B horizons have a similar texture          and
        A and B horizons have a similar consistency      and
        the parent material is volcanic ash       and
        the texture of the profile is loamy
then
        the soil is classified as yellow brown loam. <0.55>
```

Figure 7.03 Rule using general comparisons of soil properties.

In these rules each premise is of approximately equal ranking in importance. The reliability factors are lower and deciding on the truth of a premise usually requires a greater degree of subjective judgement on the part of the user.

7.1.1 Redesigning the structure of the Knowledge Base

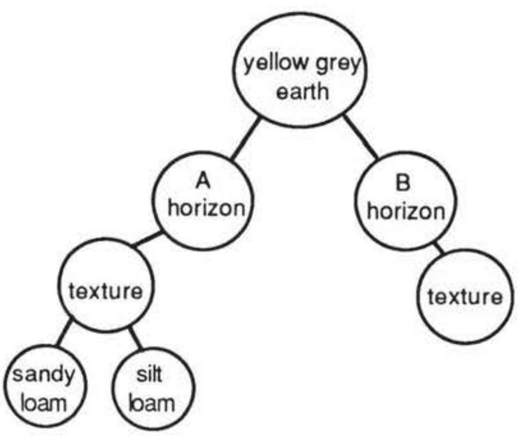
Semantic nets have been used to develop a conceptual model of the soils domain. Therefore parallels could be made with the PROSPECTOR system. Semantic net structures seemed an appropriate alternative to the rule-based structures already tried. From the conceptual model it can be seen that each object in the classification hierarchy of the generalization plane has associated with it a semantic net in the aggregation plane. Part of the knowledge base representing a typical soil description was rewritten using semantic nets matching the aggregation plane of the conceptual model. This redesign of the knowledge base resulted in six different types of knowledge structures that can be divided into three main categories of representation:

1. Rules:
   - general rules.
   - meta rules.

2. Semantic nets:
   - part-of/composed-of hierarchies
   - is-a hierarchy
   - special numeric comparisons

3. Interface text knowledge

The format chosen to represent the semantic nets was a simple relational tuple in which the names of parent nodes were carried down as foreign keys, figure 7.04. The part-of semantic nets for each of the soil groups in this system was extended by adding composed-of relations to indicate the values the different attributes may take. The relational tuple was extended so that the set of all possible valid values an attribute may take was represented as a list. Factual knowledge that required some form of numerical comparisons required special relational links to replace the composed-of relationship.



texture('A horizon','yellow grey earth',[['sandy loam',0.3],['silt loam',0.25]])

Figure 7.04 Part of the Semantic Net for a Yellow Grey Earth with modified tuple representation for A-Horizon.

The knowledge associated with the site descriptor could not all be suitably described using the part-of semantic nets. The heuristic rules were retained and meta rules added. The meta rules were introduced to control the order of processing and to identify knowledge common to more than one soil classification unit. These meta rules were particularly useful for processing the semantic nets. The disadvantage of introducing the meta rules was that they required another variation to the inference structure. A third rule format was used for rules that inferred values that the user did not know.

The certainty factor model was modified so that each composed-of relation had a reliability factor associated with each member of its set of values. These were treated as if each was an independent chunk of knowledge with respect to the accumulating formula used in the certainty factor model.

7.1.2 Evaluating the Changes

One of the objectives in designing the knowledge base was to keep it readable and understandable from the point of view of the experts. The experts considered the new format quite readable. They were enthusiastic about the improved ability of the system to rate the hypothesis but were still not enthusiastic about providing reliability ratings.

Describing the metric corresponding to the members of the attribute lists in the composed-of relations as a number ranking each of the possible values based on their relative importance was a more acceptable approach. The experts ranked the attributes in descending order of importance from 1 down to the number of attributes. These rankings were then used to devise an appropriate certainty value for the system.

Experimenting with the system highlighted problems with the sets of mutually exclusive alternative values, including:

1. How was the system to determine that a value had been assigned to a feature and thereby cease asking about that feature?
2. When none of the values on the possible list of values matches the user's data, which reliability factor is to be used for accumulating evidence against the hypothesis?
3. If the user has assigned a value to a feature earlier and a new hypothesis is now being pursued, how does this effect the calculation of the degree of belief or disbelief?
4. When the user is unsure about the value of a feature what should be stored in the data base?

Two situations can occur where the user should not be queried further about the value of a feature previously mentioned. Firstly if the user has indicated the value is unknown there is no point in pursuing the feature further. Secondly if a user has indicated a degree of belief in a value greater than 0.5, then it is assumed that all further possible values would have degrees of belief whose contribution is insignificant to the final overall certainty factor. This cut off point of 0.5 is quite arbitrary.

When none of the values in the subset list are selected by the user then the value with the maximum reliability from this list is used to calculate the degree of disbelief in the hypothesis. The subset of values represents a set of alternatives therefore the maximum of the reliability factors associated with each value is used to calculate the degree of disbelief. This corresponds to the certainty factor model's use of fuzzy logic (section 2.2.2.2). Common sense also dictates that any doubt about membership of a soil in a classification unit should be made by reference to the most common values of the soil features in the generalized description.

Where a value already exists in the database for a feature then the corresponding certainty rating will determine whether the system continues to query the user and whether the degree of belief or disbelief is incremented. All values associated with a feature that have been queried so far are recorded until one with a certainty rating greater than 0.5 is found. This is necessary because the system only ceases asking the user for values for the feature when one with a rating greater than the arbitrary cut-off value is found. The user doesn't want to be asked about the same value for that feature more than once so the system needs to know which values have been assigned degrees of belief.

In the first version of this prototype each of the different knowledge types was grouped together. This made finding knowledge about a single typical soil description quite difficult, and from a reader's point of view too fragmented. This fragmentation did not perceptibly effect the inferencing process but did effect debugging and updating the knowledge base as finding the correct line of code was difficult. The knowledge base was reorganized so that most of the knowledge associated with the typical description of a soil classification unit was grouped together. The rules common to a number of descriptions were also grouped together but the user interface information was kept separate.

The meta rules which control the traversal of the individual semantic nets are the only form of control knowledge associated with each of these descriptors. The regrouping of the knowledge formed corresponding informal structures, each describing a typical soil classification unit. These structures correspond to the structures used in CENTAUR and indicate that a similar prototype frame representation system may be more appropriate for the soils domain.

7.1.3 Expanding the Prototyping

The knowledge base at this stage contained the knowledge for the four major soil groups only. This was considered insufficient for deciding whether the semantic net based representation was adequate to fully represent the soils domain. It was decided to extend the knowledge base to include all the soil groups and then test it on eight representative soils. The aim of this exercise was to answer two questions:

1. Could the system correctly classify a soil?

2. Could the system be used for ranking the possible classification groups a soil may fall into?"

The knowledge base was extended to include the sixteen soil groups listed in figure 7.05.

Soil	Abbreviation
brown grey earth	BGE
yellow grey earth	YGE
yellow brown earth	YBE
podzol	PODZOL
gley podzol	GP
yellow brown pumice	YBP
yellow brown loam	YBL
brown granular loam	BGL
brown granular clay	BGC
red loam	RL
brown loam	BL
yellow brown sand	YBS
rendzina	RENDZINA
organic	ORGANIC
recent	RECENT
gley	GLEY

Figure 7.05 Soil groups known to the main prototype.

To test the ranking ability of the system the data describing eight representative soils was entered into the system and the system was repeatedly asked to reclassify the soil. Once a soil has been classified into a particular soil group that group was removed from the hypothesis list thus forcing the system to try an alternative. Each of the eight soils was tested against the remaining members of the hypothesis list.

Each of the soils selected is representative of its soil group. Therefore the system should be almost certain about the soils classification because the rules in the knowledge base are based on generalized soil group characteristics. Conversely the system should be certain that the test soil does not belong to any other soil group. Results from this test are shown in figure 7.06.

As expected, the system correctly classified each of the soils when the highest belief value is used. Examination of the ranking for the alternative groups shows that only the system's classification of the Otanomomo peat is unambiguous about both the soil group to which it belongs and those to which it does not belong. Analysis of the anomalies that occur in the results were found to be due to a number of factors that relate to the limitations inherent in the relatively unsophisticated knowledge base structure used in this prototype.

	YGE	YBE	Podzol	YBP	YBL	BGL	Recent	Organic
Timaru silt loam	5	0	-5	-5	-4	-2	-5	-5
Kaituna silt loam	-5	4.5	-4	-5	1.5	-4	-5	-5
Wharekohe silt loam	-3	4	5	-5	-5	-4	-5	-5
Taupo sandy loam	-5	-2	-4	5	-2	-4	-5	-5
Egmont Black loam	-5	-2	-4	-3	5	-5	-5	-5
Naike clay loam	-4	-2	-4	-4	4	5	-5	-5
Manawatu fine sandy loam	-5	1	-4	-4	1	-5	5	-5
Otanomomo peat	-5	-4	-4	-5	-5	-5	-5	5
- 5 definitely not classified as the group +5 definitely a member of the group								

Figure 7.06 Ranking of Soil Group Classification,

An examination of the degrees of belief in the column headed yellow-brown earth (YBE) show that for six of these soils the system is unreliable in clearly determining that those soils do not belong to this group. The yellow brown earth is the most common soil group in New Zealand. Most soil groups are identified as having one or more significant features that can be used to distinguish them from a yellow brown earth. This ambiguity is best illustrated by the rankings for the Wharekohe silt loam. The system correctly identifies the Wharekohe as a podzol but also indicates very strongly that it could be a yellow brown earth. This is not unreasonable because the descriptions are very similar except for one very important feature, the Wharekohe has a well developed E horizon characteristic of a podzol. A rule to the effect that a yellow brown earth can not have a well developed E horizon with a confidence factor of 1 would solve this problem. The NOT operator was not implemented in this prototype.

The feature discriminating between two specific soil groups will often identify a subset of soil groups. The rankings for the Kaituna silt loam, a yellow brown earth, illustrate this situation. The main discriminating feature between a yellow brown loam (YBL) and a yellow brown earth is the rock type of the parent material. But the fact that a soil is formed from volcanic parent material identifies a subset of six soil groups which include the yellow brown loam. To use this fact to discriminate between the two soils the rule would have to specify the particular soil units in question. A set of premises including specific features known to discriminate between the current hypothesis and alternative classification units would be possible using a negation relationship. The number of premises using negation required to specify a yellow brown earth would be relatively large and could lead the system to query the user about features that are unnecessary in the current consultation. The use of rules that discriminate between specific pairs of soil groups once the system has narrowed the set of hypothesis, would present the user with a more understandable series of questions and would also match the experts strategy of checking discriminatory features between competing solutions to confirm a hypothesis.

A further use of discriminatory features is indicated on the state specification diagram at the start of a consultation. The strategy at this point is to consider the features of greatest significance in discriminating between soils in the district. Rules could be used to create an initial set of candidate solutions. The system would require premise-action rules that identify the subsets of possible hypotheses and can thereby control the membership of the hypothesis list.

The implementation of "not" and rules that discriminate between competing hypothesis will not enable a system to determine a suitable solution if the user's data about the discriminating features is inadequate. If the data is not available then these rules could be used to advise the user on further field data collection indicating which information will be useful to discriminate between the competing hypotheses.

The system correctly identifies the Egmont black loam as a yellow brown earth (YBE) but it also indicates very strongly that a Naike clay loam is also a yellow brown earth which it most definitely is not. The Naike clay loam is a brown granular loam (BGL). The yellow brown earth and brown granular loam are quite dissimilar soils except for their parent material. But the clay content and consistency, which are the discriminatory features, are common in soils that do not have volcanic parent materials. By separating out the different soil characteristics into the semantic nets and assigning importance measures to them the relationships between parent material, consistency and clay content in the representation of the brown granular loam was lost. If both sufficiency and necessity ratings were assigned to each feature, as shown in figure 7.07 this problem would be solved.

(soil-unit, horizon, texture, value, (sufficiency, necessity))
(BGL, B-horizon, texture, clay, (0.4, 1))

Figure 7.07 Tuple example showing sufficiency and necessity.

In figure 7.07 the sufficiency for the soil to be a brown granular loam, given that the B-horizon has a clay texture, is "0.4". The soil given this evidence could be a northern yellow brown earth or a central yellow grey earth, but the necessity value is "1" because a brown granular loam must have a clay textured profile.

7.2 Structural Design for SES

The structures designed to store the soils domain knowledge must be expressive enough to represent all appropriate aspects of the domain explicitly. The knowledge in these structures must also be easily accessible to the different modules that make up an expert system. In section 4.4.5 eight different types of knowledge were identified as required by the different modules of an expert system. Figure 7.08 shows how these different types of knowledge can be incorporated into a knowledge base.

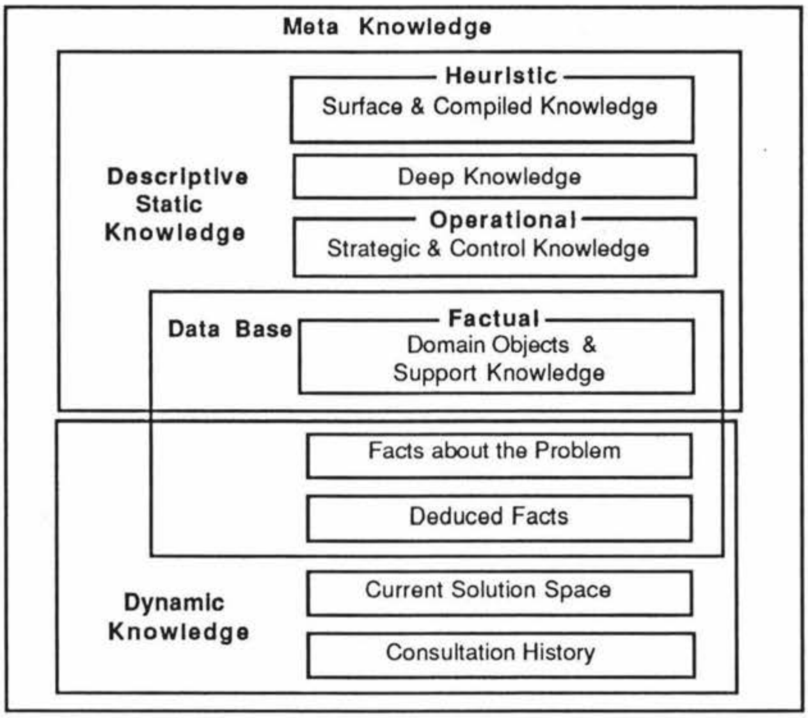


Figure 7.08 Shows groupings of different types of knowledge in the Knowledge Base.
(Based on Scott et al, 1977)

As well as representing these types of knowledge the structures designed must be able to satisfy the general requirements of the system. From the discussions of the early soil expert system prototypes, the nature of the soils domain and the requirements of the system modules a number of general issues keep recurring:

1. The way the system interfaces with the users determines whether the system is acceptable to the users.

2. Each stage in the consultation process has different knowledge requirements.

3. Each module in the system has different knowledge requirements.

4. Some features are associated with a body of heuristic knowledge that is relevant to the whole of the soils domain while other knowledge is context dependent.

7. Neither the Certainty Factor Model nor Bayesian Logic address the problem of the relative importance of different premises in a rule.

6. Instances of a classification unit require different strategies for determining membership once a subset of possible solutions has been established.

7. Descriptions associated with the different levels in the soils classification hierarchy are generalizations, mid points in a continuum. For each classification unit there is a type location which is the site of the typifying soil description for that classification unit. Therefore the classification hierarchy is a structure containing members each with a central pattern of typical knowledge.

8. The generalized descriptions detail the salient features for the soil unit and define the range of values these features may take in specific instances. The parent/child relationships between instances of classification units means that the generalized description of the parent unit defines the range of values features any child unit may take.

9. Generalized descriptions are used to answer the question:

   When does a soil match the typical pattern of a classification Unit?

   In the Soil Survey Method Taylor and Pohlen (1970) state that the field pedologist should consider the following when mapping at the type level:

   " The variations within a type should not be greater than the differences between it and other types"

   This quote can be generalized to cover all the classification units indicating that at any level within the soil classification hierarchy the variations from the general description of one instance should not exceed the variations from the general description of another instance.

From the discussion in sections 7.1.2 and 7.1.3 it is clear that structures that represent all of the knowledge about the typical soil description for a classification unit would be most appropriate for SES. These typical soil descriptions dominate the soil classification system. In CENTAUR prototypes describing typical diseases are the main knowledge structures. As both systems are characterized by typical descriptive members similar knowledge structures may also be appropriate.

To be able to meet the design requirements for SES different knowledge representations need to be used for the different types of knowledge required. The structures represent the three main types of knowledge identified during the conceptualization stage:

- generalization knowledge
- aggregation knowledge
- strategic knowledge

The simple two level hierarchy used in CENTAUR (section 6.1.3) is not sufficient to describe the soils knowledge. A structure of objects that reflects the conceptual model outlined in section 4 is required. Such object oriented descriptions are analogous to frames. Frames describing both the generalization plane and the aggregation plane are necessary to fully represent the objects in the soils domain. Common strategies can also be described using frames. Strategic knowledge that is context sensitive can be represented using rules incorporated within the appropriate frame along with knowledge required by both the communication and explanation modules.

Using the framework outlined in figure 7.08 the main structures required for the knowledge base SES are:

The Meta Knowledge:	Structures describing the structures and relationships used to represent the static and dynamic knowledge.
The Static Knowledge:	Structures representing the typical soil descriptions.
The Dynamic Knowledge:	Structures storing the field data supplied by the user.

Strategic, communication and explanation related knowledge should be associated at the correct contextual level with each of these kinds of knowledge. At a more detailed level the following structures are required:

- structures describing: objects
  strategies
  types
  operations
- interlinked hierarchies of structures
- rules
- ordered lists
- a context network

7.3 Specifying the Knowledge Structures for SES

The product of the design process is the specification of the structures required by the system. In database management systems data representation structures are specified in the form of a schema. Davis (1978) used a similar specification device to specify the structures used in MYCIN. He defined a schema as:

> " a device which provides a framework in which representations can be specified *...that emphasizes* the specification of many different kinds of information about representations."

To specify the structures required for SES a similar system based on the work reported by Davis (1978) has been used. A schema consists of meta data that details the conceptual model outlined at the specification stage. For SES four different levels of description are required:

1. The schema network: which links the different categories of schema describing the different representations used in the knowledge base.

2. Schema: structures which are the individual descriptions of each of the different structures used to describe the soils domain.

3. Objects: structures which are the individual descriptions of objects identified as part of the description of the domain knowledge.

4. Types: structures which define the values or contents of the individual attributes or component parts of the objects.

Schema for SES are described in structures similar to frames. The separate parts used to build the schema are analogous to slots in a frame. All frames used to specify structures in SES have two common sections:

Section 1. The links to the different hierarchies and to individual objects with which the schema is associated.

Section 2. The interface knowledge which defines information that will be required by a user via the explanation, knowledge elicitation or consultation interface modules.

The schema frames have a third section describing the structure of individual members or instances of the schema. Frames representing instances of objects have sections which are used to specify the domain specific knowledge. These sections vary in number and type depending on the knowledge being stored. The structure for each kind of object is outlined in the appropriate schema.

Schema are organized into a schema hierarchy, figure 7.09. The schema hierarchy links the different specification objects and the instances of those specifications together. The root of the schema hierarchy is the knowledge-structure schema. The second level in this hierarchy are the schema that describe each of the structures used in SES's knowledge base. These are:

1. The State-Schema which defines the structure of the objects representing strategic knowledge required for each consultation.

2. The Classification-Schema (is-a objects) which defines the structure of the objects representing the generalization plane. This plane describes the soils classification system used in SES.

3. The Soil-Description-Schema (part-of objects) which defines the structure of the objects representing the aggregation plane. This plane outlines a soil description.

4. The Type-Object-Schema which defines the data structure in which values acceptable for a specific feature are stored. The instances or type-objects will include associated knowledge such as rules for inferring a value,comparison or function limitations and interface information.

7. The Field-Data-Schema which defines the structures for storing the data about the current problem. This data includes the values supplied by the user as well as those inferred by the system.

6. The Rule-Schema which defines the structure of the rules and stores information such as text translations etc for each premise and conclusion.

7. The Operator Schema which defines the structures required to outline the different comparison operators and operations which define the membership of different features in a descriptor.

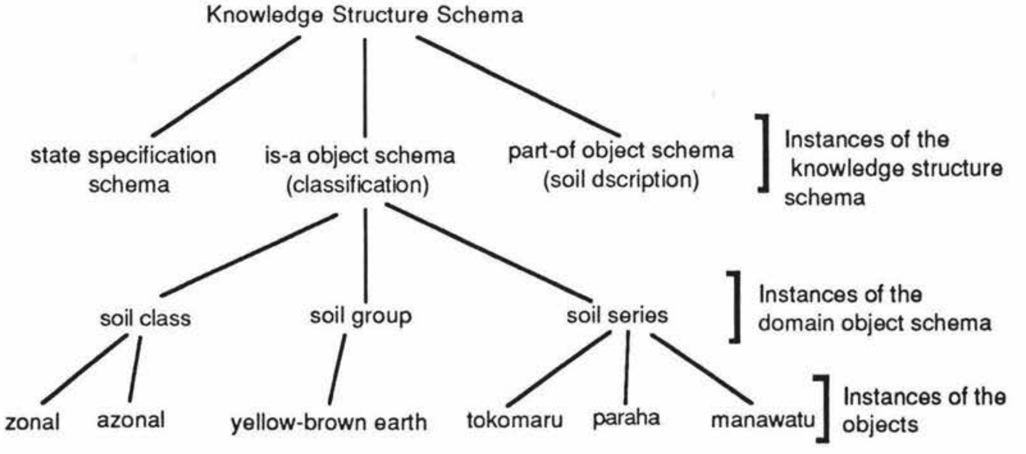


Figure 7.09 Part of the Knowledge Structure Hierarchy showing links to specific instances.

The three main members of the second level of the knowledge structure hierarchy define the three major aspects of the knowledge base as outlined in section 4.6. The middle branch in figure 7.09 shows that the soil group and the soil class are both a specific type of classification object even though there is a parent-child relationship between these two units in the generalization plane. Linked to each instance of a classification schema are the instances that occur in the knowledge base, for example, Tokomaru, Manawatu, and Paraha are all instances of soil series.

The schema matching the main knowledge base structures identified during the conceptualization stage of SES, and the root node of the schema hierarchy make up the major set of schema describing SES's knowledge base. These are:

- The Knowledge-Structure Schema
- The State-Schema
- The Classification-Schema
- The Soil-Description-Schema
- The Type-Object-Schema

7.3.1 The Knowledge-Structure Schema

A definition of the detailed structure of each kind of object is the result of the design process. Because the definition of a schema defines the structure of the instances of a schema, schema are a useful design tool. The schema do not necessarily have to be implemented but may provide the base for a fully implemented knowledge acquisition module. An acquisition module should be able to generate the complete knowledge base from the knowledge-structure-schema.

The structure of the knowledge-structure-schema is the same as for other schema-schema frames. As with all schema the knowledge-structure-schema has three sections:

- the Link Section
- the Instance-structure Section
- the Information Section

Each frame (figure 7.10) in the system has a unique number for identification purposes as well as slots for stating its type and its name. Names should also be unique and infer some meaning to the reader as to how the frame fits into the knowledge base. The link section sets up the pointers to the next level of the schema-hierarchy by naming the specific instances. The kind of instance is also defined in the link section.

The information section has four slots three of which are present in all frames. The Date-created and Author slots are self evident while the Translation slot contains text describing what the frame is and how it fits into the knowledge base. All schema-schema have a fourth slot that references the operator "create-frame" which will present the user with the appropriate dialogue to create an instance of the schema defined in the instances-structure section.

frame-number:1	
LINK SECTION	
Frame:	schema-schema
Name:	knowledge-structure-schema
Instances:	(state-schema,classification-schema,soil-description-schema,type-object-schema,field-data-schema,rule-schema,operator-schema)
Instant-type:	schema-schema
INSTANCE-STRUCTURE SECTION	
Link:	(Frame : Instance-type), (Name : one-of(Instances)) (Instance-of : Name) (Instances: list-of(Instances->Instances)) (Instance-type: one-of(state,object-schema,type-object,operator)) (Instance-root: one-of(Link.Instance)
Instance-structure:	
Information:	(Translation : text) (Instance-prompt: create-frame(Link.Instance-type) (Date-created : date) (Author : text)
INFORMATION SECTION	
Translation:	This frame describes the structure of the root node of the schema hierarchy. The knowledge-structure-schema defines the structure of the main schema for the knowledge base.
Instance-prompt:	create-frame(schema-schema)
Date-created:	24-4-88
Author:	Lis Todd

Figure 7.10 Frame describing the Root Schema of the Knowledge Structure Hierarchy.

The most important section of the schema is the instances-structure section. The instances of the knowledge-structure-schema are themselves schema-schema. They also have three sections:link,instance-structure and information. As can be seen in figure 7.10, the link section for the instances has two extra slots defining

Instances-of: which links back to the knowledge-structure-schema

Instance-root: states the root of the instance objects hierarchy

Associated with the slots defined in the instances-structure section is a definition of the type of the knowledge. These definitions often have an operator associated with them such as list-of or one-of and reference the contents of other slots in the frame, for example Name is defined as one-of the instances. When the definition has to refer to another slot that is also being defined then the conventional use of a period to reference part of a record is used. To indicate a link via the pointer structure to another level in a hierarchy the conventional arrow is used.

Although the slots have been defined in the schema only those relevant to the instance are used. Also note as expected, that the Instance-structure slot in the Instance-structure section is empty.

7.3.2 The State Schema

The state schema defines the structure of the objects that define the domain experts main strategies for classifying a soil during a consultation. The instances of this schema define in detail the arcs of the State Specification Diagram, figure 5.21. It is the activation of these state frames that control the course of a consultation. Each state frame has a section that outlines the conditions for other states to be placed on an action list. To enter the next state in the consultation the action list is searched. The action list is a stack from which the inference mechanism takes the latest frame whose conditions for being invoked have been satisfied.

The instance-structure of the state schema (figure 7.11) defines six major sections including the common link and information sections. The key slots in the link section are:

The Input-list slot: Defines the subset of the classification unit instances that have been identified as input to this transition.

The Output-list slot: Defines the subsets of the classification unit instances that have been identified as output to this transition. These belong to the same domain as the input.

INSTANCE-STRUCTURE SECTION	
Link:	(Frame : Instance-type), (Name : one-of(Instances)) (Instance-of : Name) (Input-list: all((one-of(classification-descriptor->Instances)->Instances) or list-of(candidates , possible-candidates, solutions, rejects:list) (Output-list:list-of(candidates,possible-candidates,solutions, rejects: list))
Conditions:	(Invoke: list-of (request, condition : operator(count(one-of(Input-list, Output-list)), constant)) (Output-membership: list-of(add-to(one-of(Output-list),Rule or (condition : operator(certainty-factor, numeric-constant)))
Steps:	list-of(domain-feature : in(object-descriptor))
Operations:	list-of(actions) list-of(rules)
Next state:	list-of(rules)
Information:	(Translation : text), (Date-created : date), (Author : text)

Figure 7.11 Section describing the Instance structure of the State Schema.

The other sections define the contents of the instance frame. These sections outline the domain specific knowledge required by the expert system. The state-frame has four such sections:

The Conditions Section: Outlines the triggers for invoking a state frame. Before the frame can become active these conditions must be meet. The conditions section also defines the conditions for adding and removing members from the appropriate list that are named in the Output-list slot.

The Steps Section: Details the steps that have to be carried out to move to the next state. These steps relate to the type of domain information required to be able to make a decision on where processing is to move to next.

The Operations Section: Details how these steps are met. Usually these operations are in the form of rules with the action part usually directing control to the Output membership slot. The order of these rules is significant

The Next State Section: Outlines a set of conditions for accessing states that can be reached from this state. The state frames whose conditions are satisfied are placed on the action list.

Figure 7.12 shows parts of the start-state frame which is an instance of the state-schema. The list of members of the input-list slot are given but the members of the output-list slot can only be determined during a consultation so just the names of the lists are recorded. The trigger for invoking the start-state is a request from the user for a consultation. The output-membership slot defines which rules need to be satisfied for adding and removing members from the appropriate lists named in the Output-list slot.

• • •		
Input-list:	(Manawatu series, Paraha series, Levin series,Shannon series, . . .)	
Output-list:	(candidates , possible-candidates)	
CONDITIONS		
Invoke:	consultation-requested	
Output-membership:		add-to(possible-candidates,Rule(01) or Rule(02)) add-to(candidates, (Rule(03))
STEPS		
Geographic-position:		in(geographic-position-descriptor)
Physiographic-position:		in(Physiographic-position-descriptor)
OPERATIONS		
Rule(01):	if then	same(geographical-position) Output-membership
Rule(02):	if then	same(Physiographic-position) Output-membership
Rule(03):	if then	member(&current-object,possible-candidates) & similar(geographical-position) & similar(Physiographic-position) Output-membership
NEXT STATE		
Rule(11)	if then	greater(count(candidates) , 0) invoke(test-candidates)
Rule(12)	if then	equals(count(candidates) , 0) & equals(count(solutions) , 0) & greater(count(possible-candidates) , 0) invoke(subset-candidates)
• • •		

Figure 7.12 Part of the frame describing the Start-state.

Two steps are defined as necessary for completing this state: considering geographic position and physiographic position. The first two rules of the operations section place frames into an output list while the third rule uses membership of that list as one of the clauses in its premise.

7.3.3 The Classification Schema

The generalization plane described in section 5.4.1 is defined using a matching hierarchy of object-schema frames. Figure 7.09 shows that three levels of frames are required to define the generalization plane. The classification schema describes the common structures of the object-schemas or descriptors used to describe the classification units in the soils domain. Each classification object-schema in turn describes the structure of the associated object frames, which in turn store the domain knowledge about that soil classification unit.

Figure 7.13 outlines the Link section of the frame describing the soil series classification unit. Three new slots are of significance in this description. The Parent and Children slots link the frame into the hierarchy of schema describing the generalization plane. The slot labeled Cluster links the generalization plane to the aggregation plane.

LINK SECTION	
Frame:	object-schema
Name:	series-descriptor
Instance-of:	classification-schema
Parent:	group-descriptor
Children:	type-descriptor
Instances:	(Manawatu series, Paraha series, Levin series,Shannon series, . . .)
Instance-type:	is-a-object
Cluster:	soil-descriptor

Figure 7.13 Link section of the frame describing one classification unit in the generalization plane.

Figure 7.14 outlines part of an is-a object frame that defines the Shannon series soils. As well as being linked into the hierarchy making up the instances of the generalization plane, this frame is also linked to individual frames describing either a soil series or a soil Type-Descriptor. These associated and similar slots are used to check classes of soils that may have been missed by the general strategy for selecting and checking possible solutions. An important point to note here is that the Cluster slot defining the link to the aggregation plane uses a composite key. A composite key is required to differentiate between all the different soil descriptors. Therefore the soil-descriptor (Shannon series, soil descriptor) describes the typical features of a Shannon series soil.

There is no Instance-prompt in the information section but there are two slots describing different aspects of this frame:

Translation: describes how the frame links into the different hierarchies in which it participates.

Explanation: gives a general textural description of the domain characteristics defined by the frame. In this example, a description of the Shannon series soils.

There are five sections outlining the important domain knowledge necessary for the SES. These sections as with the previously described state-objects, describe factual, meta and strategic knowledge. They are:

Optional features: are important to only a small number of instances. In this example, because it is an intergrade, the Shannon series has three parents, two soil groups and the Horowhenua county.

• • •	
Name:	Shannon Series
Instance-of:	series descriptor
Parent:	(Horowhenua, yellow-grey earth, yellow-brown earth)
Children:	(Shannon silt loam, Shannon silty clay loam, Shannon sandy variant,...)
Associated:	(Levin series, Koputaroa series)
Similar:	(Levin series, Paraha series,Koputaroa mottled sandy loam,Shannon variant)
Cluster :	(Shannon Series, soil-descriptor)
OPTIONAL FEATURES	
No-of-soil-groups : 2	
Modifier :	"Intergrade"
ESSENTIAL FEATURES	
Site-descriptor:	(Drainage = "imperfectly drained") (Slope = range(0,7))
CHARACTERISTIC FEATURES	
A-horizon:	(colour = "dark brown") (consistency = "friable") (texture = "silt loam") (structure = "moderately developed nut")
• • •	
COMPARE SOLUTION	
Rule 01:	if less(drainage) & less(Low-chroma-colours : range(30,60),5%) then consider(Levin Series)
• • •	
IDENTIFY TYPE	
Rule 11	if greater(certainty,4) & similar(texture,"silt loam") then consider(Shannon silt loam)
• • •	
INFORMATION SECTION	
Explanation :	Shannon soils describe soils formed from loess on gentle rolling slopes of the dissected terrace land and on sloping fans. The rainfall is slightly higher and more evenly distributed than the similar Tokomaru series. These soils are moderately gleyed soils developed in at least 50 cm of loess on either gravels, sands, colluvium or older loess. They are imperfectly to poorly drained with low chroma colours within 60cm of the surface. The Shannon series is an intergrade between a yellow-grey earth and a yellow-brown loam.
• • •	

Figure 7.14 Part of the frame describing the Shannon Series.

Essential features: identify the soil forming factors that are most important to the identification of this classification unit. The range of values allowable will be stored in the appropriate soil descriptor but the most common value for each feature is defined at this level for clarity and explanation purposes.

Characteristic features: identify the soil profile features that are most important to the identification of this classification unit. As with the preceding section the range of values allowable will be stored in the appropriate soil descriptor with the most common value for each feature being defined at this level.

Compare solution: A series of rules are defined which indicate under what conditions alternative solutions should be considered. The rules are listed here for clarity but if the system is to be implemented using frames each rule would be stored in it's own rule-object frame.

Identify type: These rules indicate which of the children in the classification hierarchy should be considered as possible solutions.

7.3.4 The Soil Description Schema

Three levels of description are also required to describe the aggregation plane. These are:

- the soil-description-schema
- the object-schema
- the part-of-objects

As each of the part-of objects hierarchies is clustered to a specific member of the is-a objects in the generalization plane the composite key described in section 7.3.4 has to be generated down through the aggregation plane. Each frame name therefore has two parts. The first part is the foreign key derived from the associated classification unit, for example "series-descriptor". The second part is the name for the type of descriptor being defined, for example "site-descriptor". The link section, figure 7.15, shows that these composite keys are also used to define links to other aggregation plane frames.

LINK SECTION	
Frame:	object-schema
Name:	(series-descriptor, site-descriptor)
Instance-of:	soil-description-schema
Parent:	(series-descriptor, soil-descriptor)
Children:	(series-descriptor, parent-material-descriptor)
Instances:	((Manawatu series, site-descriptor), (Paraha series, site-descriptor), (Levin series, site-descriptor), (Shannon series, site-descriptor), . . .)
Instance-type	part-of-object
Cluster:	series-descriptor
Instance-root:	(series-descriptor,soil-descriptor)

Figure 7.15 Link section of the frame describing the schema for one descriptor in the aggregation plane.

The Features slot is the most important part of the instance structure section as it defines the soil features describe a soil belonging to the particular classification unit. Each entry in the features slot names a soil feature and defines the type of values associated with that feature In SES the definitions of these structures depends on the complexity of the individual descriptions, figure 7.16. Some features such as drainage are relatively straight forward and a type-object description frame is sufficient to define the allowable values. Alternatively parent-material requires a relatively complex semantic net to describe all the options. Each leaf in the parent-material sub-tree is linked to a type-object description.

INSTANCE-STRUCTURE SECTION		
• • •		
Features :	(Physiographic-position :	list-of(physiographic-type))
	(Topography :	list-of(range(topographic-type))
	(Vegetation :	vegetation-type)
	(Land use :	land-use-type)
	(Elevation :	range(elevation-type))
	(Rainfall :	range(rainfall-type))
	(Drainage Class :	range(drainage-type))
	(Parent Material:	in (Link.Children(parent-material-descriptor))
• • •		

Figure 7.16 Defining the Features section in the instance-structure section of the site descriptor schema.

In the Shannon series site-descriptor the actual values are recorded (figure 17) for each feature where a type-object was defined. If a further level of frame descriptions is required then the operator "in" is used as in the Parent material slot. An importance rating needs to be associated with each of these values. This metric can then be used to calculate the certainty factors used in the inference process. Assigning such values at this point in the development of the system is considered an inappropriate detail better left to the implementation stage of development.

FEATURES SECTION	
Physiographic-position :	[[high, terraces, bordering , Hautere Plains] [high, terraces, North, Otaki River, to, Manakau]]
Topography :	[[high, terraces] [terraces, older than, Ohakea terrace]]
Vegetation :	"pasture grasses"
Land use :	"mixed pastoral - dairy, beef, sheep, goats, deer " "scattered horticulture"
Elevation :	10 - 100 meters
Rainfall :	950 - 1200+ millimeters
Parent-material :	in(Shannon-Series,parent-material-descriptor)
Drainage Class :	" imperfectly drained "

Figure 7.17 The Features section Shannon series site descriptor.

At this stage in the design process a decision has to be made as to what are matchable units. Physiographic-position is an important feature for differentiating between solution groups and it is necessary to be able to decide whether two descriptions are similar. For example if the user indicates that the site of the soil is on a high terrace over looking the

Otaki river then the inference is that the soil is in a similar position to that indicated in the slot. Each significant clause or word is therefore delineated. Land use is not considered an important discriminatory feature so a simple constant string is sufficient. Allowable values are defined in the appropriate type-object frames

7.3.5 The Type Object Schema

The Type-object schema describes the structure of the type-objects which in turn define the allowable values a slot with this type may take. These values are defined in the values slot. Figure 7.18 shows how each type-object is associated with an aggregation part-of-object via the used-by slot. The descriptive part of the frame has five sections:

Range information: This section outlines the limitations for comparison operators. This information is required so the system can determine, for example whether a value in the field data is similar to the typical value of the current-solution.

Special terms: These are terms that are in common use by either the users or the experts. They have to be defined in terms of the accepted values.

Infer: This section contains a list of rules the system can use either to infer a value for the feature from field data already collected, or in the case where the user does not know the value these rules can control further consultation with the user to determine a possible value.

Clarify: When the user does not understand the question the clarify option can be chosen to allow a series of questions using less technical terminology to be pursued.

Explanation: A simple explanation in the information section is insufficient for the type-object frames. The explanation section has a description for each of the valid values the features may take as well as a general definition of the feature.

• • •	
Used-by:	Site-descriptor
Values:	('very poorly drained','poorly drained','imperfectly drained','moderately well drained','well drained','somewhat excessively drained','excessively drained')
RANGE INFORMATION	
least:	'very poorly drained'
greatest:	'excessively drained'
SPECIAL TERMS	
"good drainage":	one-of('moderately well drained','well drained','somewhat excessively drained')
INFER	
Rule 01:	if same(A-horizon->(texture,peaty) & same(B-horizon->(colour,low-chroma-colours) & similar((topography, hollow) then (drainage,'very poorly drained')
• • •	
CLARIFY	
Rule 11	if ask("Does water pond on site frequently during winter") then (drainage,'very poorly drained')
• • •	
EXPLANATION	
Feature:	Drainage as a condition of the soil, refers to the frequency and duration of periods when the soil is free of complete or partial saturation. The drainage class refers to the average state of drainage over a period of time, usually a year.
'very poorly drained':	The water is removed from the soil so slowly that the water table remains at or on the surface the greater part of the time.
• • •	
INFORMATION SECTION	
Prompt:	'How would you describe the overall drainage of the soil at the site ?'
• • •	

Figure 7.18 Part of the frame describing the type-object for drainage.

Much of the knowledge contained in these sections is required by the interface and explanation modules as outlined in section 4.3. Each of the rules for inferring the value and clarifying the question by using simpler terms, should have degrees of belief associated with them to indicate their reliability in inferring the correct value for the feature. Until a working prototype is available, determining the correct values of such metrics at this point in the design phase is inappropriate. The allocation of values to this type of metric is subjective and the results of such decisions need to be easily evaluated so changes can be made.

7.3.6 The Auxiliary Schema

Three other kinds of schema are required to completely characterize the knowledge-base structures required for SES. They are:

- The Field-Data-Schema
- The Rule-Schema
- The Operator Schema

The most important of these is the field-data schema which describes the facts in the dynamic data base. The semantic net notation used for in the semantic net prototype proved a satisfactory representation. The main representation used is an extended tuple structure (figure 7.19) similar to those used in the MYCIN system.

(entity, attribute, value, certainty rating)
(parent entity, child entity)

Figure 7.19 The two types of tuples used to create semantic net for field data.

These tuples create a semantic net that matches the frame network of the aggregation plane. The values allowable in the values position are defined in the type-object frames. Figure 7.20 shows part of the semantic net representing field data for a Shannon silt loam.

(soil-descriptor, site-descriptor)
(site-descriptor, drainage, "imperfectly drained", certainty rating)
(site-descriptor, slope, 5, certainty rating)
(site-descriptor, "A horizon")
("A horizon", colour, "dark brown", certainty rating)
("A horizon", texture, "silt loam", certainty rating)
(site-descriptor, "A horizon")

Figure 7.20 Part of a semantic net recording field data.

The rule-schema sub-tree defines a structure for linking the different clauses of a rule together, both the premise and the conclusion. The sub-tree would include a schema for defining the information associated with each clause including translation, prompt and explanations text. Some instances of the clauses would need links to type-objects to facilitate good explanation.

The definition of the operator-schema would be one of the final stages of the design phase and will overlap with the implementation phase. The operator-schema will probably have at least four types of sub-schema. These include schema for:

1. Operators that manipulate lists. i.e. - add-to
   - delete-from
   - count

2. Operators that compare values. i.e. - same
   - similar
   - exists

3. Operators that invoke the other frames. i.e. - consider

4. Operators that invoke the other modules. i.e. - ask

The definition of the comparison operators are crucial to the operation of the system. They have to be able to compare like values of different types. Features that have numeric values such as rainfall are straight forward. Features that have values on a linear continuum such as drainage can also be defined with relative ease. The difficulty arises for features such as colour and texture where three variables are involved. As well as comparing like values these operators will also have to take into consideration the certainty factors associated with values in the soil field data.

As more knowledge is added to the knowledge base special relationships will have to be evaluated. These will often involve the definition of a new relational operator. The knowledge engineer needs limit the number of operators to a meaningful but manageably small set.

7.4 Summary

The design for SES proposed in this chapter attempts to use the most appropriate representation for each type of domain knowledge identified as important in the conceptual model. This can be seen by examining the summary of knowledge base structures used in SES, figure 7.21. Rules have been used for heuristic knowledge and strategic knowledge, frame-like structures have been used to describe the major objects within the domain and networks linking the objects record the relationships between these objects.

The main structures of SES are the objects describing the generalization plane and the associated aggregation plane objects. Each network of objects representing a typical soil description is linked to the appropriate classification object forming a prototypical description for the soil classification unit. For example, the description of the Shannon Series outlined in the examples from sections 5.4.3 and 5.4.4 ,forms the prototype for the Shannon Series and is clearly indicated in these structures, by the need to carry the name of the classification object down through the soil description objects with which it is associated. This is similar to propagating a foreign key down through a series of relational data base relations to maintain the integrity of the tuples.

The four levels of description used to specify the detailed design of SES do not all need to be implemented during the next stage in the development of SES. The creation of an knowledge acquisition module using the schemata would make additions to the domain knowledge relatively easy. A satisfactory system could be built in Prolog using the instances of the classification objects and soil description objects with associated type-objects. It should also be possible to construct the system using an expert system shell that has provision for both frames and rules. Obviously the mapping of the specified design to the implementation representation will require compromises to create a workable system.

Chapter 8: Summary and Conclusions

The initial purpose for this research project was to investigate the application of Expert System technology in the soil sciences. The first stage of the investigation was to clearly establish the scope of the project. From this work it became obvious that there was very little practical information available on relevant expert systems development methodologies. Although the main application area, identification of a soil from incomplete field data, was retained, the emphasis of the research evolved towards investigating development methods and tools required to help the developer of an expert system analyse a problem domain and design a suitable expert system solution.

Therefore the aims of this thesis were to:

1. Review Expert System Development Methodologies.
2. Examine the role of prototyping in the development of an expert system.
3. Evaluate methods for designing an expert system knowledge base.
4. Investigate the application of expert system technology to identification and classification of some New Zealand soils.

These aims have largely been satisfied and the experience gained in developing a design for an expert system to identify soils from incomplete field data (SES), has led to the development of the design methodology documented in this thesis.

8.1 Expert System Development Methodologies.

A number of different methodologies for developing expert systems have been examined in the course of this thesis. The expert system development methodology outlined by Hayes-Roth et al (1983) was chosen as a basis for guiding this research. Aspects of alternative methodologies have been incorporated into the general structure as required. Hayes-Roth's methodology provided a suitable framework on which to develop the design of SES. It consists of the following stages:

- identification
- conceptualization
- formalization
- implementation
- testing

These stages were compared to those of the traditional life cycle for systems development. The latter are widely known and can be used as an informal bench mark.

At each stage in their methodology Hayes-Roth et al has provided sets of guide-lines but few specific tools for actually carrying out the steps. This thesis has investigated several different tools that enable the knowledge engineer and domain expert to define and document the types of information and knowledge required to develop an expert system.

8.2 The role of Prototyping in Expert System's Development

Hayes-Roth's methodology emphasizes the importance of prototyping. Consequently, the role of prototyping techniques was used to explore and experiment with different aspects of the development process. The prototypes developed were expendable prototypes and each was developed with a specific set of objectives in mind.

The evaluation of the initial prototyping exercise established the feasibility of developing a soils expert system and also identified further areas requiring study, including:

- the representation of soils knowledge
- dealing with uncertainty,
- the types of interface required by the different user groups.

Prototyping was used at the conceptualization stage to clarify selected systems requirements. The prototypes had very specific objectives. These covered:

- Easy to use interfaces.
- Basic explanation facilities.
- Facilities for simple acquisition and volunteering information.
- Identification of the facilities required to store the domain knowledge in SES.

At the formalization stage the detailed design outlines each of the individual structures required in the knowledge base of SES, and the way in which they interrelate. Prototyping was again used as the main tool to experiment with using semantic nets for storing the factual knowledge about soil profiles. The prototyping exercise also addressed the problem of dealing with uncertainty. The MYCIN Certainty Factor Model was modified to manage the particular problems of storing soils knowledge.

8.3 Specifying the Knowledge Base.

The knowledge base is the most important module in an expert system. The tools for specifying the knowledge base outlined in this thesis were developed specifically for defining and documenting SES. A classification expert system such as SES requires a comparatively large body of factual knowledge in it's knowledge base. Therefore database design methodology becomes applicable and the investigation into suitable design tools includes several database design methods.

Many different types of knowledge are required by an expert system. Eight knowledge types were identified as being important. These are:

- surface
- deep
- compiled
- support
- strategic
- descriptive
- dynamic
- meta

This set of different types of knowledge are all considered necessary to represent the knowledge required for an expert system. The emphasis placed on each of the different types of knowledge will depend on the nature of the domain knowledge.

The three main methods of knowledge representation used in current expert systems; rules, frames and logic, are reviewed in chapter 2. To enable a workable expert system to be developed, the representation chosen for any system should support the desirable features of extendability, simplicity and explicitness. A detailed analysis of the domain knowledge for SES was undertaken in pursuit of these goals.

The basis of the soils domain knowledge includes the typical soil descriptions for the different soil classification classes. The soil classification system incorporates the associations between soil forming factors and soil morphology. The specification of this knowledge is described in the conceptual model of SES. The tools defined for analyzing, designing and specifying the conceptual model of the domain knowledge have been chosen and modified to suit the particular requirements of SES. At the conceptualization stage three aspects of the conceptual model for SES were identified as characterizing the knowledge base adequately. These are:

1. The Inference Structure

2. The Domain Objects

3. The Functional Aspect

Once the conceptual model for SES had been documented the detailed designs of the structures required to represent this knowledge were developed. First, the structures used to store domain knowledge in three similar expert systems were reviewed. This review identified eleven different categories for grouping knowledge structures. These are:

- the main knowledge structure
- relationships between objects
- values instances of objects may take
- heuristics
- relationships between heuristics
- kinds of associations between objects and structures
- patterns of common knowledge
- metrics for indicating reliability of knowledge
- dynamic data base of assertions
- inference structures

Identification of these groupings of knowledge structures led to the development of a specification method. This method is used to specify both the structures and their contents, for representing the knowledge required for SES. Schema were used to define the detail of each type of structure. The instances of structures are used to store both the descriptive knowledge outlined in the conceptual model and the specific domain knowledge it characterizes. An analysis of the knowledge structures designed for SES, by grouping SES's knowledge structures, enables a comparison to be made with the reviewed expert systems.

8.4 Expert Systems and Soil Science

The development of a variety of prototype expert systems and the analysis of the soils domain show that expert system technology is applicable to many aspects of soil science. This assertion is supported by McCraken and Cate (1986) who review the application of expert systems to the U. S. Soil Taxonomy classification system. They state that

> " Expert systems can be used to strengthen efforts to update and revise soil classification, to maintain large data bases such as Soil-5, and to involve field personnel in technical evaluation programs. Expert systems may also be applied to other areas of soil science, where judgement and practical experience by experts can be computerized, such as soil management recommendations on irrigation, fertilizer use and erosion control. "

In New Zealand similar aspects of soil science also appear to be suitable areas of application. For example the time period between data collection and analysis by the field pedologists, and publication of the Soil Bureau bulletins is often several years. A projected future use of systems similar to SES is to provide a faster medium for dissemination of new soil information. This use is dependent on the development of an easy to use knowledge acquisition module that could be used directly by pedologists. SES should also be able to link with the Soil Bureau's standard soil data bases. This facility should enable the system to obtain the required knowledge automatically from the data base of local soils maintained by each Soil Bureau district office from their field notes.

Other potential uses for expert systems in this field are:

1. Capturing of knowledge from experienced field pedologist before they retire.

2. Providing specialized interfaces into existing soil databases. These systems should enable infrequent users of the system to be more productive in their use of the data bases, particularly the use of the statistical features that are available. Expert system interfaces could also be built to provide direct public access to appropriate parts of the information resource.

3. Linking soils knowledge and associated areas of knowledge into systems associated with:
   - pasture, crop, horticulture and forestry establishment and management.
   - decisions on diversification of land use.
   - land resource management and planning.
   - soil conservation planning.

4. Assessing the impact of changes to current soil classification schemes.

5. Making comparisons between the New Zealand soil classification system and those used overseas, in particular the U.S. Soil Taxonomy.

8.5 Directions for further Research

Development of SES should continue with an investigation of the last two stages of Hayes-Roth's methodology. The next stage in this development should initiate a sequence of prototypes that will follow the evolutionary path of prototyping discussed by Huffaker (1986). As an adjunct to this development the application of Bayesian logic for dealing with uncertainty could also be explored. With *a priori* values associated with each clause in a rule and the use of measures for necessity and sufficiency as in PROSPECTOR the problems encountered with the Certainty Factor Model may be overcome.

An important aspect of continuing research related to this thesis is investigating the levels of expert system specification required for commercial expert systems. A review is required to determine:

1. The level and scope of specification required for an expert system.

2. The level of specification required before the implementation of the main prototype system is begun.

This review should determine the extent to which the three views of a knowledge base identified as necessary for SES are sufficient for characterizing: all expert system domains, specific types of expert system, only simple classification systems or just the soils system presented here.

This research should evaluate the suitability of the three documentation tools, the inference diagrams, the semantic nets and the state specification diagrams, as both design tools and as a medium for communication between the different personnel involved in developing an expert system. The study should address the question of whether such tools will help the computing professionals with limited knowledge about Artificial Intelligence methods to develop useful expert systems. These tools may also be useful to domain experts who wish to develop their own expert systems using a suitable expert system shell.

Further work should review the knowledge structures used in expert system knowledge bases for systems other than classification systems, to determine the completeness and generality for categorizing the knowledge base structures identified so far. This review should include expert systems based on the blackboard architecture, as expert systems using multiple knowledge bases have not been addressed in this thesis.

Specification of the structures required to represent the knowledge for SES led to the development of an informal specification language. This informal specification language needs to be evaluated to determine whether it is a useful medium for specifying other expert systems. If so, the language should be formally defined. Before the formalization of the language, a study of alternative specification languages, should determine whether a new language is actually required and also to identify possible inconsistencies and omissions.

8.6 Conclusion

Prototyping has almost universal acceptance as the preferred way to develop expert systems. It definitely has an important role to play in the development of expert systems. However the unfettered use of prototyping is untenable in a tight economic climate. Successful use of prototyping for developing useful expert systems requires that it be used in conjunction with other design tools which are embedded within a more comprehensive expert system development methodology.

Expert systems appear to be particularly useful for management decision support, especially for synthesizing from disparate knowledge sources. Consequently expert systems will play an increasing role in the development of management information systems and decision support systems within both business and scientific establishments. Systems that can combine different knowledge bases, such as those using the blackboard architecture, will become increasingly important.

Further development and wider use of expert systems requires a clearer definition of expert system development methodologies with associated tools. This will enable pertinent and reliable expert systems to be implemented by the general software developer.

Appendix A: Examples from Prototyping Exercise 1.

The first prototyping exercise was implemented on the PRIME 750 using the Salford Lisp/Prolog system and the simple alpha terminals.

A.1 Example Dialogue of Consultation Session.

The user's responses are in bold type.

SOIL IDENTIFICATION SYSTEM

=========================

This expert system asks you questions about different soil features.

From the answers you provide, the system will try to

identify the soil you describe.

This system will classify your soil into one of the soil groups.

This version of the system only knows about:

Brown Grey Earths &

Yellow Grey Earths.

Please answer:

YES - if true NO - if false

PRESS RETURN - if you don't know

or

A numeric value if required.

Is the the site located in Central Otago? (y/n) **y**

Is the texture of both the A and B horizons silty or sandy? (y/n) **y**

Is the colour of the A horizon grey? (y/n) **n**

Is the climate semi-arid? (y/n) **<rtn>**

What is the yearly average rainfall in millimeters? **400**

What is the yearly average temperature in centigrade? **9**

```
Is the vegetation tussocks?  (y/n)  y

Is the topography a basin?  (y/n)  n

Is the topography a valley bottom?  (y/n)  y

Is the parent material schist?  (y/n)  y

Is the colour of the A horizon grey-brown?  (y/n)  y

What is the thickness of the A horizon in centimetres?  10

Is the texture of the A horizon sandy loam?  (y/n)  y

Is the structure of the A horizon weekly developed plates?  (y/n)  y

Is the presence of soluble salts detected in the B horizon?  (y/n)  y

The system has identified the soil you described as most likely being
a zonal soil belonging to the Brown Grey Earths soil group.

>>> Please press return to finish.<<<|
```

A.2 The Dynamic Database.

This is the dynamic database created by the previous consultation session.

```
'the site located in Central Otago'.
'texture of the A and B horizons silty or sandy'.
'vegetation tussocks'.
'topography a basin'.
'topography a valley bottom'.
'parent material schist'.
'colour of the A horizon brown-grey'.
'texture of the A horizon a sandy loam'
'structure of the A horizon weakly developed plates' .
'the presence of soluble salts detected in the B horizon'.
('yearly average rainfall in millimeters',400).
('yearly average temperature in centigrade',9).
('thickness of the A horizon in centimetres',10),
```

```
identity('zonal soil').
group('brown grey earth').
significant_characteristics('brown grey earth').
soil_factors('brown grey earth').
typical_climate('brown grey earth').
A_horizon('brown grey earth').
B_horizon('brown grey earth').

asked('the site located in Central Otago').
asked('texture of the A and B horizons silty or sandy').
asked('colour of the A horizon grey').
asked('climate semi arid').
asked(range('yearly average rainfall in millimeters', 330 , 500).
asked(range('yearly average temperature in centigrade', 7 , 10).
asked('vegetation tussocks').
asked('topography a basin').
asked('topography a valley bottom').
asked('parent material schist').
asked('colour of the A horizon brown-grey').
asked(range('thickness of the A horizon in centimetres', 8 , 10)).
asked('texture of the A horizon a sandy loam').
asked('structure of the A horizon weakly developed plates').
asked('the presence of soluble salts detected in the B horizon').
```

A.3 The Knowledge base.

To simplify the implementation of the first prototypes the inference engine incorporated the use of the standard prolog inferencing mechanism. Therefore the knowledge base rules are written using Edinburgh prolog syntax for rules.

The following knowledge base is from the first prototyping exercise and contains knowledge about brown grey earths and yellow grey earths.

```
identity('zonal soil') :-
    group('brown grey earth').

identity('zonal soil') :-
    group('yellow grey earth').

group(G) :-
    significant_characteristics(G).
```

```
group(G) :-
     soil_factors(G),
     soil_description(G).

soil_description(G) :-
     A_horizon(G),
     B_horizon(G).

significant_characteristics('brown grey earth') :-
     'the site located in Central Otago' ,
     'texture of the A and B horizons silty or sandy',
     'colour of the A horizon grey',
     'structure of the A horizon weakly developed plates' .
     'the presence of soluble salts detected in the B horizon'.

significant_characteristics('yellow grey earth') :-
     'clay content of the B horizon higher than the A horizon',
     'profile characterized by the presence of a fragipan',
     'climate dominated by an annual moisture deficiency' .

soil_factors('brown grey earth') :-
     ('climate semi arid' ; typical_climate('brown grey earth')),
     'vegetation tussocks',
     ('topography a basin' ; 'topography a valley bottom' ;
                                   'topography the toe of a fan'),
     'parent material schist' .

soil_factors('yellow grey earth') :-
     ('climate dominated by an annual moisture deficiency' ;
                            typical_climate('yellow grey earth')),
     ('topography a high terrace' ; 'topography lowlands' ),
     'parent material quartzo feldspathic loess' .

typical_climate('brown grey earth') :-
     range('yearly average rainfall in millimeters', 330 , 500),
     range('yearly average temperature in centigrade', 7 , 10).

typical_climate('yellow grey earth') :-
     factor('sub soil in summer dry and hard'),
     factor('sub soil in winter wet and sticky').
```

```
A_horizon('brown grey earth') :-
     ('colour of the A horizon grey' ;
                          'colour of the A horizon brown-grey' ),
     range('thickness of the A horizon in centimetres', 8 , 10),
     ('texture of the A horizon a sandy loam' ;
                       'texture of the A horizon a silt loam'),
     'structure of the A horizon weakly developed plates' .

A_horizon('yellow grey earth') :-
     ('texture of the A horizon a loam' ;
                       'texture of the A horizon a silt loam'),
     'consistency friable grading to firm, lower in the profile'.

B_horizon('brown grey earth') :-
     'the presence of soluble salts detected in the B horizon'.

B_horizon('yellow grey earth') :-
     ('colour of the B horizon brownish grey' ;
                       'colour of the A horizon yellowish brown' ),
     'B horizon characterized by a fragipan at it's base'.
```

Appendix B: Examples from the Menu-Driven Prototype

B.1 Example of a voluntary Data entry Session

At the beginning of each consultation in the menu driven Prime based prototypes, the users have the option of entering all their formal field data descriptions before the system attempts to identifying the soil described. Once the data has been entered the system only asks the user for soil forming factors that are relevant to the soil group indicated by the user. If the systems evaluation of the identity of the soil is unsatisfactory the user has the option of asking it to examine knowledge from other soil groups to see if a better match can be made.

The screens are different sizes in the examples to save room.

```
                              soil_classification
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

                            SOIL IDENTIFICATION SYSTEM
                            ==========================

  This expert system asks you questions about different soil features.
  From the answers you provide, the system will try to
  identify the soil you describe.
  The system will try to classify your soil into one of the soils groups.


  Continue:<rtn>
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                            PRESS RETURN to continue
                                Type NO to stop.
```

```
                              soil_classification
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


  Do you want to load data from a previous consultation ?



  Enter response: n
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                       YES - if true          NO - if false
                        PRESS RETURN - if you do not know
                                H - for help
```

```
                         soil_classification
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

 Enter HORIZONS for which you have field data.to enter.

     1 - A horizon                          5 - C horizon
     2 - A2 horizon                         6 - O horizon
     3 - B horizon                          7 - G horizon
     4.- B2 horizon                         8 - E horizon

 Enter enter choice one at a time: 1 <rtn> 5 <rtn> <rtn>
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
            Type the NUMBER for required option followed RETURN
            PRESS RETURN - by itself to indicate end
```

```
                         soil_classification
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

 Enter FEATURES for which you have field data.for the A horizon.

     1 - thickness                          8  - colour patterns
     2 - colour                             9  - accumulation
     3 - texture                            10 - pans
     4.- consistency                        11 - concretions
     5 - structure
     6 - stoniness
     7 - boundary

 Enter enter choice one at a time: 2 <rtn> 3 <rtn> 4 <rtn> <rtn>
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
            Type the NUMBER for required option followed RETURN
            PRESS RETURN - by itself to indicate end
```

```
                         soil_classification
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

 What is the dominant COLOUR of the A horizon ?

     1 -red             6 -greyish brown            10 -yellowish brown
     2 -yellow          7 -dark greyish brown         11 -reddish brown
     3 -brown           8 -brownish grey             12 -brownish black
     4 -grey            9 -brownish yellow
     5 -black

 Enter choice: 6
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
           Type the OPTION NUMBER for the chosen criteria
                PRESS RETURN - if you have no data
```

```
                          soil_classification
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  What is the TEXTURE for the A horizon ?

  1 -sand                      9 -fine sandy loam   16 -silt
  2 -very coarse sand         10 -sandy loam        17 -silty clay
  3 -coarse sand              11 -medium sandy loam 18 -silty clay loam
  4 -medium sand              12 -clay loam         19 -silt loam
  5 -fine sand                13 -sandy clay loam   20 -loamy
  6 -loamy sand               14 -sandy clay
  7 -coarse loamy sand        15 - clay
  8 -fine loamy sand

  Enter choice: 13
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
            Type the OPTION NUMBER for the chosen criteria
                 PRESS RETURN - if you have no data
```

```
                          soil_classification
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  What was the CONSISTENCY of the A_horizon ?

     DRY                           MOIST                    CEMENTED
  1 - loose                     7 - loose              13 - weakly cemented
  2 - soft                      8 - very friable       14 - strongly cemented
  3 - slightly hard             9 - friable            15 - indurated
  4 - hard                     10 - firm
  5 - very hard                11 - very firm
  6 - extremely                12 - compacted

                               WET
       16 - nonsticky                          20 - nonplastic
       17 - slightly sticky                    21 - slightly plastic
       18 - sticky                             22 - plastic
       19 - very sticky                        23 - very plastic

  Enter choice: 10
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
            Type the OPTION NUMBER for the chosen criteria
                 PRESS RETURN - if you have no data
```

```
                          soil_classification
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  Enter FEATURES for which you have field data.for the C horizon.

       1 - thickness                           8  - colour patterns
       2 - colour                              9  - accumulation
       3 - texture                             10 - pans
       4.- consistency                         11 - concretions
       5 - structure
       6 - stoniness
       7 - boundary

  Enter enter choice one at a time: 5 <rtn> 9 <rtn> 10 <rtn> <rtn>
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
            Type the NUMBER for required option followed RETURN
            PRESS RETURN - by itself to indicate end
```

```
                              soil_classification
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  Describe any ACCUMULATION that may be present in the C horizon

       1 - calcium carbonate                 3 - soluble salts
       2 - calcium sulphate                  4 - illuvial clay coatings

  Enter choice: 4
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
             Type the OPTION NUMBER for the chosen criteria
             PRESS RETURN - if you have no data
```

```
                              soil_classification
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  Describe any PAN that may be present in the C horizon

       1 - clay pan                          3 - fragipan
       2 - iron pan                          4 - lime pan (caliche)
       2 - silicate pan

  Enter choice: 3
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
             Type the OPTION NUMBER for the chosen criteria
             PRESS RETURN - if you have no data
```

```
                              soil_classification
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  Describe the STRUCTURE for the C horizon

        GRADE OF DEVELOPMENT                      SIZE OF AGGREGATES
     structureless                                  very-fine(thin)
     moderately_developed                           fine(thin)
     weakly_developed                               medium
     strongly_developed                             coarse
                                                    very_coarse
                             SHAPE OF AGGREGATES
     massive                   single_grained       platy
     prismatic                 columnar             granular
     blocky                    nutty                crumb

  Enter description: strongly_developed_columnar
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
          Type description. Use UNDERSCORES between words(No Spaces)
                   PRESS RETURN - if you have no data
```

```
                          soil_classification
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  Which of the following soil groups do you think the soil may belong to ?

                     AZONAL and INTRAZONAL SOILS
               VOLCANIC                             SEDIMENTARY
  1 - recent soils from volcanic ash           8 - recent soils
  2 - yellow brown pumice                      9 - yellow brown sands
  3 - yellow brown loams                      10 - gley soils
  4 - brown granular loam                     11 - saline soils
  5 - brown granular clay                     12 - organic soils
  6- -red loam                                13 - rendzina
  7 - brown loam
                           ZONAL SOILS
             14 - brown grey earth            17 - podzol
             15 - yellow grey earth           18 - gley podzol
             16 - yellow brown earth

  Enter choice: 15
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
            Type the OPTION NUMBER for the chosen criteria
                 PRESS RETURN - if you have no data
```

```
                        yellow_grey_earth          soil forming factor
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


  Could the climate be described as having an annual moisture deficiency ?


  Enter choice: y
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
              YES - if true                      NO - if false         .
        PRESS RETURN - if you do not know
        H - for help
-------------------------------------------------------------------------------
  Certainty between (0 and 5): <rtn>
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
              1 - if very uncertain        3 - if it probably is       .
                  PRESS RETURN - for absolutely certain
                               H - for help
```

After each yes or no response the system requests the user to rank their belief in the accuracy of the data they are entering. Only the bottom prompt area of the screen changes. Following examples do not show this second prompt.

```
                    yellow_grey_earth          soil forming factor
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  Could the topography be described as terraces ?

  Enter choice: y
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
              YES - if true                      NO - if false          .
                    PRESS RETURN - if you do not know
                              H - for help
```

```
                    yellow_grey_earth          soil forming factor
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  Could the rock type be broadly classified as quartzo feldspathic ?

  Enter choice: <rtn>
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
              YES - if true                      NO - if false          .
                    PRESS RETURN - if you do not know
                              H - for help
```

```
                    yellow_grey_earth          soil forming factor
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  Is the dominant parent material loess ?

  Enter choice: y
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
              YES - if true                      NO - if false          .
                    PRESS RETURN - if you do not know
                              H - for help
```

```
                    SOIL IDENTIFICATION SYSTEM
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

     *******************************************************
     The soil you have described has been identified as a
     yellow grey earth
     with a certainty of 4.8 on a scale
     of -5 to 5

     A yellow grey earth is a zonal soil.
     *******************************************************

  Stop:<rtn>
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                       PRESS RETURN to stop
                       Type NO to continue.
```

```
                    SOIL IDENTIFICATION SYSTEM
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  Do you want to to save the data from this consultation ?

  Enter response: n
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                  YES - if true          NO - if false
                  PRESS RETURN - if you do not know
                            H - for help
```

```
                         EXITING FROM
                    SOIL IDENTIFICATION SYSTEM
```

B.2 Examples of the Help Screens

The help option is very basic and mostly displays the same central part of the screens used by the voluntary system.

```
                          yellow_grey_earth         soil forming factor
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                             HELP OPTION

  The following list gives the alternative values for the parent material.
  This list only contains those values known to this system.

      1 - volcanic                              7 - quartzo feldspathic
      2 - plutonic                              8 - siliceous
      3 - metamorphosed                         9 - micaceous
      4.- basalt                               10 - calcareous
      5 - andesite                             11 - non-calcareous
      6 - rhyolite                             12 - marble
                            13 - limestone

  Continue:<rtn>
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                         PRESS RETURN to continue
                             Type NO to stop.
```

```
                          yellow_grey_earth         soil forming factor
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                             HELP OPTION

      Depending on your degree of belief in either the correctness
                          or incorrectness
      in the value for the soil feature, enter the appropriate number:

                     between 1 (low degree of belief)
                     and     5 ( for completely true)
                       zero represents - don't know
                           Default value is 5.

  Continue:<rtn>
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                         PRESS RETURN to continue
                             Type NO to stop.
```

Appendix C: Selected Sources for Soils Knowledge.

Location: Johnson's property ("Ngakaror") behind nomestead, on Te Horo - Hautere Cross road, 6 km SSW of Otaki Railway Station. NZMS1 N57/666795.

Elevation: 52 m; Rainfall: 1150 mm; Slope 1°; Aspect: southwest; Parent Material: quartzo-feldspathic loess; Landform: High terrace (Tokomaru Terrace); Overall drainage class: well drained.

Ap	0-20 cm	dark brown (10Y3/3) silt loam; very friable; moderately weak ped strength; firm penentration resistence; semi-deformable failure; moderately devekoped fine subangukar blocky structure; many very fine roots; distinct wavy boundary,
Bw1	20-53 cm	yellowish brown (10YR5/4) clay loam with 10% casts of A; very friable; firm penetration resistence; semi-deformable failure; plastic; non sticky; moderately weak ped strength; moderately developed medium subangular blocky structure breaking to moderately fime granular and subangular vlocky structure; common very fine roots; strong reaction to NaF; diffuse boundary,
Bw2	53-83	brownish yellow (10YR6/6) clay loam; friable; sticky; very plastic; stiff penetration resistence; semi-deformable failure; moderately weak ped strength; moderately developed fine angular blocky structure and granular structure; few finw roots; weak reaction to NaF; indistinct wavy boundary,
Bwg	83-100+ cm	light yellowish brown (10YR6/4) clay loam; common distinct medium strong brown (10YR5/6) mottles and few faint pale olive (5Y6/3) mottles; friable; slightly sticky; plastic; stiff penetration resistence; semi deformable failure; moderately weak ped strength; modertely developed medium angular blocky structure; few fine roots; weak reaction to NaF; water flowing from rear face of pit above Bwg horizon.

Classification

NZ: Moderately leached intergrade between yellow-brown earths and yellow-brown loams.

USDA: Umbric Dytrochrept, fine-silty, mixed mesic.

Figure C.01 Typical Profile description.
(from Wilde and Palmer (1986), page 61)

Order of Profile Descriptions

1. Depth or thickness of horizon; 2. Colour; 3. Texture; 4. Consistence and porosity; 5. Structure; 6. Roots, etc.; 7. Kind of boundary.

Repeat for each horizon and record location, position in landscape and slope, etc., vegetation, parent material, together with climate, land use, etc.

Colour Patterns

	Mottles		Gammate Forms
Abundance (%)	*Size (mm)*	*Contrast*	*Shape*
Few, <2	Fine, <5	Faint	Subgammate
Many, 2–20	Medium, 5–15	Distinct	Gammate
Abundant, >20	Coarse, >15	Prominent	Net-gammate
Profuse, *c.* 100			

Consistence

Dry	*Moist*	*Wet*		*Cementation*
Loose	Loose	Nonsticky	Nonplastic	Weakly cemented
Soft	Very friable	Slightly sticky	Slightly plastic	Strongly cemented
Slightly hard	Friable	Sticky	Plastic	Indurated
Hard	Firm	Very sticky	Very plastic	
Very hard	Very firm			
Extremely hard	Extremely firm			

Structure

Grade of development

Structureless (massive, single grain), weakly developed, moderately developed, strongly developed

Shape and size of aggregates (mm)

Platy	*Prismatic and Columnar*	*Blocky and Nut*	*Granular and Crumb*
V. thin, <1	V. fine, <10	V. fine, <5	V. fine, <1
Thin, 1–2	Fine, 10–20	Fine, 5–10	Fine, 1–2
Medium, 2–5	Medium, 20–50	Medium, 10–20	Medium, 2–5
Thick, 5–10	Coarse, 50–100	Coarse, 20–50	Coarse, 5–10
V. thick, >10	V. coarse, >100	V. coarse, >50	V. coarse, >10

Stoniness

With stones < 7 per cent, stony 7 – 30 per cent, very stony > 30 per cent.

Horizon Boundaries

Sharp, almost a line; distinct < 1 in. (or < 10 per cent of horizon if thin); indistinct 1–3 in.; diffuse > 3 in.

Approximate Equivalents

mm.	5	10	15	20	50	100
in.	0·2	0·4	0·6	0·8	2	4

Figure C.02 Main soil profile features.
(from Taylor and Pohen (1970), page 227)

SITE/PROFILE CARD–GENERAL Dec 1984 2nd Edition

1

CARD TYPE	PROFILE NO.	PROJECT NO.	DATE	AUTHOR	SPARE 1
SPGI :	:	:	:	:	:

2

SURVEY	COUNTY
:	:

3

SOIL TYPE NAME	E/P
	: :

4

SOIL SERIES NAME	E/P
	: :

5

LOCATION			ANNUAL RAIN (mm)	MEAN TEMP ANNUAL (C)	MEAN TEMP WARM MONTH	MEAN TEMP COLD MONTH	ELEVATION (m)	SPARE 2
NZMS SERIES	MAP NO.	GRID REF						
:	:	:	:	:	:	:	:	:

6

POSITION IN LANDSCAPE / LANDFORM GENISIS

ELEMENT	Crest 1, Ridge 2, Slope 3, Floor 4, Scarp 5, Surface 6
LANDFORM	Mountain A, Steepslope B, Hill C, Valley D, Terrace E, Plain F, Delta G, Fan H, Mound I, Tidal flat J, Plateau K, Dune L, Levee M, Organic deposit N
LANDSCAPE	Mtn range 1, Steepland 2, Hill country weakly diss 3, Hill country mod diss 4, Hill country strongly diss 5, Rolling country 6, Flat country 7, Coastal system 8, Terrace system 9
LANDFORM GENISIS	Colluvial A, Eolian B, Fluvian C, Fluvio–Eolian D, Glacio–Fluvial E, Glacio–Lacustrine F, Glacio–Marine G, Lacustrine H, Marine I, Morrainal J, Rock K, Tephra L, Undifferentiated M

7

MICRO RELIEF	Flat 1, Forest dimples 2, Terracettes 3, Humocky 4
SLOPE – SHAPE	Planar 1, Convex 2, Concave 3
SLOPE – FACET	Crest 1, Shoulder 2, Backslope 3, Midslope 4, Toeslope 5
SLOPE	Angle :, Length (m) :, Aspect :
	% surface covered by rock outcrop :
STONINESS	% surface covered :
STONINESS – SIZE	Small 1, Medium 2, Large 3
STONINESS – KIND	Stones 1, Boulders 2
IMPROVEMENTS	Oversown 1, Fertilised 2, Ploughed 3, Irrigated 4, Drained 5, Subsoiled 6, Erosion control (wind) 7, Erosion control (water) 8

8

SLOPE MOVEMENT	% area affected :
SLOPE MOVEMENT – TYPE	Rockfall 1, Rockslump 2, Earthslump 3, Debris slide 4, Debris flow 5, Debris avalanche 6, Creep mantle 7, Rock slide 8, Solifluction lobes 9
	Soil erosion % topsoil loss :
FLOODING	Every year 1, Every 2 years 2, Every 5 years 3, >10 years 4
PROFILE DRAINAGE	V. poorly 1, Poorly 2, Imperfectly 3, Mod. well 4, Well 5, Somewhat excessive 6, Excessively 7
LAND USE	Urban A, Recreation B, Horticulture C, Orchard–vineyard D, Cropped E, Ley grass F, Long term grass G, Rough grazing H, Shelter belt I, Exotic forestry J, Forestry K, Unused–natural L, Unused–reverted M
VEG. TYPE	Wetland 1, Dryland grass 2, Dryland scrub 3, Dryland forest 4, Alpine 5, High country 6

9

VEGETATION SPECIES LIST (USE COMMON NAMES. LINK AS RED BEECH–SILVER BEECH–KAMAHI) :

10

PARENT MATERIAL

WEATHERING	Unweathered 1, Weakly weathered 2, Moderately w'thered 3, Strongly w'thered 4
IN SITU CONSISTENCE	Very soft 1, Soft 2, Firm 3, Stiff 4, Very stiff 5, Hard 6
CEMENTATION	Uncemented 1, Weakly cem. 2, Strongly cem. 3, Indurated 4
JOINT	Coherent 1, >10 cm between cracks 2, <10 cm between cracks 3
PARENT ROCK MODIFER (CONTINUED ON NEXT LINE)	Undifferentiated A
Plutonic + Volc. rocks + Gneiss	Normal B, Acidic C, Alkaline D, Basic E, Potassic F, Sodic G
Greywacke	Arkosic H, Arcosic–tuffaceous I, Siliceous J
Tephra	Coarse rhyolitic K, Fine rhyolitic L, Mixed rhyolitic + andesitic M, Coarse andesitic N, Fine andesitic O, Coarse basaltic P, Fine basaltic Q
Sedimentary rocks, loess, windblown sands, alluvial silt, clay	Bentonitic R, Calcareous S, Carbonaceous T, Crystalline U, Ferro magnesian V, Glauconitic W, Highly micaceous X, Highly siliceous Y, Quartzose Z, Quartzo-feldspathic 0

line 10 contd

PARENT MATERIAL CONTINUATION

PARENT ROCK MODIFER – Gravels	Andesitic 1, Greywacke 2, Mixed g'wacke granite 3, Mixed g'wacke granite schist 4, Mixed g'wacke schist 5, Schist 6, Ultrabasic 7, Alluvial 8
PARENT ROCK	Granite A, Gneiss B, Rhyolite C, Dacite D, Ignimbrite E, Andesite F, Andesitic basalt G, Diorite H, Basalt I, Ultra mafite J, Greywacke K, Argillite L, Mixed g'wacke argillite M, Schist N, Sandstone O, Silty sandstone P, Siltstone Q, Sandy siltstone R, Conglomerate S, Shelly limestone T, Limestone U, Marble V, Tephra W, Loess X, Loess from g'wacke + argillite Y, Loess from schist Z, Gravel in similar matrix 0, Gravel in dissimilar matrix 1, Gravel 2, Windblown sand 3, Sand 4, Silty sand 5, Sandy silt 6, Silt 7, Clay 8, Peat 9

11

PARENT ROCK MODIFER + PARENT ROCK (FREE FORMAT) :

Figure C.03 Page 1 of soil profile recording card for recording soil forming factors.

PROFILE DESCRIPTION 3

No.	Group	Attribute	Class	1	2	3	4	5	6	7	8	9	10
51	COATINGS, ETC	ABUNDANCE	None 0%	1	1	1	1	1	1	1	1	1	1
			Few 0–10%	2	2	2	2	2	2	2	2	2	2
			Common 10–50%	3	3	3	3	3	3	3	3	3	3
			Many >50%	4	4	4	4	4	4	4	4	4	4
52		DISTINCTNESS	Faint	1	1	1	1	1	1	1	1	1	1
			Distinct	2	2	2	2	2	2	2	2	2	2
			Prominent	3	3	3	3	3	3	3	3	3	3
53		TYPE	Clay	1	1	1	1	1	1	1	1	1	1
			Organic	2	2	2	2	2	2	2	2	2	2
			Fe/Mn	3	3	3	3	3	3	3	3	3	3
			Carbonate	4	4	4	4	4	4	4	4	4	4
54		COLOUR	Hue										
55			Value										
56			Chroma										
57	PANS	CONTIN	Continuous	1	1	1	1	1	1	1	1	1	1
			Discontinuous	2	2	2	2	2	2	2	2	2	2
			Broken	3	3	3	3	3	3	3	3	3	3
58		KIND	Ironpan	1	1	1	1	1	1	1	1	1	1
			Humuspan	2	2	2	2	2	2	2	2	2	2
			Claypan	3	3	3	3	3	3	3	3	3	3
			Fragipan	4	4	4	4	4	4	4	4	4	4
			Duripan	5	5	5	5	5	5	5	5	5	5
			Limepan	6	6	6	6	6	6	6	6	6	6
			Ploughpan	7	7	7	7	7	7	7	7	7	7
			Fe/Mn pan	8	8	8	8	8	8	8	8	8	8
59	LIVE ROOTS	ABUNDANCE	None	1	1	1	1	1	1	1	1	1	1
			Few 1–10/1–5 †	2	2	2	2	2	2	2	2	2	2
			Common 10–25/5–10	3	3	3	3	3	3	3	3	3	3
			Many 25–200/10–20	4	4	4	4	4	4	4	4	4	4
			Abundant >200/>20	5	5	5	5	5	5	5	5	5	5
60		SIZE	Very fine <1 mm	1	1	1	1	1	1	1	1	1	1
			Fine 1–2 mm	2	2	2	2	2	2	2	2	2	2
			Medium 2–5 mm	3	3	3	3	3	3	3	3	3	3
			Coarse >5 mm	4	4	4	4	4	4	4	4	4	4
61	COARSE FRAGMENTS	ABUNDANCE	None	1	1	1	1	1	1	1	1	1	1
			Few <5%	2	2	2	2	2	2	2	2	2	2
			Common 5–15%	3	3	3	3	3	3	3	3	3	3
			Many 15–35%	4	4	4	4	4	4	4	4	4	4
			Abundant 35–70%	5	5	5	5	5	5	5	5	5	5
			Profuse >70%	6	6	6	6	6	6	6	6	6	6
62		WEATHER	Unweathered	1	1	1	1	1	1	1	1	1	1
			Weakly w'thed	2	2	2	2	2	2	2	2	2	2
			Moderately	3	3	3	3	3	3	3	3	3	3
			Strong	4	4	4	4	4	4	4	4	4	4
63		ROUNDING	Rounded	1	1	1	1	1	1	1	1	1	1
			Subrounded	2	2	2	2	2	2	2	2	2	2
			Subangular	3	3	3	3	3	3	3	3	3	3
			Angular	4	4	4	4	4	4	4	4	4	4
64		SIZE	Gravels	1	1	1	1	1	1	1	1	1	1
			Stones	2	2	2	2	2	2	2	2	2	2
			Boulders	3	3	3	3	3	3	3	3	3	3
65	Parent rock code												
66	LOWER BOUND'Y	DISTINCTNESS	Sharp	1	1	1	1	1	1	1	1	1	1
			Distinct	2	2	2	2	2	2	2	2	2	2
			Indistinct	3	3	3	3	3	3	3	3	3	3
			Diffuse	4	4	4	4	4	4	4	4	4	4
67		SHAPE	Smooth	1	1	1	1	1	1	1	1	1	1
			Wavy	2	2	2	2	2	2	2	2	2	2
			Irregular	3	3	3	3	3	3	3	3	3	3
			Discontinuous	4	4	4	4	4	4	4	4	4	4

† roots per 100 cm^2 First range for fine and very fine/second for medium and coarse

Figure C.04 Page 2 of soil profile recording card for recording profile characteristics.

PHYSIOGRAPHIC LEGEND OF SOILS

SOILS OF THE RIVER FLATS

Rapidly Accumulating

Rangitikei loamy sand	1
Rangitikei loamy sand, shallow phase	1a
Rangitikei sandy loam	1b
Rangitikei mottled sandy loam	1c
Rangitikei fine sandy loam	1d
Rangitikei fine sandy loam, shallow phase	1e
Rangitikei mottled fine sandy loam	1f
Parewanui silt loam	2
Parewanui fine sandy loam	2a
Parewanui heavy silt loam	2b

Slowly Accumulating

Manawatu silt loam	3
Manawatu mottled silt loam	3a
Manawatu fine sandy loam	3b
Manawatu fine sandy loam, stony phase	3c
Manawatu mottled fine sandy loam	3d
Manawatu sandy loam	3e
Manawatu sandy loam, gravelly phase	3f
Manawatu mottled sandy loam	3g
Kairanga silt loam	4
Kairanga fine sandy loam	4a
Kairanga heavy silt loam	4b
Kairanga peaty silt loam	4c
Opiki peaty silt loam	5
Makerua peaty silt loam	6

Non-accumulating

Karapoti black silt loam	7
Karapoti black sandy loam	7a
Karapoti brown sandy loam	7b
Karapoti brown sandy loam, gravelly phase	7c
Te Arakura silt loam	8
Te Arakura fine sandy loam	8a
Te Arakura sandy loam	8b
Te Arakura sandy loam, shallow phase	8c

SOILS OF THE TERRACE LAND AND FANS

Soils of Terraces and Fans

Ohakea silt loam	9
Ohakea peaty loam	9a
Tokomaru silt loam	10
Tokomaru silt loam, rolling phase	10a
Milson silt loam	11
Marton silt loam	12
Marton silt loam, rolling phase	12a
Ashhurst silt loam	13
Ashhurst silt loam, stony phase	13a
Shannon silt loam	14
Shannon silt loam, rolling phase	14a

PEDOLOGICAL LEGEND OF SOILS

RECENT SOILS
from alluvium

Rapidly accumulating
- Rangitikei soils

Slowly accumulating
- Manawatu soils

Non-accumulating
- Karapoti soils

GLEY RECENT SOILS
from alluvium

Rapidly accumulating
- Parewanui soils

Slowly accumulating
- Kairanga soils
- Opiki soils

GLEY SOILS
from alluvium
- Te Arakura soils

ORGANIC SOILS
from peat and alluvium
- Makerua soils

YELLOW-GREY EARTHS
from loess, colluvium, and alluvium
- Ohakea soils
- Tokomaru soils
- Tokomaru hill soils
- Milson soils
- Marton soils

from sandstone, conglomerate, and loess
- Halcombe hill soils

Associated yellow-brown Shallow and Stony Soils
from loess, colluvium, and alluvium
- Ashhurst soils

STEEPLAND SOILS RELATED TO YELLOW-GREY EARTHS
from sandstone, conglomerate, and loess
- Halcombe steepland soils**

INTERGRADES BETWEEN YELLOW-GREY AND YELLOW-BROWN EARTHS
from loess and colluvium
- Shannon soils

from sandstone and loess
- Raumai hill soils

from sandstone, conglomerate, and loess

Figure C.05 Part of a soil map legend.
(from Cowie et al, 1972)

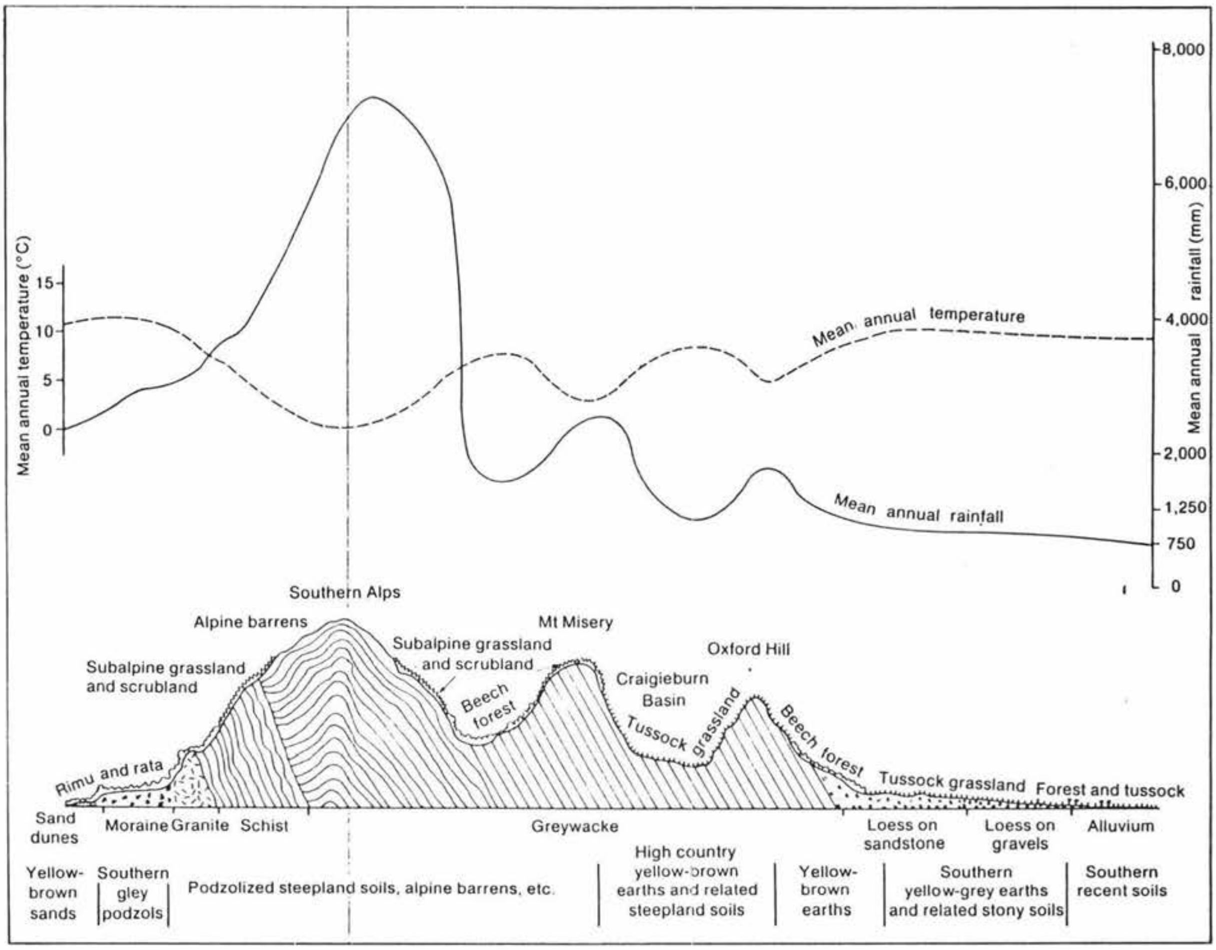


Figure C.06 Diagram showing the relationships between climate, topography and soil development. (from Gibbs (1980), page 54)

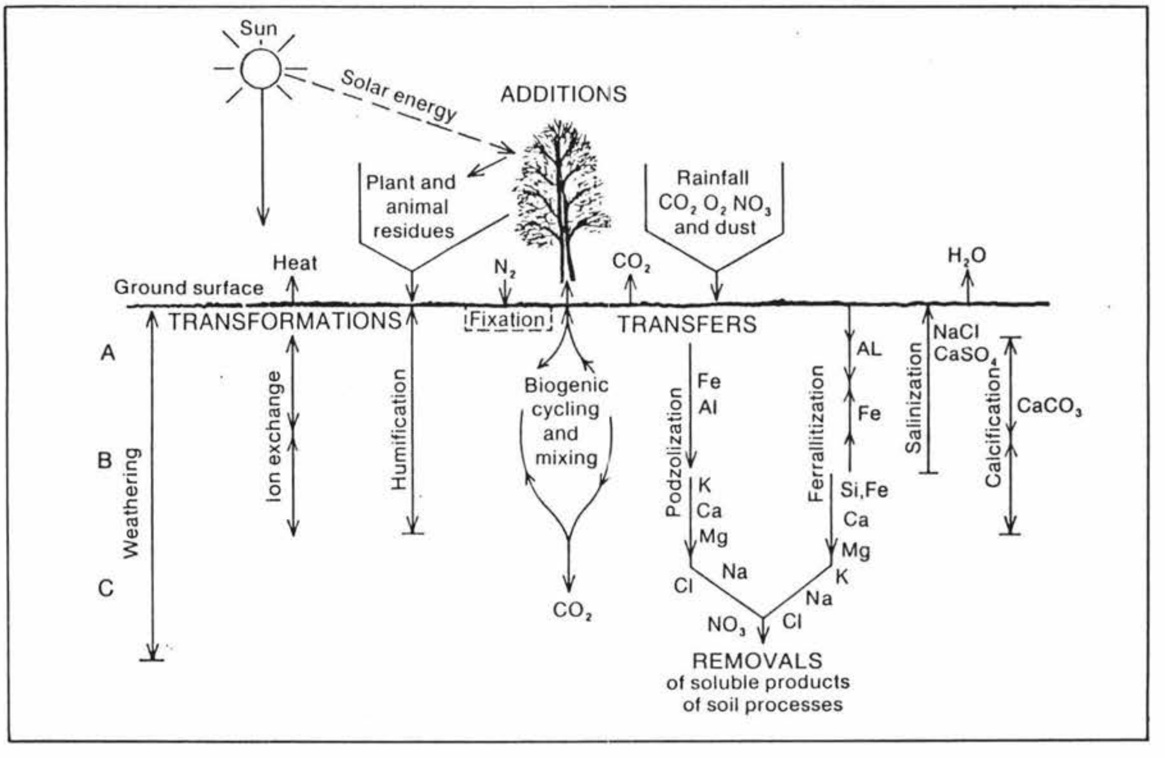


Figure C.07 Diagram showing some of the soil forming processes.
(from Gibbs (1980), page 16)

Appendix D: Design Aspects of the Generalization Plane for SES.

The generalization plane for SES is represented in the conceptual design by using the semantic net diagrams (figure D.2) and in the associated detail design by the hierarchy of frames for which the Classification Descriptor Schema is the root node. All schema are linked back to the Knowledge Structure Schema as shown in Figure G.1 which outlines examples from the hierarchy detailed in this appendix.

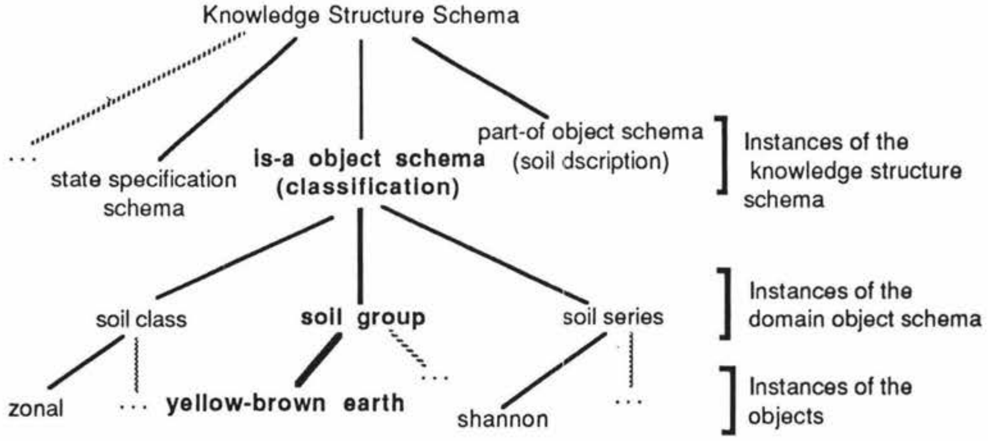


Figure D.01 Part of the Knowledge Structure Hierarchy with classification subtree highlighted.

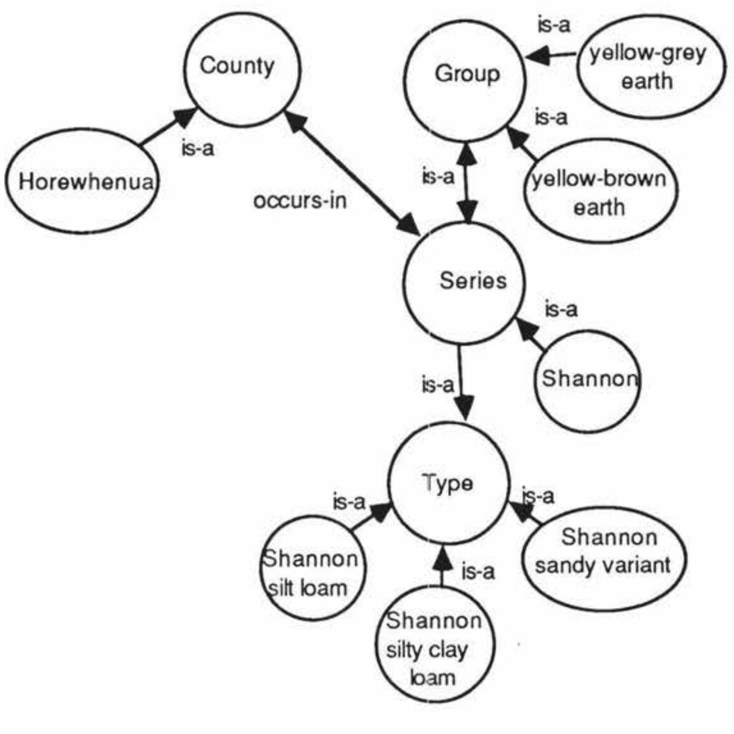


Figure D.02 Part of the Semantic Net for the Schema for the Generalization Plane.

frame-number: 201	
LINK SECTION	
Frame:	schema-schema
Name:	classification-descriptor
Instance-of:	knowledge structures
Instances:	(county-descriptor, class-descriptor, group-descriptor, series-descriptor, type-descriptor)
Instance-type	object-schema
INSTANCE-STRUCTURE SECTION	
Link:	(Frame : Instance-type) (Name : one-of(Instances)) (Instance-of : Name) (Parent : one-of(Instances) (Children : one-of(Instances) (Instances : list-of(Instances->Instances)) (Instance-type: is-a-object) (Cluster: soil-descriptor)
Instance-structure:	
Information:	(Translation : text) (Instance-Prompt : create-frame(Link.Instance-type)) (Date-created : date) (Author : text)
INFORMATION SECTION	
Translation:	This is a schema that describes the structure of the descriptors used to describe the classification units in the soils domain.
Instance-Prompt::	create-frame(object-schema)
Date-created:	21-4-88
Author:	Lis Todd

Figure D.03 Frame representing the classification schema which describes the common structures of the object-schemas for each classification unit.

frame-number: 205	
LINK SECTION	
Frame:	object-schema
Name:	series-descriptor
Instance-of:	classification-descriptor
Parent:	group-descriptor
Children:	type-descriptor
Instances:	(Manawatu series, Paraha series, Levin series,Shannon series, . . .)
Instance-type:	is-a-object
Cluster:	soil-descriptor
INSTANCE-STRUCTURE SECTION	
Link:	(Frame : "is-a object"), (Name:one-of(Instances)) (Instance-of: Name), (Parent:list-of(Parent>Instances), (Children:list-of(Children>Instances), (Associated:list-of(Instances), (Similar:list-of(Instances) (Cluster: (Link.Name), soil-descriptor))
Optional Features:	(No-of-soil-groups : (count(Parent)-1)) (Modifier :condition(if No-of-soil-groups > 1 then "Intergrade"))
Essential Features:	(list-of(in(Cluster -> site-descriptor))
Characteristic Features:	(list-of(in(Cluster -> profile-descriptor))
Compare Solution:	(list-of(Rules))
Identify type:	(list-of(Rules))
Information:	(Translation : text) (Explanation : text) (Date-created : date) (Author : text)
INFORMATION SECTION	
Translation:	This is a schema that describes the structure of the descriptors used to describe a specified soil series classification unit.
Instance-Prompt:	Create-frame (is-a-object)
Date-created:	21-4-88
Author:	Lis Todd

Figure D.04 Example of a classification object-schema. This frame is the Series Schema which describes the structure of the frames representing each of the Soil Series known to SES.

frame-number: 214	
LINK SECTION	
Frame:	is-a-object
Name:	Shannon Series
Instance-of:	series descriptor
Parent:	(Horowhenua, yellow-grey earth, yellow-brown earth)
Children:	(Shannon silt loam, Shannon silty clay loam, Shannon sandy variant)
Associated:	(Levin series, Koputaroa series)
Similar:	(Levin series, Paraha series,Koputaroa mottled sandy loam,Shannon variant)
Cluster :	(Shannon Series, soil-descriptor)
OPTIONAL FEATURES	
No-of-soil-groups :	2
Modifier :	"Intergrade"
ESSENTIAL FEATURES	
Site-descriptor:	(Drainage = "imperfectly drained")
	(Slope = range(0,7))
CHARACTERISTIC FEATURES	
A-horizon:	(colour = "dark brown") (consistency = "friable") (texture = "silt loam") (structure = "moderately developed nut")
B-horizon:	((colour = "yellowish brown")&(colour = "olive brown") (texture = "silty clay loam") (structure = "weak blocky")
Colour-patterns:	(Low-chroma-colours : range(30,60)&("net gammat patterns")
COMPARE SOLUTION	
Rule 01:	if less(drainage) & less(Low-chroma-colours : range(30,60),5%) then consider(Levin Series)
Rule 02:	if same(parent-material , "sand") then consider(Koputaroa mottled sandy loam)
Rule 03:	if not-same(parent-material) & not-same(physiographic-position) then consider(Paraha Series)
Rule 04	if (greater(stony) or "lenses of sand") then consider(Paraha Series)
Rule 05	if greater(colour,greyer) & present(E-horizon) then consider(Shannon variant)
IDENTIFY TYPE	
Rule 11	if greater(certainty,4) & similar(texture,"silt loam") then consider(Shannon silt loam)
Rule 12:	if same(A-horizon->(texture, "sandy")) then consider(Shannon sandy variant)
Rule 13	if same(A-horizon->(consistency, "sticky")) then consider(Shannon silty clay loam)
INFORMATION SECTION	
Translation:	This frame describes the Shannon.series of soils. It links into the generalization hierarchy that represents the soil classification hierarchy. The range of profile features are described in the linked soil descriptor part-of object structure.
Explanation :	Shannon soils describe soils formed from loess on gentle rolling slopes of the dissected terrace land and on sloping fans. the rainfall is slightly higher and more evenly distributed than the similar Tokomaru series. These soils are moderately gleyed soils developed in at least 50 cm of loess on either gravels, sands, colluvium or older loess. They are imperfectly to poorly drained with low chroma colours within 60cm of the surface.
Date-created:	23-4-88
Author:	Lis Todd

Figure D.05 Example of an instance of a Series Schema. This frame describes the Shannon Series of soils.

Appendix E: Design Aspects of the Aggredation Plane for SES.

The aggregation plane for SES is represented in the conceptual design by using a series of semantic net diagrams representing the features required to describe a soil. The associated detail design is represented by a hierarchy of frames for which the Soil Description Schema is the root node but the leaf nodes also link to the Type Descriptor Schema. Features such as drainage are relatively straight forward and a simple descriptive frame is sufficient. Figure G.1 outlines the examples from this heirarchy which are detailed in this appendix.

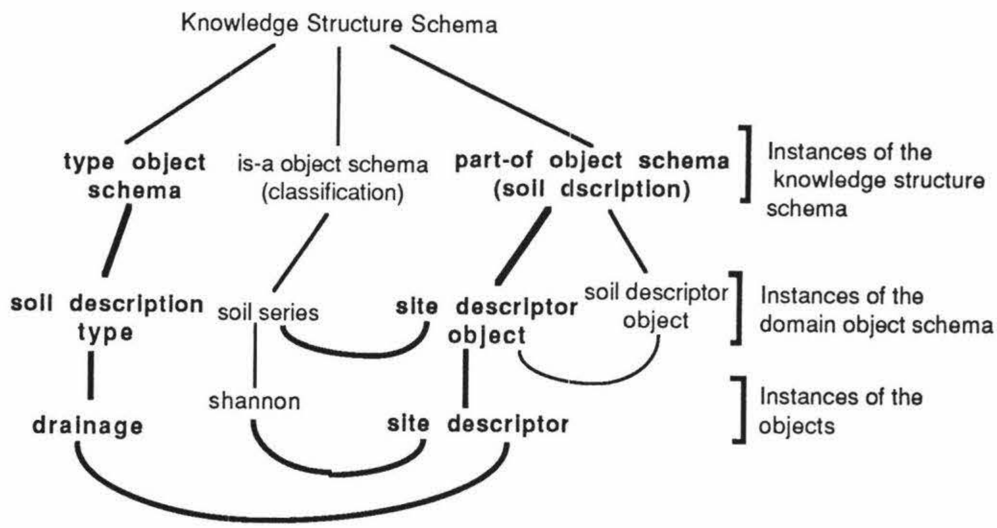


Figure E.01 Part of the Knowledge Structure Hierarchy with soil description subtree highlighted.

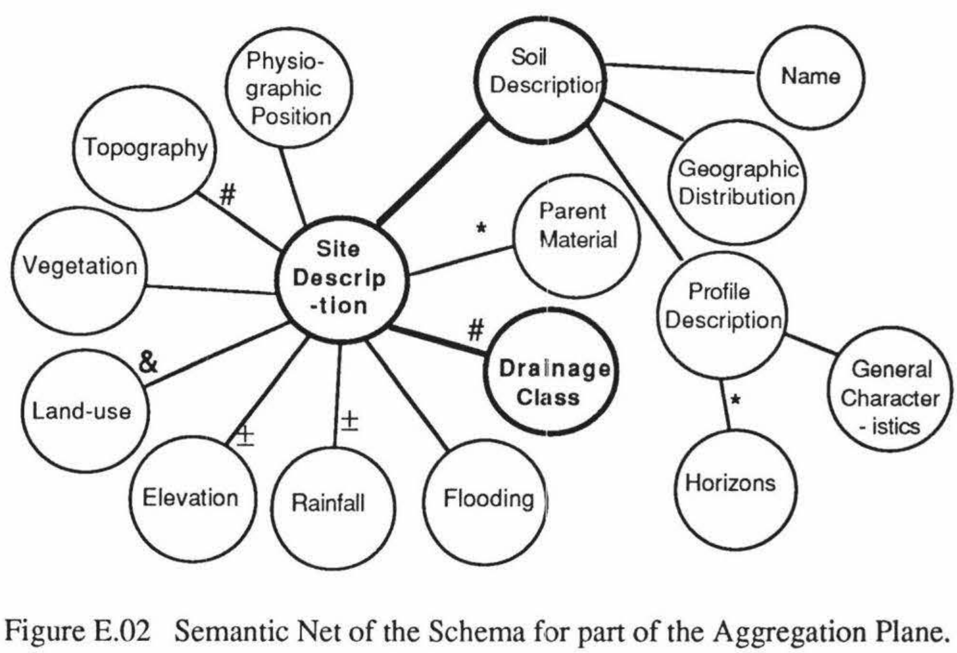


Figure E.02 Semantic Net of the Schema for part of the Aggregation Plane.

frame-number: 301	
LINK SECTION	
Frame:	schema-schema
Name:	soil-description-schema
Instance-of:	knowledge-structure-schema
Instances:	(soil-descriptor, site-descriptor, profile-descriptor, horizon-descriptor,...)
Instance-type	object-schema
Instance-root:	soil-descriptor
INSTANCE-STRUCTURE SECTION	
Link :	(Frame: Instance-type) (Name: (Link.Cluster, one-of(Instances) (Instance-of: Name) (Parent: (Link.Cluster,Link.Name) (Children: (Link.Cluster, one-of(Instances) (Instances: (list-of (Link.Cluster.Instances->Instances,Link.Name)) (Instance-type: part-of-object) (Cluster: (if Instance-root then one-of(Classification-descriptor->Instances)) (if not Instance-root then Parent->Cluster) (Instance-root: (Link.Cluster, soil-descriptor)
Instance-structure:	
Information :	(Translation : text) (Instance-Prompt : create-frame(Link.Instance-type)) (Explanation : text) (Date-created : date) (Author : text)
INFORMATION SECTION	
Translation:	This frame describes the structure
Prompt :	create-frame(object-schema)
Date-created:	24-4-88
Author:	Lis Todd

Figure E.03 Frame representing the soil description schema which describes the common structures of the object-schemas for each object in a soil descriptor.

frame-number: 302	
LINK SECTION	
Frame:	object-schema
Name:	(series-descriptor, site-descriptor)
Instance-of:	soil-description-schema
Parent:	(series-descriptor, soil-descriptor)
Children:	(series-descriptor, parent-material-descriptor)
Instances:	((Manawatu series, site-descriptor), (Paraha series, site-descriptor), (Levin series, site-descriptor),(Shannon series, site-descriptor), . . .)
Instance-type	part-of-object
Cluster:	series-descriptor
Instance-root:	(series-descriptor,soil-descriptor)
INSTANCE-STRUCTURE SECTION	
Link :	(Frame: Instance-type) (Name: one-of(Instances) (Instance-of: Name) (Parent: (Link.Cluster,soil-descriptor) (Children: (Link.Cluster,parent-material-descriptor) (Cluster: (if Instance-root then one-of(Cluster->Instances)) (if not Instance-root then "Parent->Cluster")
Mandatory :	(Physiographic-position : list-of(physiographic-type)) (Topography : list-of(range(topographic-type)) (Vegetation : vegetation-type) (Land use : land-use-type) (Elevation : range(elevation-type)) (Rainfall : range(rainfall-type)) (Drainage Class : range(drainage-type)) (Parent Material: in (Link.Children(parent-material-descriptor))
Information :	(Translation : text) (Explanation : text) (Date-created : date) (Author : text)
INFORMATION SECTION	
Translation:	This frame describes the structure
Prompt :	create-frame(part-of-object)
Date-created:	24-4-88
Author:	Lis Todd

Figure E.04 Example of an instance of a site descriptor. This frame describes the general features required to characterize one of the soil Series.

frame-number: 344	
LINK SECTION	
Frame:	part-of-object
Name:	(Shannon-Series,site-descriptor)
Instance-of:	(series-descriptor, site-descriptor)
Parent:	(Shannon-Series,soil-descriptor)
Children:	(Shannon-Series,parent-material-descriptor)
Cluster:	Shannon-Series
MANDATORY-SECTION	
Physiographic-position :	[[high, terraces, bordering , Hautere Plains] [high terraces, North, Otaki River, to, Manakau]]
Topography :	[[high, terraces] [terraces, older than, Ohakea terrace]]
Vegetation :	"pasture grasses"
Land use :	"mixed pastoral - dairy, beef, sheep, goats, deer " "scattered horticulture"
Elevation :	10 - 100 meters
Rainfall :	950 - 1200+ millimeters
Parent-material :	in(Shannon-Series,parent-material-descriptor)
Drainage Class :	" imperfectly drained "
INFORMATION SECTION	
Translation:	This frame describes the structure
Explanation :	Shannon
Date-created:	24-4-88
Author:	Lis Todd

Figure E.05 Example of an instance of a site descriptor. This frame describes the site of a Shannon Series soil.

Type 401	
LINK SECTION	
Frame:	type-object-schema
Name:	soil-description-type
Instance-of:	knowledge-structures
Instances:	(drainage, . . .)
Instance-type:	type-object
INSTANCE-STRUCTURE SECTION	
Link:	(Frame: type-object) (Name: text) (Parent Frame: one-of(part-of-object)) (Values: list)
Range information:	(least: one-of(Link.Values)) (greatest: one-of(Link.Values)) (list-of(rules))
Special terms:	(text: one-of (list-of(Link.Values) (text: list-of(Rules)
Infer:	(list-of(rules))
Clarify:	(list-of(rules))
Explanation:	(Feature: text) (Each (Link.Values): text) (Reference: source of explanation text)
Information :	(Translation : text) (Prompt: value-prompt) (Date-created : date) (Author : text)
INFORMATION SECTION	
Translation:	Drainage is a Soil Forming Factor and is recorded as one of the features in the site description.
Instance-Prompt:	Create-frame(type)
Date-created:	24-4-88
Author:	Lis Todd

Figure E.06 Frame representing the Type-object Schema which describes the common structures of the object-schemas for each type required by the SES.

Type 404	
Frame:	type-objects
Name:	drainage
Parent Frame:	Site-descriptor
Values:	('very poorly drained','poorly drained','imperfectly drained','moderately well drained','well drained','somewhat excessively drained','excessively drained')
RANGE INFORMATION	
least:	'very poorly drained'
greatest:	'excessively drained'
SPECIAL TERMS	
"good drainage":	one-of('moderately well drained','well drained','somewhat excessively drained')
INFER	
Rule 01:	if same(A-horizon->(texture,peaty) & same(B-horizon->(colour,low-chroma-colours) & similar((topography, hollow) then (drainage,'very poorly drained',0.9)
Rule 02:	if similar(B-horizon->(colour,"low chroma colours") & similar((topography, one-of(hollow, flat, depression) then (drainage, 'poorly drained',0.8)
Rule 03	if present(profile,"slowly permeable layer") & present(B-horizon->(colour,"low chroma colours") & then (drainage, 'imperfectly drained',0.70)
Rule 04	if similar(A-horizon->(texture,loam)) then (drainage, 'well drained',0.60)
Rule 05	if similar(A-horizon->(texture,sandy)) & less(A-horizon->(boundary, indistinct) & greater(profile,porous) then (drainage, 'somewhat excessively drained',0.90)
Rule 06	if (greater(slope, 15) & less(A-horizon->(thickness,10)) OR same(profile,"very porous") then (drainage, 'excessively drained',0.90)
CLARIFY	
Rule 01	if ask("Does water pond on site frequently during winter") then (drainage,'very poorly drained',0.85)
Rule 02	if ask("Does water pond on site after moderate rain") & ask("Does the ponded water usually drain within 12 hours") & ask("Do post holes usually fill with water ") then (drainage,'poorly drained',0.85)
EXPLANATION	
Feature:	Drainage as a condition of the soil, refers to the frequency and duration of periods when the soil is free of complete or partial saturation. The drainage class refers to the average state of drainage over a period of time, usually a year.
'very poorly drained':	The water is removed from the soil so slowly that the water table remains at or on the surface the greater part of the time.
'poorly drained':	The water is removed so slowly that the soil remains at field capacity for a large part of the time. The water table is commonly at or near the surface during a considerable part of the time.
'imperfectly drained':	The water is removed slowly enough to keep it at field capacity for significant periods but not all of the time.
'moderately well drained':	The water is removed readily but not rapidly from the soil which consequently are not wet for a small but significant part of the time.
'well drained':	The water is removed readily but not rapidly from the soil, which consequently are not wet for a significant part of the time although they commonly retain near optimum amounts of moister for lengthy periods.
'somewhat excessively drained':	The water is removed from the soil rapidly, so that moister deficiency frequently limits plant growth.
'excessively drained':	The water is removed from the soil very rapidly and little is retained.
Reference:	Soil Survey Method (N.Z. Soil Bur. Bull. 25)
INFORMATION SECTION	
Prompt:	'How would you describe the overall drainage of the soil at the site ?'
Translation:	Drainage is a Soil Forming Factor and is recorded as one of the features in the site description.
Date-created:	24-4-88
Author:	Lis Todd

Figure E.07 Example of an instance of a Type-object Schema. This frame describes the soil feature, drainage.

Appendix F: Functional Design Aspects of SES

The functional design for SES is represented in the conceptual design by using the State Specification diagram (figure F.2) and in the associated detail design by the hierarchy of objects for which the State Schema is the root node. Figure F.1 outlines the examples from this heirarchy which are detailed in this appendix.

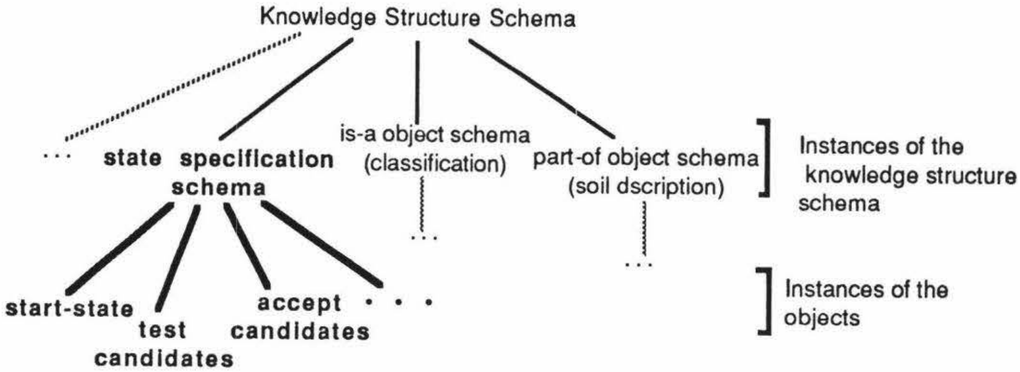


Figure F.01 Part of the Knowledge Structure Hierarchy with part of the state specification subtree highlighted.

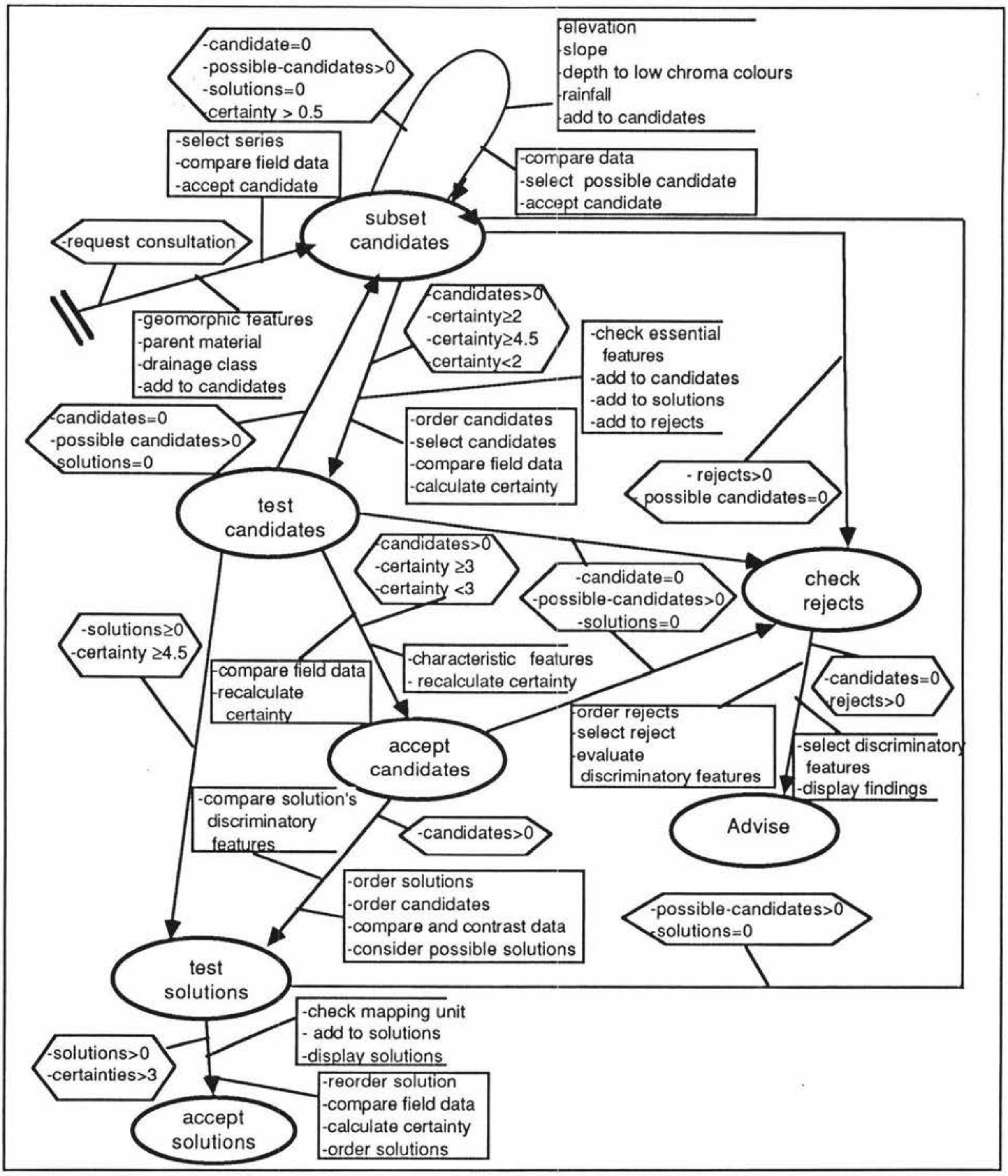


Figure F.02 State Specification diagram for SES

frame-number: 101	
LINK SECTION	
Frame:	schema-schema
Name:	state-schema
Instance-of:	knowledge structures
Instances:	(start, subset-candidates, test-candidates, accept-candidates, test-solutions, accept-solutions, check-rejects, advise)
Instance-type:	state
INSTANCE-STRUCTURE SECTION	
Link:	(Frame : Instance-type), (Name : one-of(Instances)) (Instance-of : Name) (Input-list: all ((one-of(classification-descriptor->Instances)->Instances) or list-of (candidates , possible-candidates, solutions, rejects:list) (Output-list: list-of(candidates , possible-candidates, solutions, rejects : list))
Conditions:	(Invoke: list-of (request, condition : operator(count(one-of(Input-list, Output-list)), constant)) (Output-membership: list-of(add-to(one-of(Output-list),Rule or (condition operator(certainty-factor, numeric-constant)))
Steps:	list-of(domain-feature : in(object-descriptor))
Operations:	list-of(actions) list-of(rules)
Next state:	list-of(rules)
Information:	(Translation : text), (Date-created : date), (Author : text)
INFORMATION SECTION	
Translation:	This frame describes the structure of the state frames used to describe the different states the identification system can be in.
Instance-prompt:	create-frame(state)
Date-created:	24-4-88
Author:	Lis Todd

Figure F.03 Frame describing the Root Schema of the State Schema Hierarchy.

frame-number: 102		
LINK SECTION		
Frame:	state	
Name:	start-state	
Instance-of:	state-descriptor	
Input-list:	(Manawatu series, Paraha series, Levin series,Shannon series, . . .)	
Output-list:	(candidates , possible-candidates)	
CONDITIONS		
Invoke:	consultation-requested	
Output-membership:		add-to(candidates,Rule(01) or Rule(02)) ((add-to(possible-candidates, (Rule(03) or Rule(05)) and remove-from(candidates, (Rule(03) or Rule(04)))
STEPS		
Geographic-position:		in(geographic-position-descriptor)
Physiographic-position:		in(Physiographic-position-descriptor)
OPERATIONS		
Rule(01):	if then	same(geographical-position) Output-membership
Rule(02):	if then	same(Physiographic-position) Output-membership
Rule(03):	if then	member(&current-object,candidate) & not(similar(geographical-position) Output-membership
Rule(04):	if then	member(&current-object,candidates) & not(similar(Physiographic-position)) Output-membership
NEXT STATE		
Rule(11)	if then	greater(count(candidates) , 0) invoke(test-candidates)
Rule(12)	if then	equals(count(candidates) , 0) & equals(count(solutions) , 0) & greater(count(possible-candidates) , 0) invoke(subset-candidates)
INFORMATION SECTION		
Translation:	This frame describes the start state in the inference process for identifying a specific soil from the field data collected.	
Date-created:	23-4-88	
Author:	Lis Todd	

Figure F.04 Frame describing the Start State for a consultation in SES.

frame-number: 103		
LINK SECTION		
Frame:		state
Name:		test-candidates
Parent:		state-schema
Input-list:		candidates
Output-list:		(candidates,rejects,possible-solutions)
CONDITIONS		
Invoke:		greater(count(candidates) , 0)
Output-membership:		add-to(possible-solutions, greater-or-equal(certainty , 4.5)) add-to(candidates, greater-or-equal(certainty , 2.5)) add-to(rejects, less(certainty , 2.5))
STEPS		
Essential-features:		in(current-series -> "ESSENTIAL FEATURES")
Drainage-class:		in(&current-object->drainage-class)
OPERATIONS		
Order-list:		on(certainty-factor , candidates)
Rule(01):	if then	member(&current-object,candidates) & similar(drainage) Output-membership
Rule(02):	if then	member(&current-object , candidates) & similar(essential-features) Output-membership
NEXT STATE		
Rule(11)	if then	equals(count(candidates) , 0) & greater(count(rejects) , 0) & invoke(check-rejects)
Rule(12)	if then	greater(count(candidates) , 0) invoke(accept-candidates)
Rule(13)	if then	greater(count(possible-solutions) , 1) invoke(test-solutions)
INFORMATION SECTION		
Translation:		This frame describes the state frame test-possible-candidates which checks candidate solutions as possible classification units for a specified soil description.
Date-created:		23-4-88
Author:		Lis Todd

Figure F.05 Frame describing the state for testing whether the essential features of the candidate solutions match those of the field data.

frame-number: 104		
LINK SECTION		
Frame:	state	
Name:	accept-candidates	
Parent:	state-schema	
Input-list:	candidates	
Output-list:	(candidates, rejects)	
CONDITIONS		
Invoke:		greater(count(candidates) , 0)
Output-membership:		add-to(candidates, greater-or-equal(certainty , 3)) add-to(rejects, less(certainty , 3))
STEPS		
Characteristics-of-profile:		in(current-series -> "CHARACTERISTIC FEATURES")
OPERATIONS		
Order-list:		on(certainty-factor , candidates)
Rule(01):	if then	member(current-series , candidates) & similar(essential-features) Output-membership
NEXT STATE		
Rule(11)	if then	equals(count(candidates) , 0) invoke(check-rejects)
Rule(12)	if then	greater(count(candidates) , 0) invoke(confirm-solutions)
INFORMATION SECTION		
Translation:	This frame describes the state frame test-candidates which checks candidate solutions as possible classification units for a specified soil description.	
Date-created:	23-4-88	
Author:	Lis Todd	

Figure F.06 Frame describing the state for checking that the profile characteristics of the candidate solutions match those of the field data.

frame-number: 105		
LINK SECTION		
Frame:	state	
Name:	test-solutions	
Parent:	state-schema	
Input-list:	multiple(candidates,solutions)	
Output-list:	solutions	
CONDITIONS		
Invoke:		greater(count(candidates) , 0) or greater(count(solution),0)
Output-membership:		add-to(solutions, greater-or-equal(certainty , 3))
STEPS		
Discriminatory-features:		in(current-series -> "COMPARE SOLUTION")
OPERATIONS		
Order-list:		on(certainty-factor , candidates)
Order-list:		on(certainty-factor , solutions)
Rule(01):	if then	member(current-series , solutions) & check(Discriminatory-features) Output-membership
Rule(02):	if then	member(current-series , candidates) & check(Discriminatory-features) Output-membership
NEXT STATE		
Rule(11)	if then	greater(count(solutions) , 0) invoke(accept-solutions)
INFORMATION SECTION		
Translation:	This frame describes the state frame confirm-solutions which checks candidate solutions as possible classification units for a specified soil description.	
Date-created:	23-4-88	
Author:	Lis Todd	

Figure F.07 Frame describing the state for checking associated and similar solutions which may better match the field data.

frame-number: 106		
LINK SECTION		
Frame:	state	
Name:	accept-solutions	
Parent:	state-schema	
Input-list:	solutions	
Output-list:	solutions	
CONDITIONS		
Invoke:		greater(greater(count(solution),0)
Output-membership:		add-to(solutions, greater-or-equal(certainty , 3))
STEPS		
Identify-mapping-unit:		in(current-series -> "IDENTIFY TYPE")
OPERATIONS		
Order-list:		on(certainty-factor , solutions)
Rule(01):	if then	member(current-series , solutions) & check(Identify-mapping-unit) Output-membership
Rule(02)	if then	greater(count(solutions) , 0) display(solutions)
NEXT STATE		
INFORMATION SECTION		
Translation:	This frame describes the state frame confirm-solutions which checks candidate solutions as possible classification units for a specified soil description.	
Date-created:	23-4-88	
Author:	Lis Todd	

Figure F.08 Frame describing the state for checking frames lower in the solution heirarchy.

frame-number: 107	
LINK SECTION	
Frame:	state
Name:	check-rejects
Parent:	state-schema
Input-list:	rejects
Output-list:	rejects
CONDITIONS	
Invoke:	equals(count(candidates) , 0) & equals(count(solutions),0) & equals(count(possible-candidates),0) & greater(count(rejects),0)
STEPS	
OPERATIONS	
NEXT STATE	
Rule(11)	if greater(count(rejects),0) & request(advice) then invoke(advise)
INFORMATION SECTION	
Translation:	This frame describes the state frame check-rejects which asks the user whether they would like advice on further data collection when no solutions found.
Date-created:	24-4-88
Author:	Lis Todd

Figure F.09 Frame for describing conditions for asking the user if they would like advise.

References

Aikins, J. S. Prototypical Knowledge for Expert Systems. *Artificial Intelligence*, **20**, pp. 163-210, (1983).

Baker, J. D. Dipmeter Advisor : An Expert Log Analysis System at Schlumberger. *In* "The AI business" (P. H. Winston and K. A. Prendergast, eds.). The MIT Press, Cambridge, Massachusetts, pp. 51-66, (1984).

Barton, J. Seminar communication. *At* "The Second New Zealand Conference on Expert Systems (NZES'87)", (1987).

Bennett, J. S. and Englemore, R. SACON: a knowledge-based consultant for structural analysis. *Proceedings of the Sixth International Joint Conference on Artificial Intelligence (IJCAI-79)*, pp. 47-49, (1979).

Bonnet, A. *Artificial Intelligence: Promise and Performance.* Prentice Hall International, pp. 221, (1985).

Boose, J. Rapid Acquisition and Combination of Knowledge from Multiple Experts in the same Domain. *Proceedings of the International Conference on Future Advances in Computing,* (1986).

Brachman, R. J., Amarel, S., Engelman, C., Englemore, R. S., Feigenbaum, E. A. and Wilkins, D. E. What Are Expert Systems? *In* "Building Expert Systems." (F. Hayes-Roth, D. A. Waterman, and D. B. Lenat, eds.). Addison-Wesley, Reading, Massachusetts, pp. 31-57, (1983).

Brakto, I. *Prolog: Programming for Artificial Intelligence.* Addison-Wesley, Great Britain, (1986).

Brodie, M. L. On the Development of Data Models. *In* "On Conceptual Modelling." (M. L. Brodie, J. Mylopoulos and J. W. Schmidt, eds). Springer-Verlag, New York Inc., pp. 3-17, (1986).

Buchanan, B. G. and Duda, R. O. (1983). Principles of Rules-Based Expert Systems. *Advances in Computers,* **22**, pp. 163-216.

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

Buchanan, B. G. and Shortliffe, E. H. Uncertainty and Evidential Support. *In* "Ruled Based Expert Systems: The MYCIN experiments of the Stanford Heuristic Programming Project." (B. G. Buchanan. and E. H. Shortliffe, eds.). Addison-Wesley, Reading, Massachusetts, pp. 209-232, (1984).

Building Expert Systems. Edited by F. Hayes-Roth, D. A. Waterman, and D. B. Lenat. Addison-Wesley, Reading, Massachusetts, pp. 422, (1983).

Bylander T, and Chandrasekaran B. Generic tasks for knowledge-based reasoning: the "right" level of abstraction for knowledge acquisition. *International Journal of Man-Machine Studies*, **26**, pp. 231-343, (1987).

Cendrowska, J. and Bramer, M. A. A Rational Reconstruction of the MYCIN System. *International Journal of Man-Machine Studies*, **20**, pp. 229-317, (1984).

Chen, P. P. The Entity-Relationship Model: Toward a Unified View of Data. *ACM Transactions on Database Systems*, **1(1)**, (1976).

Clancey, W. J. Tutoring Rules for Guiding a Case Method Dialogue. *International Journal of Man-Machine Studies*, **11**, pp. 25-49, (1979).

Clancey, W. J. The Epistemology of a Rule-Based Expert System. - A Framework for Explanation. *Artificial Intelligence*, **20**, pp. 215-251, (1983).

Clancey, W. J. Extensions to Rules for Explanation and Tutoring. *In* "Ruled Based Expert Systems: The MYCIN experiments of the Stanford Heuristic Programming Project." (B. G. Buchanan. and E. H. Shortliffe, eds.). Addison-Wesley, Reading, Massachusetts, pp. 209-232, (1984).

Clancey, W. J. Heuristic Classification. *Artificial Intelligence,* **27**, pp. 289-350, (1985).

Clancey, W. J. From GUIDON to NEOMYCIN and HERACLES in Twenty Short Lessons: ORN Final Report 1979-1985. *The AI Magazine,* **7**(3), pp. 40-60, (1986).

Cooke, N. M. and McDonald, J. E. A Formal Methodology for Acquiring and Representing Expert Knowledge. *BYTE*, Sept, pp.238-281, (1986).

Cowie, J. D., Kear, B. S.and Orbell, G. E. Soil mpa of Kairanga County, North Island, New Zealand. (Scale 1:63,360; 1 inch to 1 mile). *N.Z. Soil bureau Map 102.* (1972).

Davis, R. Interactive Transfer of Expertise: Acquisition of New Inference Rules. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI-77)*, Cambridge, Massachusetts, pp. 321-328, (1977).

Davis, R. Knowledge Acquisition in Rule-Based Systems - Knowledge about Representations as a basis for System Construction and Maintenance. *Reprinted in* "Tutorial: Context Directed Pattern Recognition and Machine Intelligence Techniques for Information Processing " (1982). (Y. Pao and G. W. Ernst, eds), pp. 303-337, (1978).

Davis, R. and Buchanan, B. G. Meta-Level Knowledge: Overview and Applications. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI-77)*, Cambridge, Massachusetts, pp. 920-927, (1977).

Davis, R. and Buchanan, B. G. Meta-Level Knowledge. *In* "Ruled Based Expert Systems: The MYCIN experiments of the Stanford Heuristic Programming Project." (B. G. Buchanan and E. H. Shortliffe, eds.). Addison-Wesley, Reading, Massachusetts, pp. 507-529, (1984).

Davis, R., Buchanan, B. G. and Shortliffe, E. H. Production Rules as a Representation for a Knowledge-Based Consultation Program. *Artificial Intelligence* **8**(1), pp. 15-45, (1977).

Dryer, R. Why Am I Doing This? *AI Applications in Natural Resource Management* **2**(1), pp. 72-73, (1988).

Duda, R., Hart, P., Nilsson, N. J. and Sutherland G. L. Semantic Network Representation in Rule-based Inference Systems. *In* "Pattern-Directed Inference Systems". (D. A. Waterman and F. Hayes-Roth, eds), Academic Press, Inc, pp. 203-221, (1978).

Duda, R., Gaschnig, J. and Hart, P. Model Design in the Prospector Consultant System for Mineral Exploration. *In* "Expert Systems in the Micro Electronic Age". (D. Michie, ed), Edinburgh University Press, pp. 153-167, (1979).

Duda R., and Gacshnig, H. Knowledge-Based Expert Systems come of Age. *BYTE*, **Sept**, pp. 238-281, (1981).

Erlandsen, J. and Holm, J. Intelligent Help Systems. *Information and Software Technology*, **29**(3), pp. 115-121, (1987).

Fikes, R. and Kehler, T. The Role of Frame-based Representation in Reasoning. *Communications of the ACM.*, **28**(9), pp. 920, (1985).

Forsyth, R. Fuzzy Reasoning Systems. *In* "Expert Systems". (R. Forsyth, ed). Chapman and Hall Computing, London, pp. 51-61, (1984).

Friedland, P. Special Section on Architectures for Knowledge-based Systems (editorial). *Communications of the ACM.*, **28**(9), pp. 920, (1985).

Gane, C., and Sarson, T. *Structured Systems Analysis: Tools and Techniques.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, pp. 241, (1979).

Ganascia, J. G. Using an Expert System in merging qualitative and quantitative data analysis. *International Journal of Man-Machine Studies*, **20**, pp. 319-330, (1984).

Genesereth, M. R. and Ginsberg, M. L. Logic Programming. *Communications of the ACM.,* **28**(9), pp. 933-941, (1985).

Gibbs, H. S. *New Zealand Soils: An introduction.* Oxford University Press, Wellington, pp.117, (1980).

Harmon, P and King, D. *Expert Systems - Artificial Intelligence in Business.* A Wiley Press Book, New York, pp. 283, (1986).

Hayes-Roth, B. A Blackboard Architecture for Control. *Artificial Intelligence* **26**, pp. 251-321, (1985).

Hayes-Roth, F. Rule-Based Systems. *Communications of the ACM.*, **28**(9), pp. 921-932, (1985).

Hayes-Roth, F.,, Waterman, D. A. and Lenat, D. B. An Overview of Expert Systems. *In* "Building Expert Systems." (F. Hayes-Roth, D. A. Waterman, and D. B. Lenat, eds.). Addison-Wesley, Reading, Massachusetts, pp. 3-29, (1983).

Hayward, S. A Structured Development Methodology for Expert Systems. *Proceedings of KBS '86.* Online Publications, Pinner, pp.195-203, (1986).

Hayward, S. A., Wielinga, B. J. and Breuker, J. A. Structured analysis of knowledge.*International Journal of Man-Machine Studies*, **26**, pp. 487-498, (1987).

Hekmatpour, S. and Ince, D. C. Rapid Software Prototyping. *OSIT*. February 28, pp. 53, (1986).

Hoffman, R. R. The Problem of Extracting the Knowledge of Experts from the Perspective of Experimental Psychology. *The AI Magazine*, **8**(2), pp.53-67, (1987).

Huffaker, D. Prototyping Business Applications within the Traditional Life Cycle. *Journal of Information Systems Management*, Fall, pp. 71-74, (1986).

Jackson, P. *Introduction to Expert Systems.* International computer science series, Addison-Wesley Publishing Limited, pp. 246, (1986).

Johnson, P.E., Zualkernan, I. and Garber, S. Specification of expertise.*International Journal of Man-Machine Studies*, **26**, pp. 161-181, (1987).

Kemp, R. H. and Boorman, A. Using Wimps to beat the expert system blues. *Proceedings of The Second New Zealand Conference on Expert Systems (NZES'87)*, pp. 69-75, (1987).

Kemp, R. H. Frames as an alternative to Production Rules in Expert Systems. *Proceedings of the International Conference on Future Advances in Computing*, (1986).

Kunz, J., Fallat, R., McClung, D., Osborn, J., Votteri, B., Nii, H. P., Aikins, J., Fagan, L. and Feigenbaum, E. A physiological rule based system for interpreting pulmonary function test results. *Heuristic Programming Project Working Paper*, **HPP-78-19**, (1978).

LeFrance, M. The Knowledge Acquisition Grid: a method for training knowledge engineers. *International Journal of Man-Machine Studies*, **26**, pp. 245-255, (1987).

Littleford, A. Expert Systems & Computer Diagnostics. */usr/group/CommUNIXations* (**Aug-Sept**), pp. 10-11, (1985).

McCracken, R.J., and Cate R.B. Artificial intelligence, cognitive science, and measurement theory applied in soil classification. *Soil Science Society of America J.*, **50**, pp. 557-561, (1986).

McDermott, J. R1: an expert in the computer system domains. *AAAI-80*, pp. 269-271, (1980).

Minsky, M. A Framework for Representing Knowledge. *Reprinted in* " Context-Directed Pattern Recognition and Machine Intelligence Techniques for Information Processing". (Y. Pao and G.W. Ernst, eds), pp. 339-419, (1974).

Mylopoulos, J. and Levesque, H. J. An Overview of Knowledge Representation. In "On Conceptual Modelling." (M. L. Brodie, J. Mylopoulos and J. W. Schmidt, eds). Springer-Verlag, New York Inc., pp. 3-17, (1986).

Neall, V. E., Palmer, A. S. and Pollock, J.A. *Extramural Notes for Soil Science 1*. Massey University, (1987).

Neches, R., Swartout, W.R. and Moore, J. D. "Enhanced Maintenance and Explanation of Expert Systems through Explicit Models of their Development". *IEEE Transactions on Software Engineering*, **11**, pp. 1337-1350, (1985).

Newell, A., Shaw, J. C. and Simon, H. A. Empirical Explorations with the Logic Theory Machine: A Case Study in Heuristics. *In* "Computers and Thought" (E. A. Feigenbaum and J. Feldman, eds.). McGraw-Hill, New York, (1963).

Nii, H. P. Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures. *The AI Magazine*, **7**(2), pp. 38-53, (1986).

Nii, H. P. Blackboard Systems: Blackboard application Systems, Blackboard Systems from a Knowledge Engineering Perspective. *The AI Magazine*, **7**(3), pp. 38-53, (1986).

On Conceptual Modelling. Edited by M. L. Brodie, J. Mylopoulos and J. W. Schmidt. Springer-Verlag, New York Inc., pp. 510, (1986).

Partridge, D. *Artificial Intelligence - Applications in the future of Software Engineering*. Ellis Horwood Limited, England, pp. 240. (1986).

Prerau, D. S. Selection of an Appropriate Domain for an Expert System. *The AI Magazine*, **6(2)**, pp. 26-30, (1985).

Ramsey, C. L., Reggia, J, A., Nau, D. S. and Ferrentino, A. A comparative analysis of methods for expert systems. *International Journal of Man-Machine Studies*, **24**, pp. 475-499, (1986).

Rich, E. *Artificial Intelligence*. McGraw-Hill, Singapore, pp. 436, (1983).

Rich, E. Users are Individuals: individualizing user models. *In* "Intelligent Information Systems: progress and prospects." (R. Davies, ed). Ellis Horwood Ltd, pp. 184-201, (1986).

Ruled Based Expert Systems: The MYCIN experiments of the Stanford Heuristic Programming Project. Edited by B. G. Buchanan. and E. H. Shortliffe. Addison-Wesley, Reading, Massachusetts, pp. 745, (1984).

Scott, A. C., Davis, R., Clancey, W. J. and Shortliffe, E. H. Explanation Capabilities of Production-based Consultation Systems. *Heuristic Programming Project Memo 77-1*, **STAN-77-593**, pp. 30, (1977).

Scott, A. C., Clancey, W. J., Davis, R. and Shortliffe, E. H. Methods for Generating Explanations. *In* "Ruled Based Expert Systems: The MYCIN experiments of the Stanford Heuristic Programming Project." (B. G. Buchanan. and E. H. Shortliffe, eds.). Addison-Wesley, Reading, Massachusetts, pp. 338-362, (1984).

Shaw, M. L. G. Knowledge Engineering for Expert Systems. *Proceedings of Interact '84. North-Holland*, Amsterdam, pp. 328-332, (1984).

Shortliffe, E. H. *Computer-based Medical Consultations: MYCIN*, Elsevier, New York, (1976).

Shortliffe, E. H. Details of the Consultation System. *In* "Ruled Based Expert Systems: The MYCIN experiments of the Stanford Heuristic Programming Project." (B. G. Buchanan. and E. H. Shortliffe, eds.). Addison-Wesley, Reading, Massachusetts, pp. 78-132, (1984).

Shortliffe, E. H. and Buchanan, B. G. A Model of Inexact Reasoning in Medicine. *In* "Ruled Based Expert Systems: The MYCIN experiments of the Stanford Heuristic Programming Project." (B. G. Buchanan. and E. H. Shortliffe, eds.). Addison-Wesley, Reading, Massachusetts, pp. 233-262, (1984).

Stefik, M., Aikins, J., Balzer, R., Benoit, J., Birnbaum, L., Hayes-Roth, F and Sacerdoti, E. The Architecture of Expert Systems. *In* "Building Expert Systems." (F. Hayes-Roth, D. A. Waterman, and D. B. Lenat, eds.). Addison-Wesley, Reading, Massachusetts, pp. 3-29, (1983).

Sowa, J. F. *Conceptual Structures*. Addison-Wesley, Reading, Massachusetts, pp. 470, (1983).

Taylor N. H. and Pohlen, I. J. "Soil Survey Method." *N.Z. Soil Bureau Bulletin* No. **25** (2nd ed), pp. 242, (1970).

Todd, E. G. Knowledge Representation for a Soils Identification Expert System. *Proceedings of The Second New Zealand Conference on Expert Systems*, pp. 83-88, (1987).

Tsichritzis, D. C. and Lochovsky, F. H. *Data Models*. Prentice-Hall, Inc, New Jersey, pp. 381, (1982).

Weiss, S.M. and Kulikowski, C.A. *A Practical Guide to Designing Expert Systems*. Rowman and Allanheld, New Jersey, pp. 174, (1984).

Whitney, R.S. Knowledge-based Systems for Building Technology: Strategic Aspects. *Proceedings of The Second New Zealand Conference on Expert Systems*, pp. 5-11, (1987).

Wilde R. H. and Palmer A. S. *Field Trip - Otaki District*. Unpublished notes for Soil Bureau staff field trip. (1986)