A methodology for the interpretation of ground conditions from borehole information

Vaptismas, Nikitas

A METHODOLOGY FOR THE INTERPRETATION OF GROUND CONDITIONS FROM BOREHOLE INFORMATION

A thesis submitted to the

School of Engineering and Computer Science

University of Durham

for the degree of

Doctor of Philoshophy

by

Nikitas Vaptismas

January 1993

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Geotechnical design requires the interpretation of the information obtained from a site investigation. One aspect of the interpretation is the identification of the ground conditions across the site, based on observations at discrete points, such as boreholes. If a computer system is to assist in this process it must be able to compare soils observed at two or more locations, in order to identify whether the soils observed belong to the same horizon.

A methodology has been developed whereby the similarity of two soils can be calculated, based on engineering soil descriptions. The qualitative terms are converted into quantitative representations from which a Similarity Number can be derived. Individual Similarity Numbers can be calculated with respect to soil type, consistency, structure and colour. These are normalised to give values between 0 and 100 (with 100 indicating identical features) and combined using appropriate weighting factors to give an Overall Similarity Number which represents a comparison based on these features.

Using the quantitative representation of the soil descriptions, a preliminary assessment of the ground conditions can be made. The correlation of the borehole information is approached at two levels. At the site-wide level, an attempt is made to identify marker beds, that is soil layers which 'stand out' from the general ground conditions. A search for possible marker beds is first made at each borehole. The search is then extended to pairs of boreholes and further, the continuity of marker beds is established inside triangles which are formed having the boreholes as vertices. Where continuous layers are observed within triangles, the dip angle and dip orientation are calculated, to form the geometrical parameters on which preliminary

conclusions are based. Compatibility (or not) of these parameters between neighbouring triangles is the key factor for assessing continuity of the marker beds.

Finally, the detailed ground conditions are examined on a borehole-to-borehole level. At this level, a set of hypotheses about the ground conditions is constructed by looking at pairs of adjacent boreholes. Hence, a set of hypotheses is produced, even for areas for which the site-wide level approach is unable to establish trends.

ACKNOWLEDGEMENTS

CHAPTER 1

INTRODUCTION

1.1   General

A major part of geotechnical design is the interpretation of ground conditions from site investigation information, whether this is in the form of borehole logs, geophysical records or insitu test profiles. This requires interpolating or extrapolating from observations at discrete points such as boreholes and involves considerable engineering and geological judgement. Knowledge-Based Systems can be particularly useful in this since the interpretation process can be time consuming and tedious to do manually. Such systems use specific knowledge about an application area, usually in the form of facts, heuristics or rules of thumb, contained in their knowledge base. By modelling the reasoning scheme of human experts, they are considered appropriate for solving ill-structured problems, [43].

Geotechnical engineering is a field of civil engineering where one has to deal with natural materials (soil and rock) the distribution and properties of which often are highly variable and complex and difficult to evaluate. Site investigation is the first, and probably one of the most important, stages in engineering works, because it considers the identification of the soil profile and subsequently provides the means for a safe and economic design, which is the primary objective of the engineer.

One of the most important tasks in site investigation is the correlation of information recovered from boreholes. The engineer has to deal with detailed interpretation of the

ground at unobserved and unsampled areas and such a procedure is vulnerable to misinterpretations which can lead to imprecise conclusions about the soil profile. At its simplest, the operation involves linear matching of layers observed at different boreholes but when complex conditions are encountered, knowledge of geological processes is often necessary to arrive at a 'correct' solution. There is certainly not a standard way of handling the problem of correlation or interpretation. The correlation process is largely dependent on the experience of the engineer conducting it, the particular site under consideration and the type of the proposed works.

The scope of this work is to present a methodology for use within Knowledge-Based Systems, for the interpretation of ground conditions from borehole information. The proposed methodology involves processing engineering soil descriptions in order to calculate the similarity of two soils. The descriptive terms are converted to quantitative parameters, thus allowing a numerical form of comparison that results in a Similarity Number which is indicative of the similarity between two soils and is based on key features of the soils' description such as soil type, consistency, structure and colour. It must be noted that the system does not deal with rock but it can be extended to treat rock layers without involving large effort.

Based on the concept of the Similarity Number, the correlation process is approached at both site-wide and borehole-to-borehole levels. At the site-wide level an attempt is made to identify marker beds, which are layers that 'stand out' from the general ground conditions and can thus be more easily traced across the site, whereas at the borehole-to-borehole level the detailed ground conditions between pairs of boreholes are examined. Based on the hypotheses generated, the engineer can proceed with a preliminary assessment of the ground conditions.

A brief description of the contents of each of the following chapters is presented next, followed by a description of the hardware-software configuration used for the implementation of the methodology.

1.2   Overview of the Thesis

A review of Knowledge-Based System (KBS) applications in geotechnical engineering is briefly presented in Chapter 2, categorised according to the area of geotechnics that they deal with. KBSs which are involved with the process of site characterization, are presented in greater detail.

Chapter 3 is concerned with the current state of practice in site investigation. Initially, the structure, aims and procedures of site investigation are briefly described. Then, the soil components and the means to identify them is presented, followed by a description of the soil characteristics. The percentages of soil types participating in a soil description are defined in the form of ranges of percentages by the British Soil Classification System (B.S.C.S.), in accordance with the descriptive terms and these ranges, together with some comments on the functionality of the B.S.C.S., are presented next. Finally, the way of reporting borehole information together with some of the basic concepts of the correlation process and its possible problems are discussed.

The prototype system that has been developed for interpreting ground conditions is described in Chapter 4. Initially the components of an engineering soil description are presented. A parser module of the system is then described, for breaking down complex descriptions into their constituent parts. Then, through a Value Assignment and SImilarity Calculation (VASIC) module that is attached to the parser, numerical values are assigned to the descriptive terms which allow comparisons to be made between different features. Initially, comparisons between individual soils are discussed. However it is often necessary to make comparisons between layers

containing multiple soils, and a way of dealing with this problem is presented. In order to make the comparison between soils the concept of a Similarity Number is used. Similarity Numbers are calculated for each of the features and have a range of 0-100%; a value of 100% indicating identical features. The individual Similarity Numbers are combined using weighting factors (which represent the importance of each feature) to give an overall Similarity Number which represents a comparison based on all aspects of the description. Finally, in the discussion some of the improvements to be made are identified.

In Chapter 5, a methodology is presented for interpreting layering from borehole information. This takes place at both site-wide and borehole-to-borehole levels.
A search for marker beds is first made at each borehole. Continuity of marker beds is then investigated between pairs of adjacent boreholes. The investigation is then extended to groups of three boreholes (triangles) and finally over the whole site. Continuous marker beds identified in this way are used to construct an initial site-wide model of the ground conditions. At the same time, areas that appear to deviate from the overall trends are highlighted for further investigation. Finally, the detailed ground conditions are examined on a borehole-to-borehole basis. At this level a set of hypotheses about the ground conditions is constructed by looking at pairs of adjacent boreholes. The site-wide model is used to constrain the number of possible hypotheses at the detailed borehole-to-borehole level. The assessment of layering from borehole information has been implemented only for the detailed borehole-to-borehole analysis.

In Chapter 6, the advantages of the proposed methodology and possible future improvements are identified. In addition, the interaction of the system and the engineer is addressed together with the place of the system within a larger system which is under development in Durham University for the interpretation of site

investigation information. Finally, the conclusions reached from the development of the methodology are discussed.

1.3   Implementation

The parser and VASIC modules as well as the borehole-to-borehole correlation were implemented using the Phar Lap Dos-Extender Version of PDC Prolog 3.20, [51, 52, 53] on a 386 RM Nimbus VX/2 Personal Computer with 2Mb internal memory. Initially, PDC Prolog Version 3.20 was used on the same PC with 1 Mb internal memory. During this stage, Prolog did not seem to have enough addressable memory to be able to carry out its internal calculations and a 'Heap Overflow' error terminated the program's execution during run time, once the executed program had exceeded a certain size. It was understood that the memory problems originated from the fact that this Prolog version was not able to utilise any memory above 640 Kb allowed by the MS-DOS operating system. By using the Phar Lap Dos-Extender version and by expanding the PC's internal memory to 2 Mb, PDC Prolog was enabled to address enough memory above 640 Kb to run the programs and the memory problems that were slowing down the development of the system were eliminated.

CHAPTER 2

KNOWLEDGE-BASED SYSTEMS APPLICATIONS IN GEOTECHNICAL ENGINEERING

2.1   Introduction

Civil engineering and in particular geotechnical engineering involves the use of engineering judgement and dealing rationally with considerable uncertainty. Unlike most engineering fields, geotechnical engineering requires that the engineer work with natural materials (soil and rock) the distribution and properties of which are often highly variable and difficult to evaluate.

The following quote by Terzaghi and Peck, [66], expresses why geotechnical engineering may be the most appropriate field within civil engineering for the development of Knowledge-Based Systems.

" In foundation and earthwork engineering, more than in any other field in civil engineering, success depends on practical experience. The design of ordinary soil-supported or soil-supporting structures is necessarily based on simple empirical rules but these rules can only be used by the engineer who has a background of experience."

Knowledge-Based Systems (KBS) are computer programs that use knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution, [22]. These systems use specific knowledge about an application area, usually in the form of facts, heuristics or rules of thumb,

6

contained in their knowledge base, and by modelling the reasoning scheme of human experts, are considered appropriate for solving ill-structured problems.

Knowledge-Based Systems technology originates from a branch of computer science that is referred to as Artificial Intelligence (AI). AI is concerned with a broad range of topics that are related to simulating human intelligence in computers. Some of the better known areas of AI are natural language understanding and robotics. The application of AI in geotechnical engineering has evolved during the last few years and as AI technology develops and familiarity with such systems increases, it is likely that they will become important tools in that engineering field. Toll has outlined some of the reasons that indicate the necessity of the new technology's application in geotechnical engineering, [68].

Some of the Knowledge-Based Systems applications in geotechnical engineering are briefly described in the next section. Then, a more detailed description of such systems, specifically concerned with the important task of site characterization is presented.

2.2   Description of Geotechnical KBSs

Several Knowledge-Based Systems have been developed in different areas of geotechnical engineering. These systems are briefly described in the following, categorised according to their objective.

In the area of general foundation design, Meyer, [38], describes a Knowledge-Based Expert System (KBES) that addresses the preliminary foundation design of multi-storey buildings using the expert system shell EDESYN. The system uses preliminary soil data (SPT N-value for cohesionless soils and undrained shear strength and Attenberg limits for fine grained soils) and the building's potential

configuration in order to characterize the underlying soil and to produce a set of feasible solutions to the preliminary foundation design problem.

Another system using the expert system shell EDESYN is FOOTER, [1]. This is a KBS that performs design synthesis for building foundations using as input soil conditions, water table location, depth to bedrock and the imposed loading conditions. The output of the system comprises all feasible foundation alternatives which are consequently evaluated by the user.

Rowlinson, [58], briefly describes Geotech, a KBS under development to assist in foundation design in Hong Kong. The factors which are to be considered in the development of the system and which determine its structure are technical, legal and commercial, as well as local practice.

Finally, Rashad *et al*, [54], describe FOUNDation CONsultant (FOUNDCON), a modular Knowledge-Based Computer-Aided Design (CAD) System under development to assist in foundation design.

Some of the problem-solving modules of the system as these are envisaged, are :

o   Interpretation Module that provides a preliminary validity check of the input data and performs soil data interpretation.

o   Preliminary Design Module that selects the most appropriate foundation system.

o   Modelling and Analysis Module that models the structural configuration proposed above, and predicts its response to external conditions.

o   Detailed Design Module that performs the final design, ensuring that all constraints are satisfied.

In the area of shallow foundation design, Yehia and El-Hajj present FOOT, [77], a KBS to assist in the selection and design of spread footings. The program consists of four main modules which are:

○ MAIN is the program module concerned with the problem-specific data such as number, distribution and loading of the columns.

○ DECIDE is the module corresponding to the inference engine of the program. It must be noted that the columns' distribution must be rectangular so if that is not the case, fictitious columns are incorporated into the site plan.

○ GRAPH is the module that provides general plans of columns and footings, and also plots the reinforcement details for footings, only for the best choice because of memory requirements.

○ DESIGN performs the structural design after searching into its databank for similar cases. After every run of the program its database becomes larger and so in future problems the solutions are bound to improve.

GEOTECH, [50], is a KBS that was developed to aid in shallow foundation design by calculating bearing capacity and settlement and ultimately producing the foundation design. It considers several properties of the ground like soil type and water table and structural information like load and column dimensions. The system incorporates the uncertainty involved in foundation design by using fuzzy logic.

For the task of pile selection, Santamarina and Chameau, [59], have developed PILE, a prototype KBS to assist in the selection of the appropriate type of pile foundation. The output of the system is a list of the most promising alternatives based on technical constraints. Then, it is up to the user to consider additional factors (e.g. economical), in order to reach a final decision. The perfomance of PILE has been successfully evaluated in a wide range of cases and its production rules have been proven efficient and sufficient for small tasks.

Wong *et al*, [75], have developed SUPILE, a KBS to assist in the evaluation of suitability of different types of piles and in the estimation of the required pile size and length. The selection of a pile type is performed by finding how many problems could exist if a specific type was used. These problems are quantified in the form of a problem score and finally a suitability score is produced for each pile type. It has a value between 0 and 99, where the higher the suitability score, the more suitable is the pile.

Finally, Elton and Brown, [20], describe PILEX, a KBS to aid in the selection of reliable pile types by considering spread footings and timber, concrete and steel piles. The input to the system includes loading parameters and soil and groundwater conditions.

The problem of bridge foundations is focused on by Stuckrath and Grivas, [65]. A prototype KBS is presented and its objective is to assist in the selection of bridge foundations at the planning and preliminary design stages. Based on user input concerning structural (load applied directly to the foundation element, admissible settlement) and geotechnical (ground type defined either by laboratory test results, if available, or based on visual examination of the site, stratigraphy, ground water) specifications, the system presents preliminary designed options such as shallow foundations (isolated or strip footings and rafts), improved ground (through compaction or grouting) and deep foundations (piles or combinations of piles and footings or rafts).

Another approach to the bridge foundation problem is BABE, [78], a KBS developed to help the engineer in the selection of the most appropriate type of foundation for a specific structure and a set of site conditions. The selection of the foundation type (footings, piles or caissons) to be used is based on the loads, the superstructure conditions, geological and hydrogeological characteristics, the potential problems in

construction and the cost of the foundation. The type of foundation selected as well as the loads and design criteria are considered by the system in order to achieve the optimum design of abutments and piers.

In the area of earth retaining structures design, Oliphant and Blockley, [49], developed a KBS that advises the user on decisions concerning the selection of earth retaining structures. The knowledge base of the system consists of 11 retaining wall case studies and provides a narrative of the history of each case study in terms of why it was selected or considered as an alternative, leaving the user to compare these with a proposed retaining wall. The system incorporates the uncertainty involved using the support logic program "shell" called FRISP. The KBS is divided into three main parts; the construction process, the design process and the environmental impact.

Arockiasamy *et al*, [3], describe a KBS for retaining wall selection and design. The system consists of two modules, the selection and the design modules. In the selection module a wall is selected based on the given set of criteria. The selection is made from a list of ten walls types including concrete gravity, cantilever, counterfort, gabions, reinforced-earth, crib, slurry, sheet-pile, tieback and soil nailed walls. The user is asked to describe the site given a list of site locations. Then he/she is queried about site geometry, wall height, project time, material and labour availability, equipment access, construction familiarity and aesthetical considerations. Finally the most appropriate wall types are selected.

Hutchinson *et al*, [30], present a rule-based KBS that concentrates on the selection of the applicable earth retaining structures. The system first evaluates if a retaining wall is required or an embankment or cut would be satisfactory guided by the user's input about the type of application, topographical and soil conditions. If a wall is found necessary, the system evaluates which of the nine wall types that it knows about (brick wall, blockwork wall, crib wall, gabions, gravity wall, railway sleeper wall,

reinforced earth, reinforced concrete wall, sheet piling) is applicable in that specific case. If more than one wall type is applicable, the system bases its recommendations on the first satisfactory solution encountered.

Some systems have also been developed for the assessment of foundation and retaining wall failures. WADI, [9], is a prototype KBS developed for the preliminary diagnosis of retaining wall failures. WADI is applicable to either cantilever reinforced concrete walls or gravity concrete or rubble walls, having a maximum height of 8 metres. At the beginning of execution, input information concerning the wall under examination, the backfill soil, the bearing soil, the angle of the backfill and the failure symptoms of the wall, is given. WADI classifies the bearing and backfill soils in order to determine their engineering design characteristics. Then, it performs some preliminary investigations of the failure data in order to identify the general areas of retaining wall problems such as a footing problem, drainage problem, weak bearing soil, construction problem, that may be relevant to this failure. The system proceeds to a stability analysis of the retaining wall using conventional design calculations and checking through computations, a factor of safety against each type of failure (overturning, sliding or settlement). The final conclusions as to what might have caused the type of wall failure observed and recommendations as to what actions might be taken are given by the system combining the preliminary problems generated and the different unacceptable factors of safety.

Adams *et al*, [1], developed a KBS for allowing categorisation and organisation of knowledge relating to failure and rehabilitation of earth retaining walls. The system was developed in a modular manner, having modules that treat site identification, failure diagnosis, design synthesis and cost estimation. Upon completion of the failure diagnosis module, a table of wall failure modes with associated certainties is generated. Associated with each failure mode is heuristic knowledge regarding design components that may be used for rehabilitation. Each rehabilitation strategy is

related to a set of soil and construction constraints and a preliminary design is produced for each one. Then by combining design components a complete design is achieved.

Hadipriono *et al*, [26, 27], developed a KBS for the assessment of foundation failures. Soil settlement, expansive soil, soil erosion, bearing capacity, slope instability and foundation corrosion are identified as the possible causes for foundation failure. The user is queried by the system about the evidence that show a possible foundation failure (such as crack pattern, joint openings, wall deflection etc.) and about known soil information. The system employs the use of fuzzy sets for the calculation of uncertainty.

In the area of slope stability analysis, Faure *et al*, [21], are developing XPENT which is a KBS to assist the engineer in slope stability problems. The data concerning the analysis of the problem are stored in a database through an interface able to assist in the visualization of the situation. The model can simulate operations such as embankment building, drainage and the consequent evaluation of change in stability. The system is a prototype at the stage of final validation.

Wislocki and Bentley, [74], describe the development of a prototype KBS for the determination of planning applications with respect to landslide hazard existing in South Wales. The system attempts to assess the landslide hazard that may affect proposed development sites and it produces output in the form of planning response options (which have been formulated to allow almost direct integration into the planning process operated by Local Planning Authorities in UK).

Gillette, [24], describes a Computerised Adviser on Soil Strength (CASS) that is a KBS to assist in the selection of shear strength parameters for use in stability analysis. After the preliminary data entered by the user, the system begins by assessing the

main outcomes, which are the shear strength parameters ϕ and c, a recommendation about the strength representation in the analysis, advice on soil behaviour and warnings about possible problems. Checks on the consistency and validity of the input information are also performed by the system. CASS is being used by the geotechnical company that developed it, on projects where soil information is mainly historic, sparse or not yet available. Finally, comparisons between CASS's predictions and values from appropriate testing have shown generally good agreement.

In the area of ground improvement, Chameau and Santamarina present IMPROVE, [10], which is a prototype Knowledge-Based decision-support System designed to assist in the selection of soil improvement techniques. The system consists of four parts; the preprocessor that helps the user decide if there is need for soil improvement, the classification system that selects the best alternative soil improvement technique and can continue the search for less satisfactory solutions after the user's request, the case-based system that selects the case histories that best resemble the project from 50 case histories that are included and the postprocessor that is a ruled-based system which provides final information and suggestions.

Motamed *et al*, [41], describe an Expert System for Preliminary Ground Improvement Selection (ESPGIS) that advises users in selecting ground improvement methods or to evaluate the suitability of a user's preselected method given the characteristics of the site. EPSGIS allows the user to define the problem by specifying, with varying degrees of certainty, the nature of the ground improvement need, subsurface conditions and other relevant parameters. It questions the user on stratigraphy and simple index properties of the underlying soil and assigns typical values for design parameters for the soils based on the soils' description and its index properties. The questions not requiring numerical values are accompanied by choices and the user has to define his/her certainty on a scale from 0 to 100. When

quantitative parameters are required, there are no choices and confidence factors, and the system assigns a certainty factor of 100. According to the authors, it seems to be in good agreement with some case studies that were used for validation of the system.

In the area of geosynthetics, Maher and Williams, [36], describe a hybrid expert system that selects geosynthetics materials and performs detailed designs for different geotechnical applications. The knowledge incorporated in the system was obtained through a literature survey. It contains information for material selection for five different geosynthetic uses. These are stabilization to reduce erosion, separation of soil layers, reinforcement to improve soil strength, drainage material to remove water and filtration to reduce cross plain flow of soil particles.

Edge Drain by Expert System, [18], is a computer software package that was developed to assist in the design and specification of the geotextile component of the edge drain. It accepts as input raw site data in the form of rainfall and native soil characteristics, design requirements consisting of subbase material characteristics, pavement system and edge drain cross section information and construction conditions. The system considers commercially available geotextiles that are non-woven and perform the dual functions of drainage and separation. The output consists of the required hydraulic and mechanical properties which are determined using typical algorithmic solutions and a list of the ten thinnest (lightest) candidate products arranged in ascending order. One limitation of the system is the nature of the underlying soils. It cannot handle soil conditions which include gap-graded or internally unstable silts.

More systems have been developed in other areas of geotechnical engineering. Davey-Wilson and May , [14, 16], describe GroundWater eXpert (GWX), which is a prolog-based system to advise on appropriate methods for groundwater control in excavations.

Davey-Wilson, [15], has also developed a KBS for soil shear strength analysis that uses soil descriptions as input in order to infer shear strength in degrees, to a maximum accuracy of $\pm 1^o$.

Juang and Lee, [34], developed the Rock Mass Classification (RMC) system which is based mainly on Bieniawski's geomechanics classification scheme.

SOLES, [60], is a KBS to assist in evaluating the liquefaction potential of soil subjected to earthquake excitations. Validation of the system showed an approximate 90% success rate in predicting soil liquefaction potential during earthquakes.

Mi and Jieliang, [39], describe a KBS to assist in the prediction of the value of surface settlement and the degree of damage to buildings, caused by shield-driven tunnelling and to propose preventative and strengthening measures.

Finally, a prototype KBS that has been developed for providing assistance to the planning of safety precautions for a trench according to the soil conditions encountered, [61]. The system is based on two new soil classification systems developed by the National Bureau of Standards (USA) in order to increase the safety of this type of excavation.

2.3 KBSs involving Site Characterization

In this section, some of the systems that are concerned with the task of site characterization, are presented.

2.3.1 Dipmeter Advisor

The Dipmeter Advisor, [64], is an interactive system that uses sequences of dip estimates from a dipmeter log together with knowledge about local geology to infer subsurface geologic structure. The dipmeter tool measures the conductivity of rock at a number of depths and directions within a borehole.

The system is made up of :

- a knowledge base consisting of 90 production rules grouped into several distinct sets according to their function (e.g. structural vs stratigraphic rules)

- a forward chaining (or data driven) inference engine that resolves conflicts by rule order

- a set of feature detection algorithms for a preliminary interpretation of log data

- a menu-driven graphical user interface.

The conclusions obtained from the rules are stored on a blackboard that is divided into 15 layers of abstraction (e.g. patterns, lithology). Dipmeter interpretation is subdivided into 11 successive phases and after each is completed an interactive dialogue is set up for the user to examine and, if needed, modify results reached by the system. However, the system is familiar with a relatively small number of different lithologies due to the limited knowledge included and has a very local view of consistency in the vertical sequences of dips, attributable to the fact that the system's reasoning is based on empirical rules, not having any model of the ground conditions that lead to the rules. According to the authors, improvements on the latter aspect cannot take place without redesigning the system.

17

The Dipmeter Advisor is written in INTERLISP and operates on the Xerox 1100 Scientific Information Processor.

2.3.2 SITECHAR

Norkin, [47], and Rehak *et al*, [55], present a KBS component of a geotechnical site characterization workbench, called SITECHAR. The purpose of this system is to develop inferences on the depositional patterns of the subsurface materials and their physical properties by interpreting field and laboratory data and taking into account existing experience of geology and geomorphology at a specific site or similar ones.

The system uses the 'blackboard model' architecture, which represents a complex problem solving technique. The blackboard is a dynamic global database through which a set of diverse and independent knowledge-based processors, called knowledge modules, communicate with each other. The initial SITECHAR system incorporates the following ruled-based knowledge modules: knowledge of geometry and trends, matching soils by description , proximity (such as "near", "above", etc), geomorphology (such as erosional surfaces, channel cutting, etc), geology (such as faults, folds, etc) and searching for marker beds. This knowledge can be divided into macro/micro-level knowledge and strategic/tactical knowledge. Macro- and micro-level inferences represent different hierarchical levels of problem abstraction. Strategic knowledge works on a higher level, by defining an immediate goal or strategy for the problem solution. Tactical knowledge works for and under the guidance of the strategic knowledge. These knowledge classes are not necessarily distinct. Each knowledge module is potentially able to work at all levels. The inference engine, which supports both forward-chaining and backward-chaining problem solving techniques, controls the manipulation of, and interaction between, the blackboard and the knowledge modules.

SITECHAR is a prototype system demonstrating the advantages of KBS for geotechnical applications, whereas commercial use of that system would involve gathering of significant amounts of additional knowledge.

2.3.3 CONE

CONE, [44, 45], is a KBS that interprets raw data from the cone penetrometer (CPT) in order to perform an input and validity scan on the raw data, classification of the soil types (including the profiling of layers) and inference of design parameters with respect to the shear strength of sands and clays. The soils are classified using two electric-CPT based classification systems, the Dutch classification system and the Douglas and Olsen classification system. Another system was also used which is a fuzzy set representation, based on the raw database used to develop the Douglas and Olsen system. The shear strength of sands and clays are estimated using empirically and rationally based methods.

Fuzzy sets are used to treat uncertainty with respect to linguistic data (i.e. soil classification), numeric data (i.e. determination of shear strength) and quality information (i.e. appropriateness of a soil classification system, the accuracy of the system for certain soil types etc). This is in the form of a Belief that expresses the strength of belief to the associated domain knowledge and a Weight that measures the relative importance of that piece of knowledge compared to other pieces of knowledge at the same focus level. Both Belief and Weight are expressed as linguistic variables (high, medium, low). The fuzzy sets are represented over a five-valued universe and are implemented using OPS5 rules and LISP functions.

Although the system does not consider soil descriptions containing the terms 'slightly', 'very' and 'gravelly', its performance in two case-studies was found satisfactory by the authors, especially with site-specific adaptation of knowledge. According to the authors, improvements are envisaged in the form of more extensive

expert interaction and the inclusion of multiple case histories. Finally, a typical run of CONE may take up to 1.5 hours on a lightly loaded DEC-20, depending on the length of the CPT log.

2.3.4 SOILCON

SOILCON, [61], is a prototype KBS that was developed for assisting the engineer in deciding the level of geotechnical investigation necessary. This is based on the requirements of a proposed structure and the level of information known about the site in order to reduce the risk involved with the subsurface to an acceptable level. The system was implemented using the M.1 rule-based expert system shell which provides a backward chaining control strategy. The knowledge base contains 24 investigation techniques ranging from preliminary (e.g. reviewing topographical maps) to more sophisticated (e.g. pressuremeter), and these are used to make the ultimate recommendation.

SOILCON starts by querying the user for preliminary project and site data. These queries are then followed by higher level questions partly based on the answers already given. Based on the amount of data available the system makes recommendations on the level of investigation, increasing the complexity of the recommended investigation when there is a large amount of site data available.

A limitation of the system is that it does not handle geometric descriptions of the problem and site quantitively. The size of a project is described as being large, medium or small, while the foundation geometry is given as either shallow or deep. If the system is to be commercially successful, its scope must be limited in order to provide solutions in greater detail. The system is classified as being a prototype under development.

20

2.3.5 Soil Investigation

Alim and Munro, [2], describe a very simple prototype expert system on soil investigation. It offers guidance on soil identification based on visual and physical observation of soil characteristics and provides judgement about the most likely foundation type under given soil and loading conditions. Based on these two conclusions it gives the possible foundations problems and finally it combines all this information to suggest the most suitable sampling and drilling techniques for the particular investigation scheme. This expert system was written in micro-PROLOG and uses the PROLOG expert system shell APES. The complete system exists as six separate files, which are fully compatible with each other and can be used both independently or by loading them all into memory at once.

The system handles uncertainty and imprecise knowledge using fuzzy logic to produce degrees of belief which take numerical values from zero to unity. The paper presented by Alim and Munro was discussed by Davey-Wilson *et al*, [17], and some interested points arose like the limitations of the software used (micro-PROLOG and APES).

2.3.6 SITECLAS

SITECLAS, [76], is a KBS to assist in the classification of a site according to the Australian Standard AS2870.1. The system was developed by using the expert system shell SUCAM. SUCAM is written in TURBO PROLOG and runs on an IBM PC or compatible microcomputers under MS-DOS. Its main components are: a knowledge base, a fact base, an inference engine, a user interface, an explanation facility and modules for different functions like selecting the appropriate Rule File (an ASCII file storing the domain knowledge), reading the Rule File, reading and writing the Result File, specifying Consultation Control, goals and facts and showing results.

The knowledge base stores the knowledge about a subject domain in the form of IF-THEN or IF-THEN-ELSE production rules, procedures, tables and comments. The fact base stores the consultation specifications, the input goals, the input facts and the conclusions of the consultation, providing the advantage of being able to modify the input facts without starting a new consultation. The inference engine is based on the backward chaining reasoning. The user interface is screen-driven which makes the system user-friendly. A data sheet entry form is used for input and output with functions to invoke the explanation facility. The explanation facility allows three types of explanation: Rule Explanation (to explain why certain information is required), Rule file Explanation (to explain how a certain conclusion was reached), Help File Explanation (for further explanations, comments, remarks, and notes). SUCAM is able to find all the goals in the knowledge base as well as all the input information required for the specified goals. It is a deterministic system which does not deal with imprecise, uncertain or conflicting knowledge.

SITECLAS contains about 100 rules in order to classify a site by using different procedures (e.g. identification of the soil profile, classification, computation of the predicted surface movement in accordance with engineering principles and visual assessment of the site) according to the Australian Standards. The input for SITECLAS involves information about the natural soil or fill found at the site. Validation of the system was made for five different sites, and showed that experience or special knowledge is needed for the interpretation of parts of the Australian Standards which should be consequently included in the system, by providing more explanation of the corresponding statements in the Standards.

2.3.7 LOGS

LOGS, [1], is a KBS, based on the ideas introduced in SITECHAR, [47, 55], that treats information from several boring logs and provides the user with two dimensional subsurface profiles. It is a rule based forward-chaining system written in

the OPS5 and Common LISP languages and was implemented in the Knowledgecraft environment which provides windows and graphics interface for graphically displaying subsurface cross sections. Knowledge about geology and geomorphology is embodied in the system and it is handled through heuristics that apply to a specific region (Kane County Illinois). These rules develop hypotheses on site geology and geomorphology that must be verified using site data. The system tries to identify marker beds, lenses (wedge-shaped deposits) and lentils (strata with boundaries within the confines of the site) consisting of either till, lacustrine or outwash which are the major geologic terrains observed in that area. A soil may be identified as a continuous layer even if it is not present in all borings, based on the knowledge of the area's geology.

The current version of LOGS comprises approximately 350 rules and future improvements are identified to be three dimensional interpretation and calibration against the judgement of experts. The system is mainly site-specific and therefore its application to areas with different geologic features would involve major modifications.

2.3.8   Site Investigation

Smith and Oliphant, [62, 63], describe a prototype KBS for civil engineering site investigation. One of the authors aims with the development of this sytem was to show the potential applications of this technology in the site investigation industry. The primary requirement of the system was to act as an adviser during any stage of the site investigation process and especially during the planning stages (e.g. desk study, site reconnaissance etc).

The system has been implemented to run on an I.B.M. P.C. compatible supporting MS-DOS. It was developed using the Leonardo Development System, Level 3, produced by Creative Logic. This environment contains a text editor used to create

rules for the knowledge base and an inference mechanism which mainly uses the default technique of backward-chaining; this suits the hierarchical nature of site investigation, although forward-chaining can be enforced where necessary. Leonardo uses rulesets, objects and object frames to represent the knowledge for an application. Every application starts with the execution of a rule in a ruleset which is called the main ruleset. The goal is the object (variable) whose value is obtained through the knowledge base. The values of the objects are controlled by the contents of the object frames which consist of a number of parameters (slots) set at specific values during the development of the knowledge-base. The prototype features a systematic data input facility that helps minimize oversights or omissions of relevant data. The data obtained from the planning stages of different site investigations are fairly similar, so it was possible to create multiple choice menus as a means of getting the data from the user. The information obtained is used by the system to provide suggestions to the user for the following stage of a site investigation, the subsoil exploration (possible locations of boreholes, trial pits, etc. and suitable types of soil testing). The variability of data returned from the subsoil exploration stage of site investigations was handled by writing external executable programs. The information obtained at this stage is used for the creation of a 2-D visual representaion of the soil layers. The strength characteristics of the various soil strata (identified as keywords suggesting strength characteristics in the soil descriptions entered during the subsoil exploration) are used by the system to make recommendations for suitable foundation types for the ground conditions present.

The prototype system is user friendly, can be used as a learning tool, provides the facility for future expansion and has a cost saving capability. However, according to the authors, substantial work is still required for the development of a working tool acceptable to the site investigation industry.

2.3.9 Probabilistic Site Characterization

A KBS, that is described by Halim *et al*, [28], was developed to assist engineers in performing site exploration decisions and evaluation of geotechnical design concerning shallow foundation or slope stability, using probabilistic analysis within an interactive user-friendly environment. The prototype system was developed using the knowledge engineering environment KEE on an Apollo DN3500 workstation. The system has been implemented to perform three major tasks:

  o Inference of prior estimates of soil and anomaly characteristics using production rules.

  o Selection of the most appropriate exploration program using probabilistic analysis where anomalies and soil properties are represented by a set of attributes as probability of anomaly presence, and means and standard deviation of anomaly size and locations.

  o Reliability evaluation of the proposed geotechnical system.

The inference mechanism of the system is forward-chaining and the knowledge incorporated is represented through a combination of frames and rules, that are both features of the environment used. The system's functionality is similar to that of SOILCON, [61], with additional capabilities to handle uncertainty about the ground conditions quantitatively. According to the authors, future development of the system involves including the capability of updating the estimated soil properties, based on the site exploration results.

2.3.10 Design Parameters

Carpaneto and Cremonini, [8], describe a KBS framework for the automation of site characterization process for geotechnical design. The system is based on an existing KBS, [57], employed for geotechnical characterization of the site soil profile. The system consists of several databases, where information is stored about the site

under consideration, a knowledge base containing the domain knowledge and an inference engine capable of interpreting the available data.

The task of characterizing the site is divided into four phases :

○ An Input Phase where information from the databases is used to make some preliminary inferences about the soil profile and its properties.

○ A Comparison Phase where rules are used to validate the data obtained in the previous phase and to improve on the possible soil profile.

○ A Reduction Phase where the construction of a best solution is carried out.

○ An Output Phase where the best solutions detected for the borehole stratigraphy and the corresponding design parameters are processed for appropriate display of the results.

Some possible future improvements of the system are also discussed, mainly for making soil profile inferences in sites where limited data are available but where there is a general knowledge of the area, and the inclusion of additional software such as management system for database interaction and extended graphics packages.

2.3.11 SCHICORRE

Baumbach and Plumer, [5], briefly describe SCHICORRE, a KBS for the correlation of stratigraphic sequences. This is a highly interactive, graphic intensive system, running under MS-DOS that has been implemented in LPA Prolog Professional. It facilitates the identification of geological seams, especially coal seams, based on borehole information.

The correlation of the stratigraphic sequences is based on matching geological and spatial attributes of the seams, which are defined in the database of the system, such as typical fossils in the seam's roof, ash content of the seam, sulphur content of the seam, seam thicknesses and seam distances. Further, the correlation is also based on the concept that seams cannot cross each other. The correlation takes place between

pairs of neighbouring boreholes (local correlation) and is then extended to four boreholes forming a polygon (global correlation).

The user may reject correlations proposed by the system and can enforce own correlations. Correlations generated by the system are explained by the main geological and geometrical attributes of the involved segments. According to the authors, in the context of the European coal mine industry, it is the first running prototype producing plausible correlations and offering an interactive, user-friendly environment.

2.4 Discussion

Knowledge-Based Systems can be very useful in all areas of civil engineering and particularly in geotechnical engineering by addressing aspects involving knowledge and experience, which can not be handled by traditional programming methods.

Earlier in this chapter, several systems which either have been developed or are under development were described, all of them addressing a range of problems that a geotechnical engineer is likely to encounter. Most of the systems described were developed in order to demonstrate the potential of using KBS in geotechnical engineering, [43]. As a result of this, most of the systems require further substantial development to become commercially acceptable and consequently to convince the geotechnical engineer about their usefulness as advisory systems and not as tools targetted to replace the engineer.

Most of the problems that arose from the development process of these systems are mainly involved with the knowledge acquisition and knowledge representation procedures. It is broadly accepted that the knowledge incorporated is the most important part of a KBS, [22]. Obtaining the required knowledge is the most difficult

task in the development of a system. This is a lengthy process because in most cases personal experience and expertise can not be derived from published material. In most systems the expertise was acquired through questionnaires, [43], structured or unstructured interviews with domain experts, [62, 63] and by modelling the solving procedure that an expert follows when solving an example problem that serves as a case study, [47]. The selection of the knowledge representation scheme is also a critical decision, as it was mentioned above. It requires a good understanding of the nature of the domain knowledge and can consequently affect the selection of the implementation tool. It can be observed that the three methodologies most commonly used in the KBS involving site characterization for representing knowledge are, rule-based representation, [1, 8, 44, 45, 47, 55, 61, 62, 63, 64, 76], which seems to be the most favourite one, logic-based representation, [2, 5], frame-based representation and a hybrid representation using both rules and frames, [28]. In addition, some systems incorporate a database of case histories allowing access to prior experience.

It is, also, evident that either incomplete or very site-specific knowledge is a major problem of the existing systems. In the Dipmeter Advisor, [64], the limited knowledge about lithology and the lack of understanding of the ground conditions by the system will eventually lead to a redesign of the system, as the authors state, whereas in SITECHAR, [47, 55], commercial use of the system involves the inclusion of significant amounts of additional knowledge. In LOGS, [1], which is based on SITECHAR, the site-specific knowledge used does not provide for a flexible system that can be widely used. On the contrary, the scope of SOILCON, [61], should be reduced in order to provide detailed solutions and its weakness in handling quantitative geometric data provides an additional problem. In SITECLAS, [76], the validation process identified the need for inclusion of experience or special knowledge in the system, together with a more explicit explanation of the Australian Standards, on which the system bases its conclusions. Further, Smith and Oliphant,

28

[62, 63], recognise the need for substantial additional work in order to produce a system that can be accepted by the site investigation industry.

Another critical decision concerning the creation of a KBS is the hardware-software configuration that must be selected when designing the system. In Soil Investigation, [2], the authors identified the need for a change in the software used, which can eventually lead to redesigning the system. It is also worth noting that such problems can also arise if large amounts of additional knowledge or a different knowledge representation scheme is required in order to transform a system from a demonstrational prototype to a commercially acceptable one.

The existing KBSs have already demonstrated that geotechnical engineering is a field of engineering that has a lot to earn by the use of such systems. Many efforts towards that direction have already been made and as a result geotechnical engineers, especially in terms of research, are increasingly becoming familiar with this new technology. Their introduction in academia has provided a powerful tool for educational purposes, and as a result such systems are already used, [15]. However, the real benefits from the introduction of this technology are bound to appear when these systems become commercially acceptable and this should be the direction to follow in the future.

CHAPTER 3

CURRENT STATE OF PRACTICE IN SITE INVESTIGATION

3.1   General

Site investigation is the process by which geological, geotechnical and other relevant information, which might affect the construction or performance of a civil engineering or building project, is acquired, [12]. In this work the term "site investigation" has been used to cover all methods and enquiries that can be used to gather information on a particular site, while all activities relevant to the exploration of the subsurface conditions are termed "ground investigation".

Site investigation is the first, and probably one of the most important, stages in engineering works, because it considers the identification of the soil profile which can be highly variable and complex and subsequently provides the ground for a safe and economic design.

Details of site investigation structure, aims and procedures can be found in [73].

3.2   Objectives of Site Investigation

The primary aims of a site investigation, according to British Standard BS5930, [6], should be the collection of data for identification of the following :

o *Site Suitability* : To advise on the suitability of the site and its neighbouring areas for the construction of the proposed works from a geological and geotechnical point of view.

o *Design* : To enable an adequate and economic design for both temporary and permanent works.

o *Construction* : To predict and evaluate possible problems that may arise during construction due to ground or other local conditions, to select sites for the disposal of waste or surplus material and to select the best method of construction for the site under consideration.

o *Effect of Changes* : To consider possible changes in the environmental conditions of the site and the surrounding areas which might occur either naturally or as a result of the construction works.

o *Choice of Site* : To assess the relative suitability of different sites or parts of the same site.

Occasionally, site investigation may be carried out for assessment of the safety of existing works or for investigation of cases where failure has occured.

3.3   Phases of Site Investigation

A site investigation can be broadly divided into several phases. Although these phases are considered to be similar, regardless of the size of the site and the kind of the proposed construction, the detail to which they are carried out mainly depends on these factors. Some of these stages can be taken out of the sequence that is given below or they may overlap.

A site investigation will consist of a Preliminary site investigation including desk study, site reconnaissance and preliminary ground investigation, a Main site investigation including the main ground investigation and the identification of ground water conditions, and finally Investigation of special cases, if required. A brief description of these stages is presented later in this chapter. However, the above

31

description of a site investigation seems incomplete because there is not an initial formal phase covering the planning and design of the investigation, which is the key for an efficiently carried out investigation. This point is strongly emphasized by Head, [29], and Littlejohn, [31]. Further, according to Littlejohn, [31], current site investigation practice is inadequate because there is a lack of client awareness of the importance of site investigation, there is not enough communication between all the parties involved in the planning, design and construction stages, there is a lack of proper site supervision and finally the site investigation is not adequately related to the design and construction requirements of the proposed project.

Another major factor concerning a successful site investigation is the attitude of the engineer involved. He or she must review the model of the site every time that new information comes in and also adjust, if needed, the remaining stages of the investigation, targeting the identification of ground conditions especially in areas where problems or anomalies seem to govern.

3.3.1 Desk study

The desk study is considered to be the stage of gathering preliminary information about the site. It involves collection of available documentary information from national and local authorities in the form of topographical and geological maps and records, and aerial photographs. Private sources may also be assessed. The desk study is an essential part of site investigation because it is the stage where the engineer can have a first idea of the site quality and it provides him/her with potential information that will help him or her to plan the site investigation accordingly.

3.3.2 Site Reconnaissance

Site reconnaissance or walk-over survey involves an inspection of the site and the neighbouring areas by foot. It is the stage of an investigation where information

previously obtained will be confirmed and amplified and so a thorough study and understanding of the maps and records (desk study) is obligatory prior to site reconnaissance. The reconnaissance of the site will provide the engineer with information concerning site accessibility, presence of materials together with their distribution and general properties, topography and subsurface drainage, for use in the efficient planning of the investigation.

3.3.3 Preliminary Ground Investigation

Preliminary ground investigation is a stage of the site investigation that is not always present. In many cases, for 'difficult' sites or major works, it is appropriate to conduct one in order to obtain information that can be useful for identifying the way that the main ground investigation must be carried out. It will consist of a few boreholes and insitu tests, the number and locations of which are selected to allow inferences about the subsurface conditions and stratigraphy at an acceptable cost.

3.3.4 Main Ground Investigation

The Main ground investigation is the extended investigation of the ground conditions of the site using boreholes and trial pits. Its purpose is to identify potential problems on the site and at the same time to verify and supplement information or inferences that are based on the previous stages of the investigation. During this stage detailed insitu and laboratory testing is carried out to establish the properties of the materials present.

An efficient ground investigation is the major factor for achieving a safe and economic design and its success is highly dependent upon the information obtained through the previous stages of the site investigation. These should indicate the types of problems that are likely to be encountered in specific areas of a site.

3.3.5 Ground Water Conditions

One of the aspects of the main site investigation is the determination of ground water level and the resulting pressures developed in the ground. It is essential to clarify the potential behaviour of ground water as it plays a major role during and after the construction of engineering works.

At this stage the engineer has to be very cautious because it is possible that ground layers will be subject to different pressures, particularly the ones that are, for example, separated by relatively impermeable layers.

3.3.6 Investigation of Special Cases

Investigation of special cases, which is not always present in a site investigation, occurs either during or after the main ground investigation and depends largely upon the hypotheses that the engineer has reached by that time, concerning the subsurface conditions. He or she might have identified possible problems or anomalies that require a more detailed survey of certain areas of the site.

3.4 Description of Soils

3.4.1 General

The description of soils that is included on borehole logs is the basis for recognising the materials and their stratification at the sampling points. The detail of the engineering description of soils will vary according to the phase of the investigation, the purpose of the borehole and the experience of the staff.

Disturbed and undisturbed soil samples are recovered from the boreholes and descriptions have to accompany every sample, after their visual examination by the engineer. This description can be modified later on, when the results from the laboratory testing will be known.

The interpretation of the ground conditions at a site is mainly dependent upon the soil descriptions and so it is considered very important that all descriptions follow a uniform presentation that conforms with the British Standards. However, before proceeding to describe the constituents of a soil description, there will be a brief discussion concerning the soil components and the means to identify them.

3.4.2 Composition of Soils

Soils consist of soil particles, mineral grains and sometimes organic matter, together with variable amounts of water and air and may be cemented or uncemented, [33]. The several soil types are divided into groups according to their size and these groups are presented in Table 3.1.

An engineer must be able to identify the constituents of a soil by visual examination of the soil sample. So a gravel will comprise particles larger than 2 mm (about the size of a very large pin head), while sand is noticeable for particles that cannot be broken down in the hand but are still visible to the naked eye. For finer materials, where particles are not distinguishable with the naked eye, a gritty feel, particularly on the teeth (not recommended due to possible toxicity), together with signs of dilatancy, will indicate a silt, whereas a clay will feature a smoothness and stickiness if moist, or would be very hard if dry. However the identification of the soil types participating in a soil sample is not straightforward because more than one usually participates, and the engineer must have a feel not only of the materials present but of their percentage of participation as well. This can be achieved by close and careful examination of the soil sample and is highly dependent on general knowledge of the materials present in the area under consideration, and the experience of the engineer.

Hence, it becomes apparent that the engineering descriptions of soils are subjective and, even if they follow BS5930, borehole logs completed by different persons in the same area might have differences.

Description	Coarseness	Particle Size (mm)
Boulders		> 200
Cobbles		60 - 200
Gravel	Coarse	20 - 60
	Medium	6 - 20
	Fine	2 - 6
Sand	Coarse	0.6 - 2
	Medium	0.2 - 0.6
	Fine	0.06 - 0.2
Silt	Coarse	0.02 - 0.06
	Medium	0.006 - 0.02
	Fine	0.002 - 0.006
Clay		< 0.002
Peat and Organic Material		Variable

Table 3.1   Identification of soils according to their particle size.

3.4.3 Soil Characteristics

The characteristics of a soil can be divided into the two following main categories :

o *Mass characteristics* that can be identified from examination of undisturbed materials, recovered either in the form of undisturbed samples or from exposures and excavations.

o *Material characteristics* that can be described from examination of disturbed samples and whose description should be confirmed after completing the laboratory testing.

The description of the above characteristics, that will be described in more detail below, as well as information concerning the geological formation and composition of deposits, form the full engineering descriptions of soils.

3.4.3.1   Mass Characteristics

o *Field Strength* or *Consistency* indicates the relative density of a granular soil or the strength of a fine soil. When silt is under consideration, it must be described in terms of strength if there is a high percentage of clay present or in terms of compactness or relative density if there is a high proportion of sand.

The descriptive terms used for identification of the field strength are presented in Table 3.2, together with the corresponding Standard Penetration Test N-values for granular soils and the undrained shear strength for fine soils, in accordance with BS5930, [6].

Descriptive Term	N-value
Very Loose	0-4
Loose	4-10
Medium Dense	10-30
Dense	30-50
Very Dense	>50

(a)

Descriptive Term	Undrained Shear Strength, Cu (kPa)
Very Soft	<20
Soft	20-40
Firm	40-75
Stiff	75-150
Very Stiff (Hard)	>150

(b)

Table 3.2   (a) Definition of descriptive terms for granular soils.

(b) Definition of descriptive terms for cohesive soils.

o *Bedding* refers to the terms describing spacing between bedding discontinuities or other discontinuities. It may also describe alternating layers with bedding discontinuities when the terms "interbedded" or "interstratified" are being used.

○ *Discontinuities* are indicative of joints, fissures, faults, shear planes and bedding planes and their description identifies surface texture, spacing between them and their orientation.

○ *State of Weathering* is often difficult to identify in soils, but an attempt can be made using the terms recommended for rock (Table 3.3, after Joyce, [33]). The change of colour within a single layer may be indicative of the degree of weathering of the soil strata.

Term	Description
Fresh	No visible sign of weathered material
Slightly Weathered	Discolouration indicates weathering on discontinuity surfaces
Moderately Weathered	Less than half of the material is disintegrated or decomposed to a soil
Highly Weathered	More than half of the material is disintegrated or decomposed to a soil
Completely Weathered	All rock material is disintegrated or decomposed to soil
Residual Soil	All rock material is converted to soil

Table 3.3   Terminology and the corresponding signs of weathered rock mass that can be used for soils.

3.4.3.2   Material Characteristics

○ *Colour* is identified by describing the dominant colour, the secondary colour, if needed, and possibly its luminence. For most soils the terms, shown in Table 3.4 (after Joyce, [33]), in the third column are indicative of their dominant colour and the adjectives shown in the first and the second column can be used when it is appropriate. When necessary, the words mottled or spotted can also be used.

○ *Particle Shape* can be defined using the terms presented in Table 3.5, concerning the description of angularity, form and surface texture of the particles and is produced after visual examination of the soil sample.

38

1	2	3
Light	pinkish	pink
Dark	reddish	red
	yellowish	yellow
	brownish	brown
	olive	olive
	greenish	green
		white
	greyish	grey
		black

Table 3.4   Descriptive terms for soils' colour.

Angularity	Form	Surface Texture
angular	equidimensional	rough
subangular	flat	smooth
subrounded	elongated	
rounded	flat and	
	elongated	
	irregular	

Table 3.5   Descriptive terms for particle shape.

○ *Grading and Plasticity* are the material characteristics from which a first estimate of the soil properties can be made. The terms used to describe the grading of granular soils are well graded and poorly graded consisting of uniform and gap gradings. The terms used to describe the plasticity of cohesive soils and the corresponding range of liquid limits are presented in Table 3.6.

Descriptive Term	Liquid Limit (%)
Low plasticity	< 35
Intermediate plasticity	35 - 50
High plasticity	50 - 70
Very high plasticity	70 - 90
Extremely high plasticity	> 90

Table 3.6   Definition of descriptive terms for plasticity.

o   *Soil Type* is the main constituent of every soil description and is based on the relative proportions of the different sized constituents. Each soil description may contain several soil types which are quantified by making use of the terms "slightly" and "very", which indicate different percentages of participation in a soil. The main constituent of the soil is the soil type that is usually written with capital letters. Therefore a description may have the form "slightly clayey silty very sandy GRAVEL" and one must be able to quantify such a description in order to assess the general behaviour of that soil.

The percentages of soil types participating in a soil description are defined in the form of ranges of percentages by the British Soil Classification System (B.S.C.S.), [6], in accordance with the descriptive terms and these ranges, together with some comments on the functionality of the B.S.C.S., are presented in the next section.

3.4.4   British Soil Classification System

Soils are classified into categories indicative of their properties, in accordance with the B.S.C.S., from the terms used in their engineering descriptions. Although most of the material and mass characteristics may be described in a uniform way, confusion is observed when describing the constituents of fine soils and this can be partly attributed to the way that B.S.C.S. handles such descriptions, as is outlined by Child, [11].

According to B.S.C.S. for coarse grained soils (where the dominant soil type is sand, gravel, cobbles or boulders) the dominant soil type, which is normally given as the soil's name in capitals (e.g. GRAVEL), indicates 65-100% grains of gravel size. The descriptive term "very" (e.g. very silty) indicates 15-35% grains of silt size. The name of the soil type followed by the ending "-y" (e.g. silty) indicates that the soil consists of 5-15% grains of silt size. Finally, the descriptive term term "slightly" (e.g. slightly silty) indicates that the soil consists of 0-5% grains silt size. However, if the

40

secondary constituent is a coarse soil then the descriptive term "very" (e.g. very sandy) indicates 20-35% grains of sand size. The name of the soil type followed by the ending "-y" (e.g. sandy) indicates that the soil consists of 5-20% grains of sand size. Finally, the descriptive term term "slightly" (e.g. slightly sandy) indicates that the soil consists of 0-5% grains sand size.

For fine grained soils (where the dominant soil type is clay or silt) the B.S.C.S. indicates that for the name of the soil type followed by the ending "-y" the percentage should be 35-65%. The terms "slightly" or "very" are not defined for the fine grained soils. This scheme proposed by the British Standards Institution, [6], is presented in Table 3.7.

Soil Description	Percentage for coarse grained soils with coarse secondary constituent(s) (%)	Percentage for coarse grained soils with fine secondary constituent(s) (%)	Percentage for fine grained soils (%)
Slightly	0-5	0-5	
-y	5-20	5-15	35-65
Very	20-35	15-35	
Soil Name	65-100	65-100	35-100

Table 3.7.   Percentage ranges of coarse and fine soils according to the descriptive terms.

The B.S.C.S. states that a fine soil is identified as either a silt or clay and that depends on whether it plots below or above the A-line, in plasticity terms. Further, it proposes that there is not a fine soil that can be termed "silty clay" or "clayey silt" although inside BS5930 are examples where these descriptions are used. It also mentions that if it is not possible to distinguish between clay and silt then the 'vague' term fine soil must be used, which is a result of the elimination of the terms mentioned above. Hence, it becomes clear that a realistic description of the ground conditions is not achievable in several cases. Additionally, according to B.S.C.S., when a fine soil contains less than 35% of coarse material, the description of this material must not

41

participate in the overall soil description. Together with the omission of the words "slightly" and "very" from the vocabulary used to describe fine soils, one can only describe composite fine soils as either sandy or gravelly.

The inconsistencies that are present within B.S.C.S. make it difficult for a uniform representation scheme of soil descriptions to exist. Therefore it often happens in current practice in site investigation that descriptions do deviate from the standards that B.S.C.S. has set. One can often see borehole logs where the terms "silty clay" and "clayey silt" are used, and as a result of the above, the quantification of such terms becomes mainly subjective, in an area that there is a great need for objective and uniform interpretation of results. Hence, it becomes evident that alternative schemes for representing soil descriptions have to be adopted if a uniform way of describing soils is to be achieved. Such an alternative scheme that is in general accordance with the B.S.C.S., but tries to avoid the inconsistencies discussed earlier is presented by Norbury *et al*, [46].

3.5   Reporting Borehole Information

The first piece of borehole information that becomes known during the ground investigation is in the form of the driller's daily field report. As boring proceeds the driller must record all the strata encountered together with their corresponding description and the depths at which changes of strata occur. At this stage all information recovered must be recorded as its importance is not yet established.

The information recorded on the driller's report will largely contribute to the production of the final borehole log by the engineer. This final log is also based on visual examination of the soil samples by the engineer, insitu and laboratory test results and on knowledge about the geology of the site. The engineer can modify, if needed, the soils' descriptions in the light of the tests results in order to produce what his experience leads him to feel is an accurate picture of the ground conditions.

The information included in the final borehole logs must be described in a consistent way as these documents will be the basis for the interpretation of the subsurface conditions.

3.6   Correlation of Borehole Information

3.6.1   General

The process of characterizing a site involves the correlation of observations at discrete points such as boreholes and trial pits. Correlation is the recognition of equivalent layers or stratigraphical horizons at different observation areas. It can assist in allowing the construction of hypotheses about the subsurface conditions in order to decide the way that the site must be treated for design and construction purposes.

The correlation of borehole information can be done either by interpolating observations made at two different points or by extrapolating observations from one sampling point, [19]. The greater the distance from the borehole(s) the less reliable the correlation is. Hence it becomes evident that such a procedure is vulnerable to misinterpretation of the ground conditions and extreme caution is needed before the inferences reached are considered valid, as the implications involved, with imprecise conclusions about the soil profile, are both social and financial, [70]. At its simplest, the operation involves linear matching of layers observed at different boreholes but such a simplistic approach is not always the case, [67]. In complex conditions, knowledge of geological processes is often necessary to arrive at a solution.

There is certainly not a standard way of handling the problem of correlation or interpretation. The method of tackling it is largely dependent on the experience of

the engineer, the particular site under consideration and the type of the proposed works.

3.6.2 Procedure and Problems

The completion of the Main ground investigation will signal the beginning of the inference procedure concerning the ground conditions. The engineer must be able to interpolate or extrapolate from discrete observations and build up a three dimensional visualization of the soil profile. The site will be characterised either with respect to a general solution or according to the nature of the project to be completed. The familiarity of the engineer with the loads, likely to be imposed by the structure, the settlements that can be allowed and the effect of the proposed works on the development of the pore water pressures is a key point in assessing possible problems.

The first step in this process is the site characterization with respect to the materials present. This is achieved after having a brief look, in no particular order, at the borehole logs trying to assess the relative percentages of fine and coarse soils and identify the dominant materials the properties of which will probably govern the mechanical behaviour of the site. Depending on the general types of deposits at the site, one can draw conclusions about the geomorphology of the area. For instance, finer materials might suggest deposition from calm waters, whereas the presence of large amounts of gravel might indicate a more turbulent flow of water, [47].

The next step is a search for noticeable trends and marker beds. An indication of the existence of marker beds can occur by observing the same soil strata at several boring logs at about the same elevation level. The number of sampling points per area and the relative thickness of a soil layer to the sampling interval are the deciding factors for extracting conclusions about marker beds. For instance at a site with dimensions of 3000 metres by 1800 metres, an expert indicated that about 6 to 8 borings containing the same layer, well scattered among the corners and the centre of the site,

44

would indicate the presence of a marker bed, [47]. The identification of marker beds is an essential part of site characterization because their presence simplifies the problem of correlation. However a marker bed can appear at different depths at different boreholes, something that may suggest the existence of wavy ground. In such cases additional boreholes may be required in order to establish the geometric properties of the layer.

The observation of existing rock beds together with an assessment of their quality may be the next step in correlating borehole information. This might help in the solution of a foundation problem by considering piles resting on the bedrock. The procedure that an expert follows up to this point serves mainly in the production of a list of site characteristics that can assist with the correlation problem which has to deal with detailed interpretation of the ground at unobserved and unsampled areas.

After identifying possible marker beds one has to start looking at pairs of adjacent boreholes to make inferences about the soil profile between them, based on the similarity of the layers observed and on possible irregularities of the subsurface that are known from previous stages of the site investigation process. Some of the situations that one may have to deal with like dipping, faulting or folding are presented in Figure 3.1. Useful examples of such conditions together with the misinterpretations that may occur as a result of incorrect correlation are given by Dumbleton and West, [19], and Thomas, [67].

Therefore, the nature of the problem guides the expert to consider not 'one' solution but a set of solutions that have to be identified and subsequently evaluated. The hypothesis that will, finally, be adopted must conform with the information gathered about the geology of the area and must be consistent within sets of neighbouring boreholes. Critical points of the correlation may require additional information that can be obtained through supplementary boreholes in certain areas. In practice, the

procedure of refining and confirming the hypotheses reached may be carried out continuously, even during construction, especially when dealing with cuttings or excavations where large amount of information about the subsurface conditions is revealed.

In conclusion, one must always keep in mind that the correlation of information from borehole observations is a highly subjective procedure. The experience of the engineer conducting it, together with a good understanding of the area's geology, are the major factors in avoiding misinterpretation of the ground, thus leading to a safe and economic design which is the main objective of the site investigation.
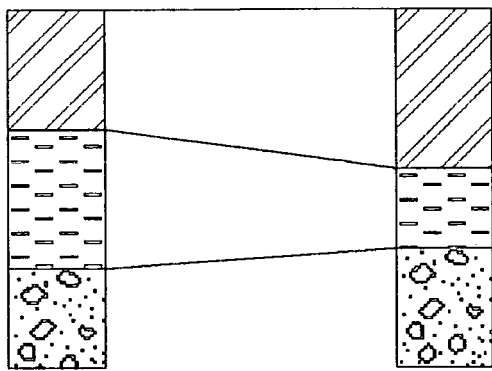
(a)   Thinning layers
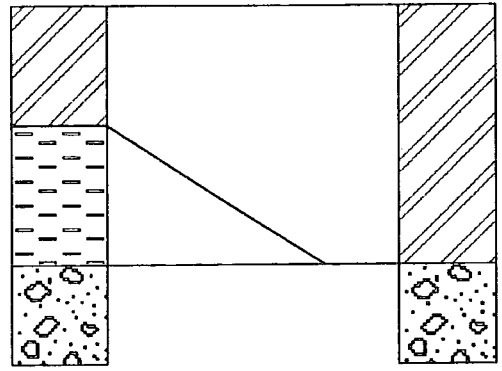
(b)   Layer dying out
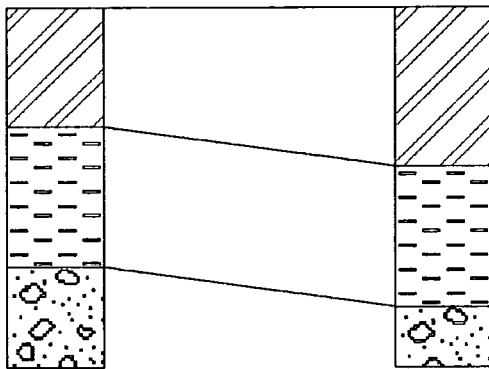
(c)   Dipping layers
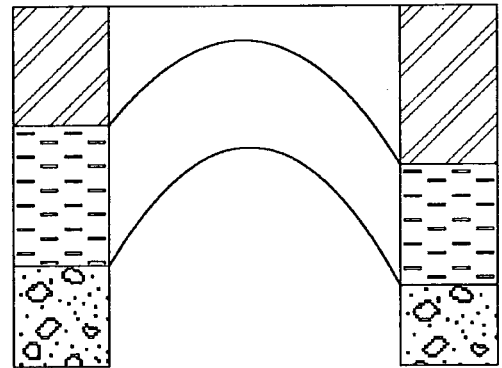
(d)   Folding
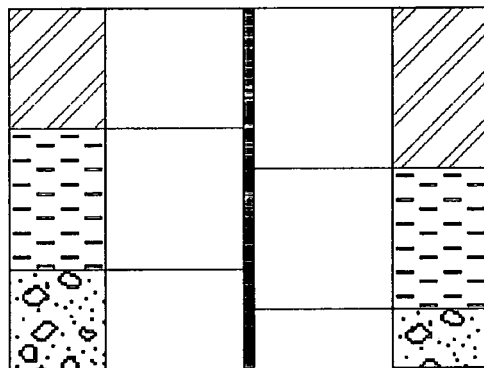
(e)   Faulting

Figure 3.1   Possible interpretations of ground conditions between adjacent boreholes.

CHAPTER 4

A METHODOLOGY FOR COMPARING SOILS

4.1 Introduction

One aspect of the interpretation of ground conditions is the ability to recognise the similarity of two soils based on engineering descriptions. A soil layer may be observed in two boreholes at different depths (and having different thicknesses) and a decision has to be made as to whether the two observations represent a continuation of the same soil layer, [68]. Engineering descriptions of soils are complex and making comparisons between two descriptions is far from straightforward.

A prototype system has been developed for interpreting ground conditions and is described in this Chapter. Initially the components of an engineering soil description, which need to be represented in a KBS, are presented. A parser module of the system for breaking down complex descriptions into their constituent parts is then described. Then, through a Value Assignment and SImilarity Calculation (VASIC) module that is attached to the parser, numerical values are assigned to the descriptive terms which allow comparisons to be made between different features. Initially, comparisons between individual soils are discussed. However it is often necessary to make comparisons between layers containing multiple soils, and a way of dealing with this problem is presented.

In order to make the comparison between soils the concept of a Similarity Number is used. Similarity Numbers are calculated for each of the features and have a range of

0-100%; a value of 100% indicating identical features. The individual Similarity Numbers are combined using weighting factors (which represent the importance of each feature) to give an overall Similarity Number which represents a comparison based on all aspects of the description. Finally, in the discussion some of the improvements which could be made are identified.

4.2   Representing Soil Descriptions

Developing Knowledge-Based Systems in Geotechnical Engineering involves representing the engineering descriptions of soils, [42, 72]. A full engineering description is a structured list of varying types of information. The required pieces of information contained in a description have been given in BS5930, [6]. A slightly extended methodology for soil description is given by Burland, [7], based on the work by Jennings and Brink, [32]. The main components of a soil description can be set out as follows :

M	- Moisture condition
C	- Consistency
C	- Colour
S	- Structure
S	- Soil type
O	- Other features
O	- Origin
W	- Ground water conditions.

The first six features are factual items that a field engineer can identify from examining a soil sample. Origin requires interpretation based on knowledge of the geology of the area in which the soil exists. Geology is therefore used as a tool for characterising the site even before commencing exploration. Origin is a property of the GEOLOGICAL HORIZON, that is, a sequence of LAYERS having the same

geological origin. Ground water conditions apply to a complete profile rather than being specific to a particular LAYER.

Individual LAYERS can be recognised within the profile and can each be defined by depth and thickness and may be described by a broad classification or by a full engineering description. Each LAYER can contain one or more SOILS. Where multiple soils exist within a layer, and it is not possible to separate them into individual layers, (e.g. *silt* interbedded with *clay*) the additional term LAYER STRUCTURE is used to describe how they are related within the layer. A SOIL can be represented by MOISTURE, CONSISTENCY, STRUCTURE and SOIL TYPE as shown in Figure 4.1, [42, 72]. Each SOIL TYPE has properties of Shape, Texture, Colour etc. Of these components the factors which are considered to be dominant in the comparison of two soils are : Soil Type, Consistency, Structure and Colour.

In the representation scheme put forward by Toll *et al*, [72], each SOIL TYPE is associated with an AMOUNT (Fig. 4.1). If the soil type is the dominant soil type then AMOUNT is given as Main. For the descriptive term *very* the AMOUNT is given as Major. For the soil's name followed by the ending -*y* AMOUNT is given as Secondary. For the descriptive term *slightly* AMOUNT is given as Minor. Other descriptive forms (other than those recommended by British Standard 5930, [6]) can also be represented in this way.

```
                          ┌──────────┐
                          │   Soil   │
                          └──────────┘

   ┌───────────┐    ┌──────────┐   ┌───────────┐   ┌──────────────┐
   │ Structure │    │ Moisture │   │ Soil type │   │ Consistency  │
   └───────────┘    └──────────┘   └───────────┘   └──────────────┘

 ┌─────────┐   ┌─────────┐  ┌────────┐                  ┌──────────┐
 │ Spacing │   │ Surface │  │ Shape  │                  │ Texture  │
 └─────────┘   └─────────┘  └────────┘                  └──────────┘

  ┌─────────────┐ ┌─────┐    ┌──────────┐        ┌──────────┐
  │ Orientation │ │ Dip │    │ Amount   │        │ Grading  │
  └─────────────┘ └─────┘    └──────────┘        └──────────┘

                    ┌──────────────┐    ┌───────────────┐
                    │ Main Colour  │    │ Secon. Colour │
                    └──────────────┘    └───────────────┘

                         ┌──────────────────┐
                         │ Colour Modifier  │
                         └──────────────────┘
```
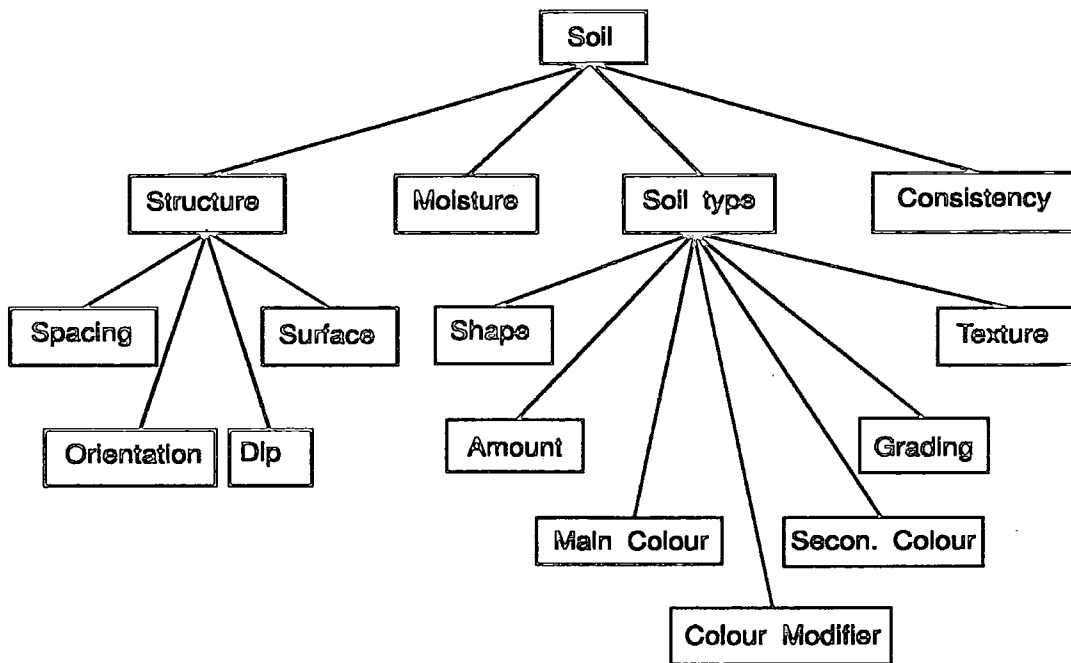
Figure 4.1   The components of a soil description.

4.3   Parsing Soil Descriptions

Engineering descriptions of soils are complex expressions containing no verbs, only adjectives and nouns. Therefore, commercially available parsers which are mainly involved with the parsing of correctly structured sentences were not very useful as they would need major modifications to handle the problem and at the same time their full functionality would not be utilised. Hence, it was decided to develop a parser that concentrates on the problem of soil descriptions and to structure it in a modular manner so that it could be easily extended if needed. This was implemented in PDC Prolog, [51, 52, 53].

The parsing of the soil descriptions is based on the recursive predicate *parse* that is actually the heart of the parser. The *parse* predicate receives a description as input in the form of an argument of string type and by using the PDC Prolog standard predicate *fronttoken*, the first lexical token of the string is separated from the rest. A

token in PDC Prolog is identified as a sequence of characters that constitute either a name according to normal PDC Prolog syntax, or a number, or a nonspace character. Thus, the first token of the string "silty sand" is "silty" which is the first sequence of nonspace characters followed by a space. Then, by using the standard predicate *member* it can be identified whether that token is a member of certain lists that contain different geotechnical terms which are stored as facts in the fact section of the program. The facts are categorised according to the different soil characteristics that they indicate. The *parse* clauses are recursive, so every time that one token is processed, *parse* calls itself, having the rest of the string as an input argument.

There are twenty seven *parse* clauses in the program, each one identifying different terms used in engineering soil descriptions. The parser can process soil descriptions containing terms referring to Soil Type, Consistency, Structure, Colour, Layer Structure, Plasticity, Texture, Particle Size Distribution, Grain Size and Shape. A full listing of the parser is contained in Appendix A, but a brief description of the *parse* clauses for each of the above categories follows.

○ Soil Type

There are three *parse* clauses referring to Soil Type. The first one identifies the main soil type by calling the fact-lists :

soil_type(inorganic, [clay, silt, sand, gravel, cobbles, boulders])

soil_type(organic, [peat])

These contain inorganic and organic soil names and the parser checks whether the token is a member of these lists. If the token is a member, the clause appends the soil type with its corresponding amount ("main") in a list called the Soil_list. The second clause treats the lesser constituents if they occur in the description <u>before</u> the main soil type by using the words *slightly*, *very* or the soil's name followed by the ending -*y*. It checks in the fact-lists :

modifier(minor, soil_before, [slightly])

modifier(secondary, soil_before, [])

modifier(major, soil_before, [very])

soil_type(Nature, List)

By combining the modifier string and the soil type it identifies the expression with its relative amount (e.g. "very silty", Amount = major). When that is found, it appends the soil name and its amount in the Soil_list. Finally, the third *parse* clause checks for the lesser constituents that are referred to <u>after</u> the main soil type. In that case the fact-lists :

modifier(minor, soil_after, [occasional, little, trace, scatter, infrequent, isolate])

modifier(secondary, soil_after, [some, few])

modifier(major, soil_after, [numerous, many, frequent])

are called and the amount is set accordingly and is stored as an output argument of *parse*. The rest of the string is passed to the recursive *parse* clause and the soil type is identified by the first soil type version of the clause. This receives the amount, that has been previously defined, as an input argument and the soil type and its corresponding amount are appended in the Soil_list which is expanded every time that a new soil name is met.

o   Consistency

There are two *parse* clauses for identifying consistency in the parser. The first one considers three input tokens and examines whether consistency is referred to as a range (e.g. "firm to stiff") by checking if the first and the third are members of the fact-strings *cons*(Term, Value) that contain the several terms used for describing consistency (and the corresponding SPT N value which is not used at this moment). The second token must be a member of the fact-list:

range([and, "-", to, becoming])

Then, the recognised consistency terms are entered in the Cons_list where these are stored. The second *parse* clause is able to identify a single term for consistency. It calls the *cons* fact-strings in order to identify the consistency term and append it in the Cons_list.

o   Structure

There are six *parse* clauses for identifying structure, three of them refer to bedding spacing and three of them to spacing of other discontinuities. The first one calls the fact-lists :

bed_spacing([thin, thick])

bedding([bedded, laminated])

and combines them in order to match them with expressions such as "thickly bedded". If this happens the description is stored in a list called Bed_list. The second and third clauses perform the same action but they also consider the quantifiers "medium" and "very" respectively. The other three clauses refer to spacing of other discontinuities and perform similar actions by calling the fact-lists :

spacing([wide, close])

discontinuity([fissures, joints])

thus processing expressions such as "very closely spaced fissures". The string will then be stored in a list called Spacing_list.

o   Colour

There are four *parse* clauses for colour. The first one calls the fact-list :

colour([red, pink, purple, yellow, brown, green, blue, white, grey,

black]).

The parser checks whether the string is a member of that list and it appends it in a list called Colour_list together with its quantifier (main) to indicate main colour in this case and the amount of the soil type that the colour refers to. The second clause calls the same fact-list of colours and after changing the ending using either -ish or -y, it recognises the secondary colour (e.g. "reddish"). The third parse clause identifies ranges for colour by calling in addition the fact-list *range*(List) and in that way descriptions like "red and brown" can be parsed. Finally, the fourth one considers the possible colour modifiers by calling the fact-list :

54

modifier(colour,[light, dark])

This is then stored in the Mod_list together with the amount of the soil type that it is referred to.

o   Layer Structure

When a soil layer contains more than one soils, this is identified by the only non-recursive *parse* clause, that calls the fact-list :

layer_structure([interbedded, interstratified, pockets, lenses, inclusions])

If such a term is recognised, the parsing of the preceding terms is considered complete and the lists created are the output arguments identifying the first of the two soils participating in the soil description. Then *parse* is called again having as input string the rest of the description. A *parse* clause, after calling the fact-list *layer_structure*, identifies the layer structure term and appends it in a list which is called Layer_str_list and this is actually a new Soil_list for the second soil. The rest of the string is parsed by the *parse* clauses that were described above under Soil Type and the identified terms are appended in the Layer_str_list which is indicative of the second soil participating in the soil description.

o   Plasticity

There are two *parse* clauses for plasticity. The first one calls the fact-list *plasticity*(List) that contains terms such as "plastic" or "plasticity" and the fact-lists *modifier*(Amount, plasticity, List) which contain terms that indicate different levels of plasticity. It then stores the plasticity term in a list called Other_list together with its quantifier. The second clause calls again the fact-list *plasticity*(List) and the level of plasticity is set to "medium". The plasticity term and its level are stored in the Other_list which was introduced for storing information for the remaining soil features.

55

- Shape

    The *parse* clause for shape calls the fact-list *shape*(List) with terms relating to shape. The corresponding term is then stored in the Other_list.

- Texture

    The *parse* clause for texture calls the fact-list *texture*(List) that contains appropriate terms. That is then stored in the Other_list.

- Distribution

    The *parse* clause for distribution calls the fact-lists :

    distribution([poor, well, gap, uniform])

    grading([grade])

    and after combining them it appends the result in the Other_list.

- Size

    The *parse* clause for size calls the fact-list :

    size([fine,medium,coarse])

    and stores the appropriate term in the Other_list.

If none of the preceding clauses is activated this means that a word contained in the soil description is either wrong or is not included in the terms that the parser can recognise. The final *parse* clause therefore produces the expression "Ignoring term" before the rest of the description is parsed. Another *parse* clause checks the input argument, to see if it contains an empty string. If so it terminates the parsing.

An example of parsing a complex soil description is diagramatically shown in Figure 4.2.

```
" loose yellowish brown slightly silty gravelly sand "
```

```
parse ───────────────────→ ["loose"] ──────────────────→ Cons_list = ["loose"]
```

```
" yellowish brown slightly silty gravelly sand "
```

```
parse ───────────────────→ ["yellowish"] ──────────────→ Col_temp = ["secondary", "yellow"]
```

```
" brown slightly silty gravelly sand "
```

```
parse ───────────────────→ ["brown"] ──────────────────→ Col_temp = ["main", "brown",
                                                                     "sec.", "yellow"]
```

```
" slightly silty gravelly sand "
```

```
parse ──────→ ["slightly"] ──────→ Soil_list_temp = ["minor", "silt"]
       ──────→ ["silt"]
```

```
" gravelly sand "
```

```
parse ──────→ [ ] ──────────────→ Soil_list_temp = ["sec.", "gravel",
       ──────→ ["gravel"]                            "minor", "silt"]
```

```
" sand "
```

```
parse ──────→ ["sand"] ──────────→ Soil_list_temp = ["minor", "silt",
                                                      "sec.", "gravel",
                                                      "main", "sand"]
```

```
" "
```

```
parse ─────────────────────────→  Cons_list = ["loose"]

                                   Colour_list
                                   ["main", "brown", "sec.", "yellow"]

                                   Soil_list
                                   ["main", "sand", "sec.", "gravel", "minor", "silt"]
```

Figure 4.2   Flow chart of the parser module for a given soil description.

The modular approach used in the *parse* clause enables modifications and extensions to be easily included. New *parse* clauses can be incorporated into the system for the identification of terms that are not currently present in the facts, without a large effort. Equally, the existing vocabulary can be simply extended by modifying the fact-lists. Thus, it would not be very difficult to extend the system for processing a larger range of soil descriptions.

Finally, the parser module is to be utilised as a critical part of the front end of a proposed site investigation database, the soil description being broken down into its constituent parts and automatically entered into the relevant tables in the database. Currently the database is under design as a part of a system to aid in the interpretation of ground conditions from site investigation information and geotechnical design, [71].

4.4 VALUE ASSIGNMENT AND SIMILARITY CALCULATION MODULE

As was mentioned earlier, the prototype system comprises a parser module for extracting the different pieces of information contained in a soil description and a Value Assignment and SImilarity Calculation module (VASIC), that allocates values to the descriptive terms and performs the numerical comparisons between pairs of soil descriptions, when required.

The methods of assigning values to the qualitative terms and how those are used to calculate similarity numbers is described in this section. The listing of the VASIC module is included in Appendix A (after page A15).

4.4.1 Soil Type

4.4.1.1 Coarse and Fine Grained Soils

Soil type is the major factor in determining how closely related two observations of soils are. The main soil types are: *Clay, Silt, Sand, Gravel, Cobbles* and *Boulders*. Obviously a soil in which the main soil type is GRAVEL is dissimilar to one which is CLAY. Also a SAND is more similar to a GRAVEL than it is to a CLAY. It would not be difficult to set up a series of rules, based on main soil type, which could differentiate between the similarities of two soil descriptions. However it is more difficult to establish such rules for complex soil descriptions where, in addition to the main soil type, one has lesser components to deal with. For example, if one considers the following descriptions :

(1) Very silty, sandy, slightly gravelly CLAY

(2) CLAY and BOULDERS

(3) Very clayey, sandy, slightly gravelly SILT

Although descriptions (1) and (2) share the same main soil type CLAY they are less likely to belong to the same soil layer than (1) and (3), where although the main soil types are different, the lesser soil types reinforce the similarity.

A previous attempt at this problem was made by Norkin, [47], and Rehak *et al*, [55], where rules were set up, based on the descriptive terms directly, and comparison was done by matching the descriptions to produce certainty factors. However, Rehak *et al*'s approach was found to be unsuitable because, due to the possible complexity of descriptions, it becomes difficult to develop a set of rules which could handle all the combinations of soil type. An alternative therefore had to be found in order to break down complex soil descriptions into their constituent parts and then to give a measure of similarity between them. The solution used was to construct a notional particle size distribution from the descriptive terms. A distribution is defined as cumulative

percentages of particle size, the traditional way of representing particle sizes in Geotechnical Engineering.

According to the British Soil Classification System (B.S.C.S.), [6], for coarse grained soils (where the dominant soil type is sand, gravel, cobbles or boulders) the main soil type, which is normally given as the soil's name in capitals (e.g. GRAVEL), indicates 65-100% grains of that size (in this case gravel). As discussed earlier, the lesser soil types are described using descriptive terms such as *slightly* (e.g. slightly clayey GRAVEL) which indicates that the soil also contains 0-5% grains of clay size. The name of the lesser soil type followed by the ending -*y* (e.g. clayey GRAVEL) indicates that the soil contains 5-15% grains of clay size if it refers to a fine lesser soil or 5-20% if it refers to a coarse lesser soil type. The descriptive term *very* (e.g. very clayey GRAVEL) indicates 15-35% grains of clay size if it refers to a fine lesser soil or 20-35% if it refers to a coarse lesser soil type..

For fine grained soils (where the dominant soil type is clay or silt) the B.S.C.S. suggests that for the lesser constituents, the name of the soil type followed by the ending -*y* indicates the percentage should be 35-65%. The ranges which correspond to different amounts are shown in Table 4.1.

Soil Description	Amount	Percentage for coarse grained soils (%)	Percentage for fine grained soils (%)
Slightly -y Very	Minor Secondary Major	0-5 5-15 or 20 15 or 20-35	35-65
Soil Name	Main	65-100	35-100

Table 4.1   Percentage ranges according to amount of soil type

The terms *slightly* or *very* are not defined for the fine grained soils in B.S.C.S. This has lead to some confusion (and anomalies) in the use of the descriptive terms for fine

grained soils, [11]. In the following the terms *slightly* and *very* have been used for fine grained soils and appropriate percentages have been adopted.

In order to construct a particle size distribution it was necessary to represent the percentage passing by a single value instead of a range of values as shown above. The allocation of values to percents of soil types according to the amounts participating in a soil description was found to be complicated. The values could not be uniquely defined for a given descriptive term but depended on the number (and amounts) of other soil types given in the description. Therefore a matrix of percentage values was defined for different combinations of amounts present in the description.

This was implemented by defining pairs of integer lists in PDC Prolog. Every pair was composed of an Amount-list and a Percent-list, each one containing four integers. The first integer in each list represents the Main soil types, the other integers represent the Major, Secondary and Minor soil types, in that order. The Amount-list is indicative of the quantity of amounts that may participate in a description. The Percent-list contains the percentage values which correspond to the amounts in the Amount-list and satisfy the British Soil Classification System (with some modification for fine grained soils). The allocation of percents was produced by selecting values close to the average number from the ranges specified in the B.S.C.S. and by allocating these numbers primarily to the Major, Secondary and Minor soil types. The remaining percent in order to achieve 100% was then assigned to the main soil type. Some adjustment was needed to ensure, for coarse grained soils, that the main soil type was the largest percentage. The pairs of Amount and Percent lists defined are shown in Table 4.2a for coarse grained soils and Table 4.2b for fine grained soils.

Therefore the Amount-list [1,2,0,1] for a coarse grained soil indicates a soil description that is formed by one Main soil type, two Major soil types, no Secondary soil type and one Minor soil type. The corresponding Percent-list is [66,16,0,2] indicating 66% grains of the Main soil type, 16% for each of the two major soil types, no percent for Secondary soil types and 2% of the Minor soil type.

Amount-list	Percent-list	Amount-list	Percent-list
[1, 0, 0, 0]	[100, 0, 0, 0]	[1, 2, 1, 0]	[65, 15, 5, 0]
[2, 0, 0, 0]	[50, 0, 0, 0]	[1, 1, 2, 0]	[65, 15, 10, 0]
[1, 1, 0, 0]	[75, 25, 0, 0]	[1, 1, 0, 2]	[69, 25, 0, 3]
[1, 0, 1, 0]	[90, 0, 10, 0]	[1, 2, 0, 1]	[66, 16, 0, 2]
[1, 0, 0, 1]	[97, 0, 0, 3]	[1, 0, 1, 2]	[84, 0, 10, 3]
[1, 1, 1, 0]	[65, 25, 10, 0]	[1, 0, 2, 1]	[77, 0, 10, 3]
[1, 1, 0, 1]	[72, 25, 0, 3]	[1, 0, 0, 3]	[91, 0, 0, 3]
[1, 0, 1, 1]	[87, 0, 10, 3]	[1, 0, 3, 0]	[70, 0, 10, 0]
[1, 1, 1, 1]	[65, 23, 10, 2]	[1, 3, 0, 0]	[40, 20, 0, 0]
[1, 0, 0, 2]	[94, 0, 0, 3]	[2, 0, 0, 1]	[48, 0, 0, 4]
[1, 0, 2, 0]	[80, 0, 10, 0]	[2, 0, 1, 0]	[45, 0, 10, 0]
[1, 2, 0, 0]	[66, 17, 0, 0]	[2, 1, 0, 0]	[37, 26, 0, 0]

Table 4.2a   Pairs of Amount-lists and Percent-lists for coarse grained soils

Amount-list	Percent-list	Amount-list	Percent-list
[1, 0, 0, 0]	[100, 0, 0, 0]	[1, 0, 0, 2]	[50, 0, 0, 25]
[2, 0, 0, 0]	[50, 0, 0, 0]	[1, 0, 1, 1]	[40, 0, 35, 25]
[1, 1, 0, 0]	[35, 65, 0, 0]	[1, 0, 2, 0]	[40, 0, 30, 0]
[1, 0, 1, 0]	[50, 0, 50, 0]	[1, 2, 0, 0]	[36, 32, 0, 0]
[1, 0, 0, 1]	[65, 0, 0, 35]	[1, 0, 1, 2]	[40, 0, 30, 15]
[1, 1, 1, 0]	[35, 40, 25, 0]	[1, 1, 0, 2]	[35, 35, 0, 15]
[1, 1, 0, 1]	[35, 45, 0, 20]	[1, 0, 2, 1]	[35, 0, 25, 15]
[1, 1, 2, 0]	[35, 25, 20, 0]	[1, 2, 0, 1]	[35, 30, 0, 5]
[2, 0, 0, 1]	[40, 0, 0, 20]	[1, 2, 1, 0]	[35, 25, 15, 0]
[2, 0, 1, 0]	[35, 0, 30, 0]	[1, 1, 1, 1]	[35, 35, 20, 10]
[2, 1, 0, 0]	[35, 30, 0, 0]		

Table 4.2b   Pairs of Amount-lists and Percent-lists for fine grained soils

An example of a soil description with the corresponding Amount-list and Percent-list is given below. The particle size distribution is shown in Fig. 4.3, together with more examples of particle size distributions for some typical descriptions of soils.
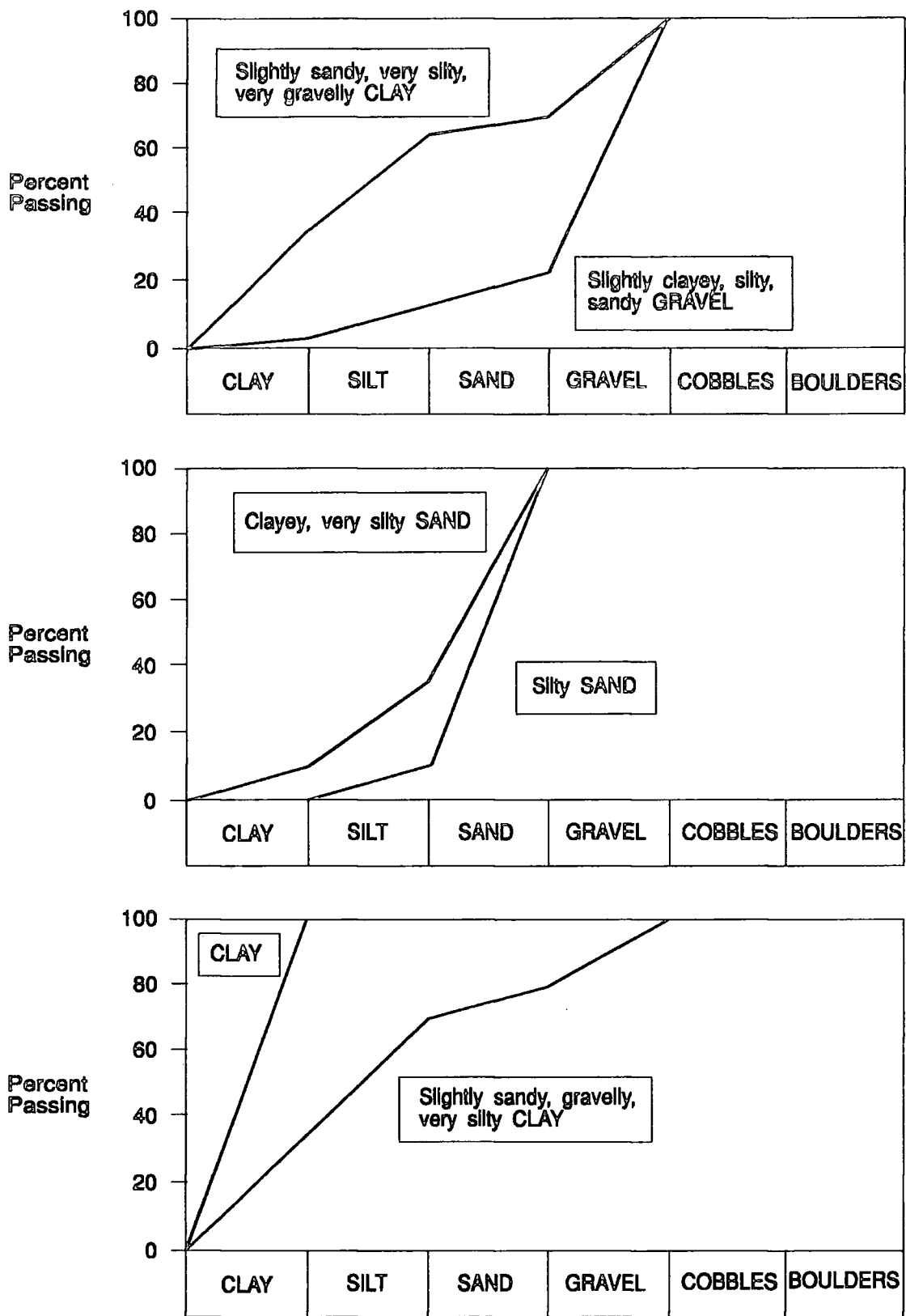
Figure 4.3   Examples of notional particle size distributions generated from soil

descriptions.

Soil description: *Slightly sandy, very silty, very gravelly CLAY.*

Main soil type:	Clay
Major soil type:	Silt, Gravel
Secondary soil type:	-
Minor soil type:	Sand

Therefore the Amount-list is [1,2,0,1] and since the soil is fine grained (main soil type: CLAY), the Percent-list is [35,30,0,5].

The particle size distribution can then be compared numerically with another distribution to give a Similarity Number. The comparison between the two distributions was made by observing the difference in percentage at a number of particle sizes. The similarity is given as 100 minus the average absolute difference. For n points on the particle size distribution

$$\text{Similarity Number} = 100 - \frac{1}{n} \sum_{1}^{n} |\text{Percentage Difference}|$$

This number is presently calculated using n=6, that represent the points that define the limits between the six different inorganic soil types (particle diameters of 0.002, 0.06, 2, 60, 200 and >200 mm). The Similarity Number has a value between 0 and 100, a higher number implying increased similarity. Some examples of comparisons together with the calculated Similarity Numbers are given in Figure 4.4. In the third example a "very silty clayey SAND", is compared to a "silty SAND", indicating the following percentage differences at the six points identified above:

Particle Diameter (mm)	0.002	0.06	2	60	200	> 200
Percent Passing of Soil 1	10	35	100	100	100	100
Percent Passing of Soil 2	0	10	100	100	100	100
Percentage Difference	10	25	0	0	0	0

The Similarity Number is : 100 - 1/6 (10 + 25), so Similarity Number = 94.

64

Similarity = 76

Percent Passing

Slightly sandy, gravelly,
very silty CLAY

Slightly clayey, silty,
sandy GRAVEL

CLAY	SILT	SAND	GRAVEL	COBBLES	BOULDERS

Similarity = 87

Percent Passing

Slightly sandy, gravelly,
very silty CLAY

Very silty, clayey SAND

CLAY	SILT	SAND	GRAVEL	COBBLES	BOULDERS

Similarity = 94

Percent Passing

Very silty, clayey SAND

Silty SAND

CLAY	SILT	SAND	GRAVEL	COBBLES	BOULDERS
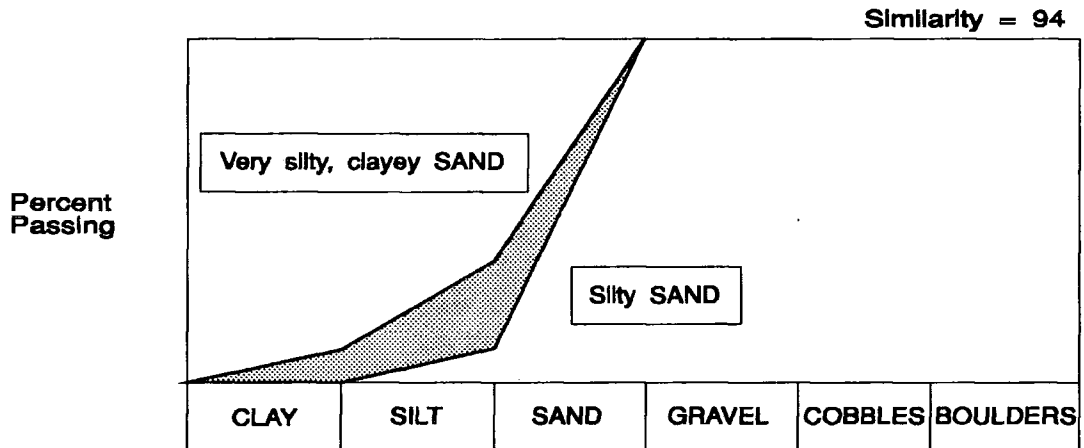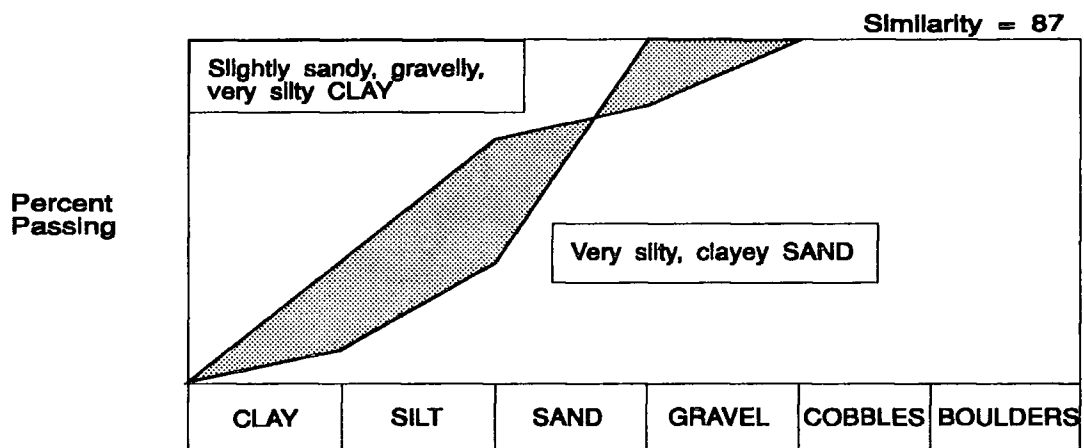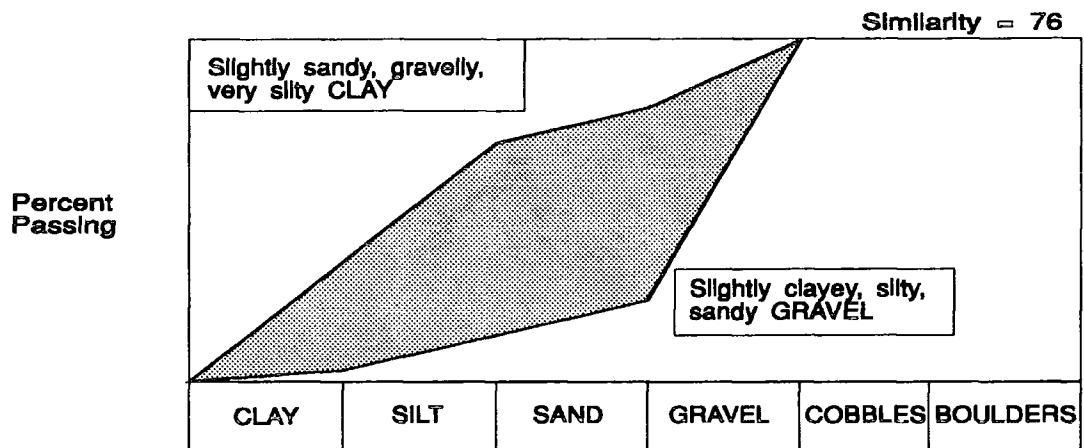
$$\text{Similarity Number} = 100 - \frac{1}{n} \sum_{1}^{n} |\text{Percentage Difference}|$$

Figure 4.4   Examples of comparisons between soils in terms of soil type.

4.4.1.2   Peat

Often, one has to deal with highly organic materials such as peats. Peats consist predominantly of plant remains, usually dark brown or black in colour and with a distinctive odour, [13]. Due to the variability of its particle size a different procedure was adopted for comparing soils consisting of peat. The comparison takes place by comparing separately the main soil types and the lesser constituents, thus producing two Similarity Numbers. These are then combined with a weighting of 70:30 towards the main soil type, to give the final Similarity Number.

For example the comparison between a silty SAND and a silty PEAT gives :

Description	silty SAND	silty PEAT	
Main soil type	SAND	PEAT	Similarity Number = 0
Secondary soil type	silt	silt	Similarity Number = 100

Final Similarity Number = 0.3 x 100 + 0.7 x 0 = 30, indicating a very low similarity as expected.

If the comparison is between a clayey PEAT and a sandy PEAT then the two main soils are identical and thus have a similarity of 100 and the lesser constituents (clayey, sandy) have a similarity of 97. Therefore the overall similarity is :

Final Similarity Number = 0.3 x 97 + 0.7 x 100 = 99, indicating very similar soils as expected.

4.4.2 Consistency

The descriptive terms for consistency depend on soil type. These can be directly related to a numerical measure of shear strength (for cohesive soils) or to a Standard Penetration Test N-value range (for granular soils), explicitly set out in the Code of Practice. A correlation between shear strength and N-value for clays, [73], was used in order to achieve a uniform representation scheme for consistency.

According to the descriptive term a range of N-values is defined and for each range a single value is identified for use in the program. These qualitative-quantitative linkages are presented in Tables 4.3a to 4.3c for sand, silt and clay respectively (the terminology given for silt is no longer widely used and the terms for granular or cohesive soils would now be applied depending on the nature of the silt). Subdivisions of the strength ranges for clay, which are widely used, have been adopted and are included in Table 4.3c.

Descriptive Term	N-value range	N-value used
Very Loose	0-4	2
Loose	4-10	7
Medium Dense	10-30	20
Dense	30-50	40
Very Dense	>50	52

Table 4.3a   Definition of Terms for Sand.

Descriptive Term	N-value range	N-value used
Loose	0-10	7
Moderately Compact	10-15	12
Compact	15-30	22
Very Compact	>30	40

Table 4.3b   Definition of Terms for Silt.

Descriptive Term	Undrained Shear Strength, Cu (kPa)	N-value range	N-value used
Very Soft	<20	0-2	1
Soft	20-40	2-4	3
Firm	40-75	4-8	6
Stiff	75-150	8-15	12
Very Stiff (Hard)	>150	>15	20
Soft to Firm	40-50	4-6	5
Firm	50-75	6-8	-
Firm to Stiff	75-100	8-10	9
Stiff	100-150	10-15	-

Table 4.3c   Definition of Terms for Clays.

The consistency of two soils can then be compared by calculating the mean difference in the N-values attached to the descriptive terms. The maximum difference in N values between the values assigned to each of the descriptive terms is 50. To convert the N value difference to a Similarity Number (with a range of 0-100%):

$$\text{Similarity Number} = 100 - 2 \times \text{N\_difference}$$

So, if one needs to compare a *stiff* clay (N = 12) with a *soft* clay (N = 3), the Similarity Number in terms of consistency will be calculated as follows:

$$\text{Similarity Number} = 100 - 2 \times (12 - 3) = 82.$$

4.4.3 Structure

Structure indicates the presence of bedding, discontinuities, or shearing within the soil. It plays a major role in the comparison between two soils because it is closely related to the behaviour of a soil. Structure is identified by the description of the feature, the spacing, dip and orientation, and details of the surface finish.

Spacing can be defined by descriptive terms which are linked with numerical values in British Standard 5930, [6]. The scale of bedding spacing and of other discontinuities is shown in Tables 4.4 and 4.5.

Descriptive Term	Mean Spacing (mm)
Very thickly bedded	over 2000
Thickly bedded	2000 to 600
Medium bedded	600 to 200
Thinly bedded	200 to 60
Very thinly bedded	60 to 20
Thickly laminated	20 to 6
Thinly laminated	under 6

Table 4.4　　　Scale of bedding spacing.

Descriptive Term	Mean Spacing (mm)
Very widely spaced	over 2000
Widely spaced	2000 to 600
Medium spaced	600 to 200
Closely spaced	200 to 60
Very closely spaced	60 to 20
Extremely closely spaced	under 20

Table 4.5　　　Scale of spacing of other discontinuities.

Since the descriptive terms for structure relate to a logarithmic scale of spacing a direct comparison in terms of physical spacing has not been adopted. Instead it has been taken that a change from the lowest spacing category to the highest category causes a change in Similarity Number of 100%. In order to define the change from one category to another the maximum difference of 100% is divided by the number of limits between the spacing categories (17% for bedding spacing, 20% for discontinuity spacing).

Therefore, a comparison of bedding spacing involving a *thinly laminated* soil and a *thickly laminated* soil (adjacent categories) gives a Similarity Number of 83% (100-

17%). A comparison of a soil with *extremely closely spaced* discontinuities with a soil having a *medium* discontinuity spacing gives a Similarity Number of 40% (100-60%).

4.4.4 Colour

As for soil type, it was found to be difficult to develop rules for directly comparing colour using descriptive terms. Colour can be represented by Main Colour, Secondary Colour and Modifier. An example is *'Dark yellowish brown'* which can be represented as:

Main Colour: Brown

Secondary Colour: Yellow

Modifier: Dark

Often one is comparing colours which have different names and yet are very similar colours. One would need to know that RED is similar to PINK but different from GREEN and so on.   Again this is further confused by secondary colours and modifiers.   It was found useful to represent colours by numerical values of *Hue, Saturation and Luminence*, [25].   Hue is the position of a colour along the colour spectrum.   Luminence is the brightness of a colour on a scale from black to white. Saturation is the purity of a colour moving from grey to the pure colour.   The Hue, Luminence and Saturation values for some colours are shown in Table 4.6, [40].   It is essential to note that Hue takes values from 0 to 240 on a circular scale, meaning that the values 0 and 240 are coincident.

Colour	Hue	Luminence	Saturation
Red	0	120	240
Pink	230	180	240
Purple	200	60	240
Yellow	40	120	240
Brown	20	60	240
Green	80	120	240
Blue	160	120	240
White	-	240	-
Grey	-	180	0
Black	-	0	-

Table 4.6   Hue, Luminence and Saturation values for several colours.

It was found to be possible to represent colour combinations as a value of Hue and a value of Luminence by combining the Hue and Luminence values for the Main colour and the Secondary colour. A weighting of 75:25 towards the Main colour was found to give realistic Hue and Luminence values for the combined colour. The only deviation from this rule was found to be necessary when white or black were present as secondary colours. In this case it was found to be more realistic to calculate the Luminence for the combined colour using a weighting of 50:50. The luminence values need to be decreased by 20% where the modifier *Dark* is used and increased by 10% where *Light* is used.

Saturation takes the same value for all colours (240) except for grey (which has a value of 0) and black or white (which can take any value). The Saturation for all colour combinations (except those including grey) was taken to be 240. When grey was involved a weighting of 70:30 towards the Main Colour was found to give a realistic value for Saturation.

In this way every colour description can be represented in terms of Hue, Luminence and Saturation. Colours can be compared by observing the numerical difference between their Hue values, their Luminence values and also between their Saturation

values. Since the scale for Hue is circular the maximum hue difference is 120. If the difference is calculated to be greater than 120 it must be subtracted from 240. The maximum differences for Luminence and Saturation are 240. The calculated differences in Hue, Luminence and Saturation are normalised by the maximum differences to fall in the range of 0-100%. The Similarity number for colour is then determined by combining the similarity (100-difference) for Hue, Luminence and Saturation. These are combined by weighting the three aspects 40:30:30 respectively.

$$\text{Simlarity Number} = 0.4 \times (100 - \frac{\text{Hue\_diff} \times 100}{120}) + 0.3 \times (100 - \frac{\text{Lum\_diff} \times 100}{240}) +$$
$$+ 0.3 \times (100 - \frac{\text{Sat\_diff} \times 100}{240})$$

For example the colour description 'Dark yellowish brown' can be represented numerically as follows.

Main colour: Brown		Secondary colour: Yellow		Yellowish Brown	
Hue:	20	Hue:	40	Hue:	25
Lum:	60	Lum:	120	Lum:	75
Satur:	240	Satur:	240	Satur:	240

So for 'Dark yellowish brown' the Luminence has to be decreased by 20% and it becomes 60.

For 'Reddish brown' the representation will be

Main colour: Brown   Secondary colour: Red   Reddish Brown

Hue:	20		Hue:	0		Hue:	15
Lum:	60		Lum:	120		Lum:	75
Satur:	240		Satur:	240		Satur:	240

The comparison between the two colours will give:

Hue_difference	= 25-15	= 10
Lum_difference	= 75-60	= 15
Sat_difference	= 240-240	= 0

This gives a Similarity Number of:

$$0.4 \text{ x } (100-10 \text{ x } 100/120) + .0.3 \text{ x } (100-15 \text{ x } 100/240) + 0.3 \text{ x } 100 = 95$$

4.4.5   Overall Similarity Number

In order to produce an overall Similarity Number which incorporates the individual Similarity Numbers, weighting factors had to be established for each parameter which reflect their relative importance. It is obvious that the Similarity Number for Soil type is more important than the Similarity Number for Colour and would therefore be assigned a higher weighting. The final weightings applied to each parameter were derived from a knowledge elicitation exercise in which a number of geotechnical experts were consulted. This exercise was carried out in the form of a questionnaire in which the experts were asked to identify the level of similarity for fifteen comparisons of soil descriptions.

The questionnaire (Appendix C) was designed so that individual comparisons incorporated different features. The initial comparisons (comparisons 1-3) included only soil type whereas the later ones incorporated other factors in addition to soil type and the last one involved layer structure. The other features were introduced in stages in the questionnaire so that the effect of each factor could be more easily identified in the analysis.

The questionnaire was sent to eight experts as a pilot study. The experts were asked to rate the comparisons using similarity ratings defined as:

A) Very similar

B) Similar

C) Slightly similar

D) Slightly dissimilar

E) Dissimilar

F) Very dissimilar

The results obtained are summarised as histograms in Figure 4.5. By observing how the experts rated the comparisons it was possible to establish ranges of Similarity Numbers for each of the six categories. A convenient scale emerged from the exercise, defined as follows:

A) 97-100   B) 94-97   C) 90-94   D) 85-90   E) 80-85   F) <80

In addition, the experts were asked to rate the four features (Soil type, Consistency, Structure and Colour) in order of importance using a scale:

A) Extremely important

B) Very important

C) Important

D) Unimportant

E) Trivial

Table 4.7 shows how the eight experts responded to this part of the exercise. It can be seen that most of the experts identified Soil type either as the most important feature or as important as Consistency, Structure or Colour. This information was made use of in developing the weighting factors.

Feature	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	Exp.6	Exp.7	Exp.8
Soil type	A	A	B	A	B	E	B	B
Consistency	B	B	C	A	B	E	B	C
Structure	B	B	C	C	B	D	C	C
Colour	B	D	C	D	C	C	E	B

Table 4.7   Level of importance attached to different features by experts.

A soil description may not always contain all the features identified above. It was found to be best to develop weighting factors according to the features participating in the comparison. The weighting factors arrived at using a trial and error method are shown in Table 4.8.

Soil type	Consistency	Structure	Colour
100	-	-	-
80	20	-	-
85	-	15	-
85	-	-	15
65	20	15	-
65	20	-	15
70	-	15	15
65	15	10	10

Table 4.8   Weighting factors applied for different combinations of features.

Table 4.8 indicates that if only Soil type is being compared then the Similarity Number for Soil type is weighted 100%. If Soil type and one other feature are involved then the weighting is reduced to 80:20 for Consistency and 85:15 for Structure or Colour, conforming with the order of importance attributed to these factors by the experts. Other combinations are shown in the table.

Comparison 1
Comparison 2
Comparison 3
Comparison 4
Comparison 5
Comparison 6
Comparison 7
Comparison 8
Comparison 9
Comparison 10
Comparison 11
Comparison 12
Comparison 13
Comparison 14
Comparison 15
Similarity Rating
Similarity Number

F   E   D   C   B   A

80   85   90   94   97   100

KEY
✗ : Calculated Similarity Numbers
▨ : Similarity ratings from the experts
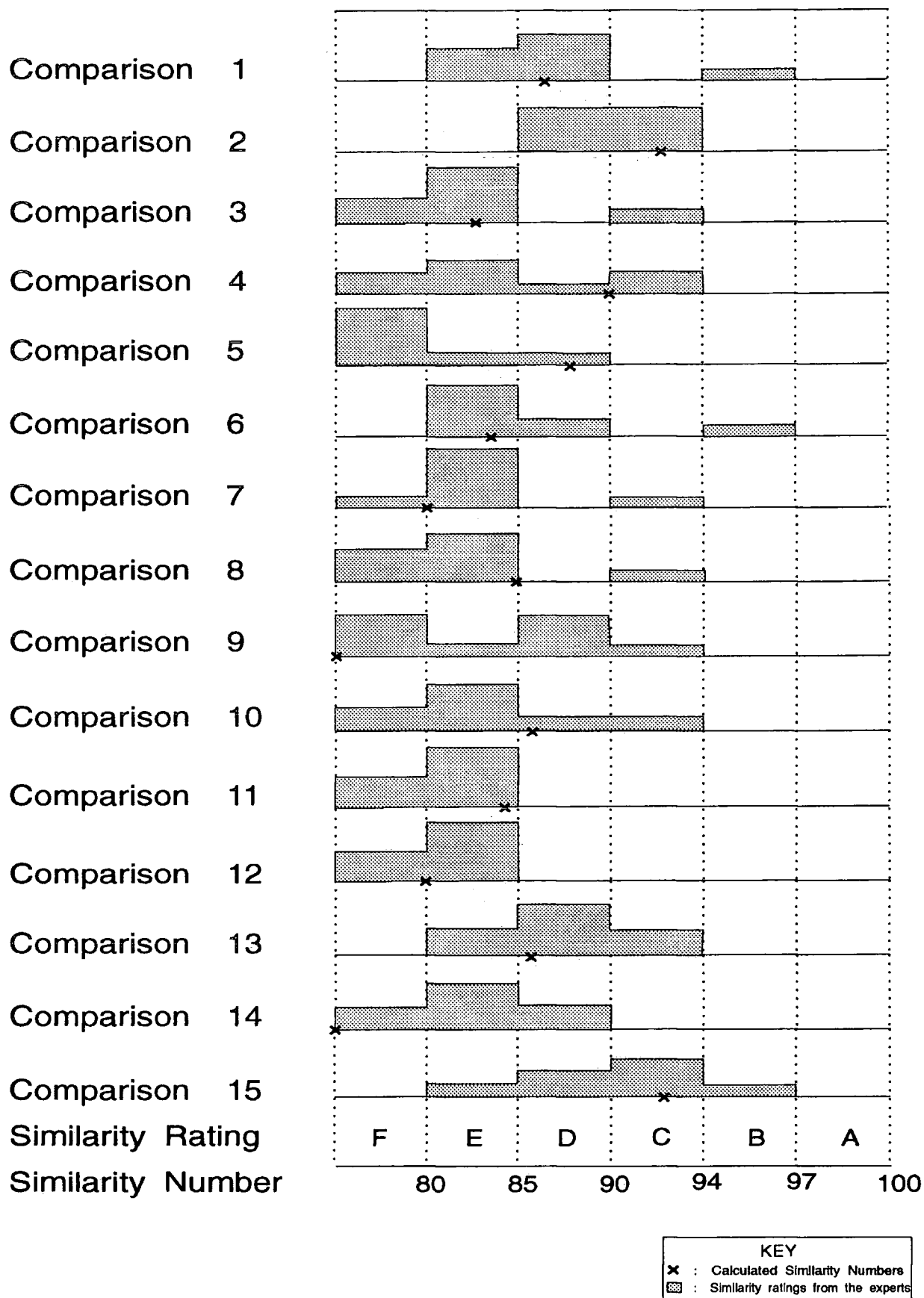
Figure 4.5   Results obtained from the questionnaire.

The Overall Similarity Numbers calculated using these weighting factors are shown in Fig. 4.5 for comparison with the similarity ratings given by the experts. In general, reasonable agreement is achieved with the mean rating for the eight experts. However, with such a small sample (a result of time constraints) it is difficult at this stage to be confident about the results, particularly when the ratings identified by the experts are not in perfect agreement. Since the pilot study has proved to be successful the questionnaire should be circulated to a larger pool of experts. The initial weighting factors should be modified, if necessary, when the full knowledge elicitation exercise has been completed.

4.4.6 Comparing Layers with Multiple Soils

So far only single soils within a layer were considered. When layers are made up of multiple soils then a method for comparing layers is required. In this case LAYER STRUCTURE defines the inter-relationships between the soils within the layers. The most common descriptive terms used in order to represent layer structure are *'interbedded with'*, *'interstratified with'*, *'with pockets of'*, *'with lenses of'* and *'with inclusions of'*. So the description "silty CLAY interbedded with gravelly SAND" means that the continuity of a soil layer of silty clay is disturbed by a relatively equal proportion of gravelly sand. On the other hand, the description "CLAY with pockets of SAND" indicates the presence of two distinct soils unequally distributed (clay is the dominant soil).

It is obvious that it becomes very difficult to directly compare such complex conditions. However, by breaking down such layer descriptions into discrete soils, i.e. *'silty Clay'* and *'gravelly SAND'*, a realistic representation of the layer description can be achieved. A Similarity Number can still be calculated between each of the soils using the method described above, and then combined using percentages of participation to give an Overall Similarity Number between layers. The percentages allocated are based on the layer structure term used. So for 'interbedded' and

77

'interstratified' the weighting is 50:50 and for 'pockets', 'lenses' and 'inclusions' it is 60:40 towards the dominant soil.

A general example is given below. The descriptions used in the example only include soil type.

'Soil A interbedded with Soil B' compared with *'Soil C with pockets of Soil D'*

The percentages of participation are 50% for soil A and soil B, 60% for soil C and 40% for soil D. When a Similarity Number is calculated for a pair of soils then this number is combined using the minimum of the percentages allocated to each one of these soils.

Similarity Number between A and C is S1   Percent=min(0.50, 0.60)=0.50

Similarity Number between A and D is S2   Percent=min(0.50, 0.40)=0.40

Similarity Number between B and C is S3   Percent=min(0.50, 0.60)=0.50

Similarity Number between B and D is S4   Percent=min(0.50, 0.40)=0.40

The final Similarity Number is then defined as follows:

Similarity Number = max { [(0.50xS1) + (0.40xS4)] , [(0.40xS2) + (0.50xS3)] }

For example, if the description *'GRAVEL interbedded with silty CLAY'* has to be compared with the description *'silty CLAY with pockets of gravelly SAND'* then:

S1 = 58   (*GRAVEL*   cf. *Silty CLAY*)

S2 = 85   (*GRAVEL*   cf. *Gravelly SAND*)

S3 = 100   (*Silty CLAY*   cf. *Silty CLAY*)

S4 = 73   (*Silty CLAY*   cf. *Gravelly SAND*)

Similarity Number = max { [(0.50x58) + (0.40x73)] , [(0.40x85) + (0.50x100)] }

= max {58,84} = 84

The above Similarity has a high value indicating two similar layers, as would be expected since one of the soils is identical in both layers.

If one compares *'CLAY with pockets of SAND'* and *'SAND with lenses of CLAY'* then:

S1 = 67   (CLAY cf. SAND)   Percent = min(0.60,0.60) = 0.60

S2 = 100   (CLAY cf. CLAY)   Percent = min(0.60,0.40) = 0.40

S3 = 100   (SAND cf. SAND)   Percent = min(0.40,0.60) = 0.40

S4 = 67   (SAND cf. CLAY)   Percent = min(0.40,0.40) = 0.40

The final Similarity Number will be:

Similarity Number = max { [(0.60x67) + (0.40x67)] , [(0.40x100) + (0.40x100)] }

= max {67,80} = 80

The actual similarity between clay and sand is 67. The above Similarity Number has a higher value because both descriptions contain pockets of soils that reinforce the overall similarity.

Finally, it must be mentioned that the percentages of participation used do not reflect the physical proportion of the soils. These numbers were selected so that when

combined with the individual similarities, which follow a non-physical scale (similarities less than 80 are considered low), they give sensible results.

4.4.7 Implementation

The parser module, that was described earlier, is able to process soil descriptions containing terms referring to Soil Type, Consistency, Structure, Colour, Plasticity, Texture, Particle Size Distribution, Grain Size and Shape. The recognised descriptive terms are stored in lists according to the soil characteristic that they indicate and these lists are passed for processing to the VASIC module (Appendix A). The *assign_values* clause performs the value assignment operation through the following predicates :

process_soil: Using the predicate *make_list* it breaks down the Soil_list into sublists that contain one or more soil types having the same amount (Main_list, Major_list, Sec_list, Minor_list) and then allocates the relative percentages to the participating soil types using the predicates *psd1* (for coarse grained soils) and *psd2* (for fine grained soils) in accordance with the pairs of Amount-list and Percent-list that were discussed under Soil Type, earlier in Section 4.4.1.

process_cons: Calls the appropriate fact-strings *cons*(Term, Value) and either allocates an SPT N value to consistency for a single descriptive term or it calculates the average N value between the upper and lower consistencies.

process_structure: Assigns a range of spacing values (in mm) to the descriptive terms by calling the fact-strings *str_spacing*(Term, Range, Percent) that contain the relative numbers.

process_col: It makes use of the predicate *calc_sat_lum_hue* in order to allocate the Hue, Saturation and Luminence values to the colour names and to combine these numbers for colour combinations according with the procedure described previously. In addition, *process_col* modifies the luminence value if required (for dark or light) using the predicate *mod_lum*.

process_other: It carries any additional information about the soil to the module without assigning any values to the descriptive terms so that if the user requires the assignment of values he or she will be informed about qualitative information that has not been converted into numbers.

When comparison between a pair of soil descriptions is considered, the *compare* clause performs the individual numerical comparisons using the following predicates :

compare_soil: Calculates the Similarity Number for soil type based on the numerical comparison of the notional particle size distributions which are defined from the Percent-list and Amount-list. When multiple soils exist within a layer, performs the required calculations using the methodology described under Section 4.4.6.

compare_cons: Calculates the Similarity Number for consistency using the expression identified under Section 4.4.2

compare_bed: Calls the fact-string str_spacing(Term, Range, Percent) and uses the Percent value to calculate the change in percent from one spacing category to the other for bedding spacing.

compare_spac: Performs the same calculations, as compare_bed, for spacing of other discontinuities.

compare_col: Calculates the difference in Hue, Luminence and Saturation and then combines these values to identify the colour difference following the procedure that was described under Section 4.4.4.

overall_sim: Performs the calculation of the Overall Similarity Number combining the individual similarities according to Table 8 in Section 4.4.5.

4.4.8 Discussion

In addition to Soil Type, Consistency, Structure and Colour there are other factors that may also influence the comparison between two soil descriptions, such as plasticity, texture etc. The parser which has been developed for processing the soil descriptions can recognise the descriptive terminology for these additional factors but these terms are not converted into quantitative parameters which will allow a numerical form of comparison. As the system is envisaged in its complete form, all these factors should be numerically processed in order to achieve a more complete estimate of the soils' similarity.

The establishment of weighting factors for the calculation of the Overall Similarity Number is another area where improvement is required. The weighting factors that are currently being used by the system are based on a questionnaire answered by eight geotechnical experts. A complete knowledge acquisition exercise with a larger sample of experts' opinions, based on the designed questionnaire, would provide more information about the values of these weighting factors.

CHAPTER 5

A METHODOLOGY FOR INTERPRETING LAYERING FROM BOREHOLE INFORMATION

5.1 General

A major aspect of the interpretation of the geotechnical information recovered from a site investigation is the identification of the ground conditions across the site, based on observations obtained at discrete points such as boreholes. The spacing between boreholes can vary from a few metres to kilometres, depending on the type and scale of the investigation. The geological conditions can be highly variable, with soil or rock layers changing in character, depth or thickness between boreholes or dying out completely, [19, 67, 68].

The correlation process is approached at both site-wide and borehole-to-borehole levels. At the site-wide level an attempt is made to identify marker beds, which are layers that 'stand out' from the general ground conditions and can thus be more easily traced across the site.

A search for marker beds is first made at each borehole. Continuity of marker beds is then investigated between pairs of adjacent boreholes. The investigation is then extended to groups of three boreholes (triangles) and finally over the whole site. Continuous marker beds identified in this way are used to construct an initial site-wide model of the ground conditions. At the same time, areas that appear to deviate from the overall trends are highlighted for further investigation. Finally, the detailed

ground conditions are examined on a borehole-to-borehole basis. At this level a set of hypotheses about the ground conditions is constructed by looking at pairs of adjacent boreholes. The site-wide model is used to constrain the number of possible hypotheses at the detailed borehole-to-borehole level.

Current practice in site investigation indicates that there is not a standard way for solving problems involving correlation of borehole information. The methodology proposed is to assist the geotechnical engineer in the interpretation of the ground conditions and to enable him/her to identify areas within a site where additional information is required.

The methodology is described in a way suitable for implementation in PDC Prolog, although full implementation has only been carried out for the borehole-to-borehole interpretation; not the whole methodology.

2.   Overview of the Methodology

A modular approach to the correlation process was adopted in order to simplify the solution of a highly complex problem and take advantage of modular structures that allow modifications to be more easily included. The interpretation of the ground conditions is achieved by examining the continuity of soil layers in terms of soil type. The methodology can similarly be applied for correlation of layers in terms of colour and consistency which are also very important soil features. The proposed structured approach consists of the following steps:

- o   Identification of Possible Marker Beds
- o   Configuration of Triangles
- o   Connection between Pairs of Marker Beds
- o   Assessment of Planar Marker Beds
- o   Calculation of Dip Angle and Dip Orientation

84

o   Continuity of Marker Beds Across the Site

o   Borehole-to-Borehole Correlation

In parallel with the methodology's description, a practical application is considered to serve as a case study. The application involves the characterization of a site near Mainsforth in County Durham. The site investigation was carried out by Northumbrian Water Authority, [48], and consisted of ten boreholes. The borehole arrangement at the site is shown in Figure 5.1, whereas the coordinates of the boreholes according to the National Grid Reference are shown in Table 5.1. The detailed descriptions of the ground conditions, included in the site investigation report, were simplified by considering only soil type. The simplified descriptions of the soil layers for each one of the ten boreholes, together with the elevation of each layer are shown in Figures 5.2a and 5.2b.

Borehole Number	East x-axis in metres	North y-axis in metres
1	431,162	530,687
2	431,500	530,550
3	431,724	530,413
4	432,165	530,458
5	432,533	530,353
6	432,718	530,306
7	432,950	530,060
8	432,615	530,130
9	431,254	530,487
10	431,985	530,180

Table 5.1   Borehole coordinates according with the National Grid Reference.

Figure 5.1   Boreholes' arrangement at Mainsforth in County Durham.
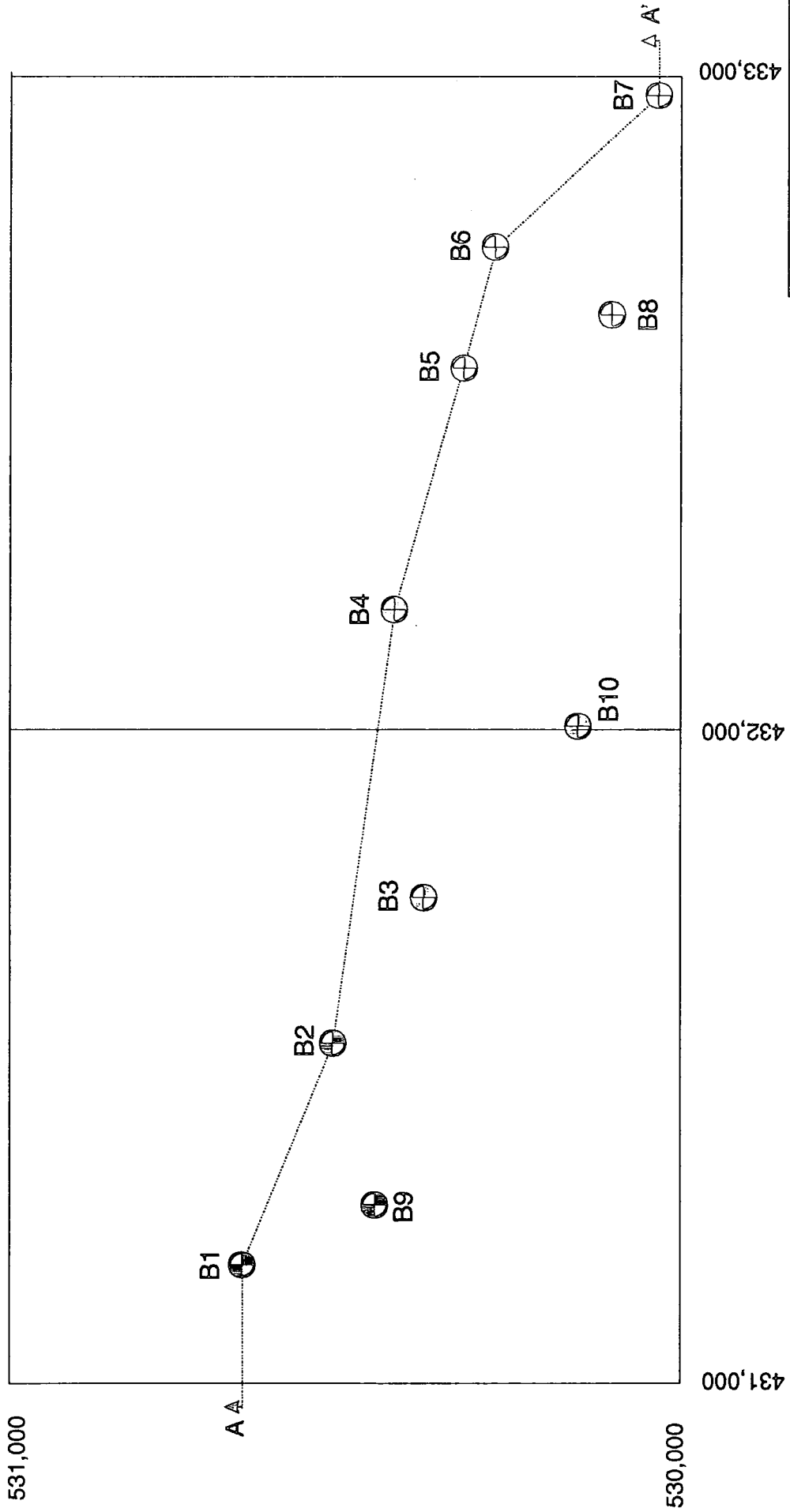
KEY
Northings and Eastings in meters.

BOREHOLE 1

Level	Description
76.43	PEAT
72.83	Very silty slightly sandy CLAY
71.63	Silty SAND with a little gravel
69.33	Sandy CLAY
65.43	Silty sandy CLAY
63.43	Very silty sandy CLAY
61.33	

BOREHOLE 2

Level	Description
76.76	PEAT
74.11	Very silty slightly sandy CLAY
72.50	Silty CLAY
70.65	SAND and GRAVEL
68.96	Sandy CLAY
67.76	Silty sandy CLAY
65.00	

BOREHOLE 3

Level	Description
76.38	PEAT
73.68	Clayey sandy SILT
72.88	Silty sandy CLAY
71.68	SAND and GRAVEL
69.08	

BOREHOLE 4

Level	Description
76.80	PEAT
73.70	Silty sandy CLAY
72.80	V. silty sandy CLAY
72.20	Very silty slightly sandy CLAY
71.25	Silty SAND with some gravel
66.90	

BOREHOLE 5

Level	Description
76.17	PEAT
74.17	V. silty sandy CLAY
73.27	Very silty slightly sandy CLAY
71.17	Silty very sandy CLAY
69.77	Clayey very sandy GRAVEL
63.67	Slightly silty sandy CLAY
53.23	Slightly gravelly SAND
50.00	Very silty sandy CLAY
48.57	

NOTES

1. Highlighting indicates potential marker beds
2. Numbers are reduced level (mOD)

Figure 5.2a   Profile of boreholes 1 to 5.

BOREHOLE 10

PEAT	76.63
	73.83
Very sandy CLAY	72.30
SAND and GRAVEL	70.20

BOREHOLE 9

PEAT	77.03
	73.50
Very sandy CLAY	72.48
SAND and GRAVEL	69.36

BOREHOLE 8

PEAT	76.24
V.silty s.sandy CLAY	74.34
	73.84
Slightly silty SAND with a trace of gravel	70.50
V.silty s.sandy CLAY	69.10
V. sandy silty CLAY	68.35
Sandy CLAY	66.50
Sandy GRAVEL	63.35
S. silty sandy CLAY	62.50
Very silty sandy CLAY	61.40
Very silty slightly sandy CLAY	60.25

BOREHOLE 7

PEAT	76.42
Very silty slightly sandy CLAY	74.97
Silty SAND	73.42
	70.62
Very silty slightly sandy CLAY	67.02
GRAVEL with some sand	62.08

BOREHOLE 6

PEAT	76.31
V.silty s.sandy CLAY	75.00
SAND with a trace of gravel	74.66
	72.46
V.silty s.sandy CLAY	72.01
GRAVEL with a little sand	69.61
Very sandy silty CLAY	68.00
V. silty sandy CLAY	67.10
V.silty s.sandy CLAY	66.35

NOTES
1. Highlighting indicates potential marker beds
2. Numbers are reduced level (mOD)

Figure 5.2b   Profiles of boreholes 6 to 10

88

5.2.1 Identification of Possible Marker Beds

(i)   Procedure

A search for possible marker beds is made in each borehole. These can be identified if soil type, colour or consistency differ significantly from the majority of the layers in the borehole. In this work, the assessment of marker beds is achieved with respect to soil type, but a brief description of the means to identify marker beds with respect to colour and consistency is also given below.

a.   Soil Type

In the previous Chapter, every soil description was quantitatively represented in terms of a notional particle size distribution. Using that representation scheme, an area can be calculated for every distribution curve by summing the observed percentages passing of the material grains at the six points that define the limits between the different soil types (Clay, Silt, Sand, Gravel, Cobbles, Boulders). Thus, a single number which is indicative of the participating soil types can be produced for every distribution. The number is called Area Identifier Number (AIN) and can be calculated using the following expression:

$$\text{Area Identifier Number} = \sum_{i=1}^{n} (\text{Percent Passing}) \quad \text{where n=6.}$$

The AIN can vary from 600 when a CLAY is under consideration to 100 when only BOULDERS are encountered. Some examples of typical particle size distributions together with the corresponding AINs are shown in Figure 5.3.

89
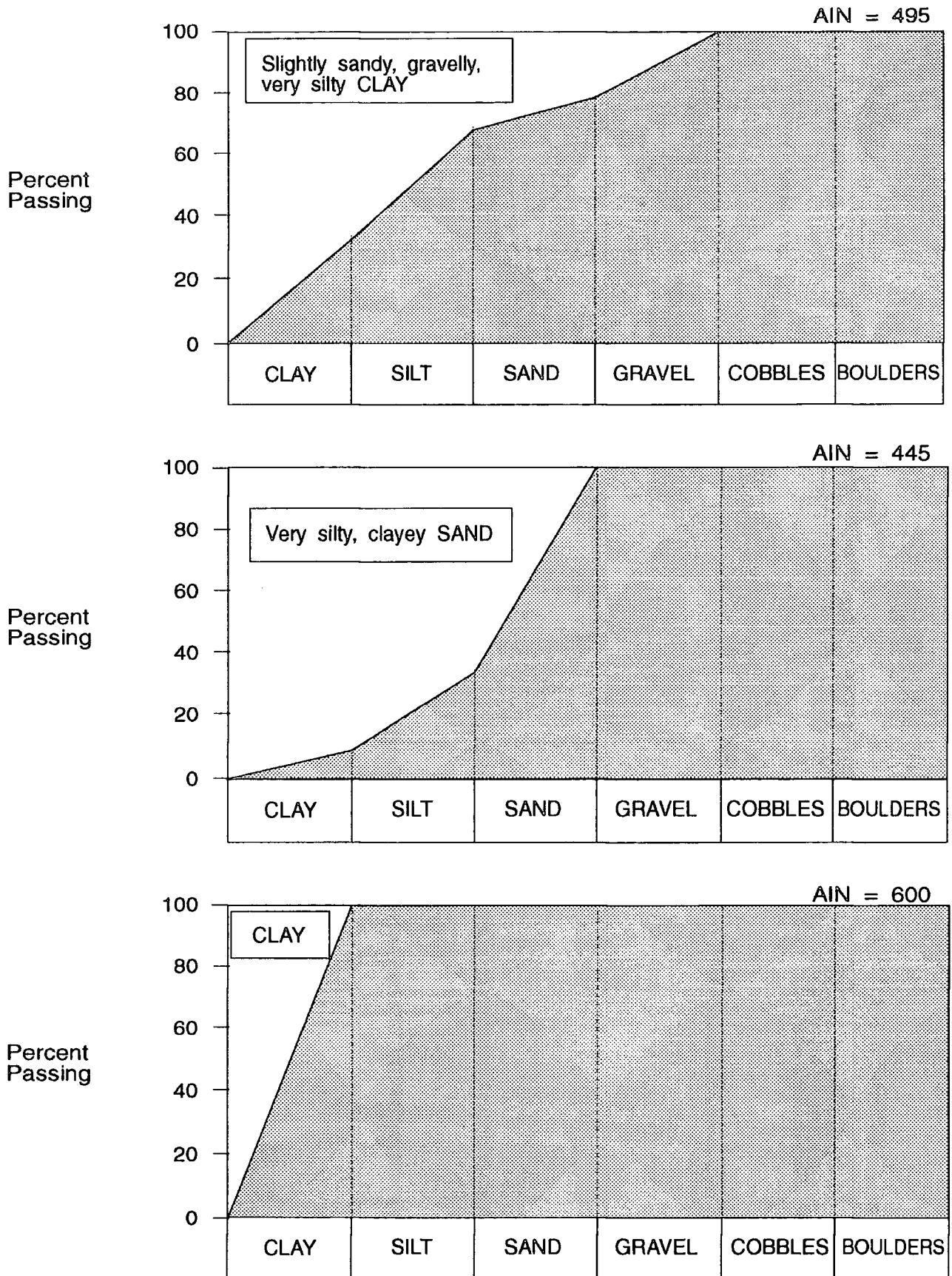
Figure 5.3   Examples of notional particle size distributions, together with their corresponding Area Identifier Numbers.

Therefore, for every soil layer within a borehole an AIN is calculated, except PEAT as this has variable grain size. Further, trends about marker beds can be numerically identified by calculating the mean ($\bar{x}$) and standard deviation (s) of these numbers.

If n layers have been recorded in a borehole and their Area Identifier Numbers are N_1, N_2, N_3, ..., N_n, then the unbiased estimation of the weighted arithmetic mean ($\bar{N}$) can be calculated as follows:

$$\bar{N} = \frac{1}{n}\sum_{i=1}^{n} N_i$$

while the estimation of the corrected standard deviation of the population is:

$$s = \sqrt{\frac{\sum_{i=1}^{n}\left(N_i - \bar{N}\right)^2}{n-1}}$$

The reason that *n-1* appears instead of the *n* denominator in the above expression is that the mean is also an estimated value, so the number of degrees of freedom has to be decreased by one in accordance with Bessel's correction, [56].

Then, by observing layers where the AIN (N_i) is more than one standard deviation from the arithmetic mean, i.e. :

$$N_i > \bar{N} + s \quad \text{or} \quad N_i < \bar{N} - s$$

these can be seen as possible marker beds because they 'stand out' from the rest of the layers in the borehole since their AINs are extreme values of the borehole's layer population.

When PEAT is considered amongst other inorganic soil layers, it is treated as another possible marker bed without participating in the numerical identification of the inorganic ones.

The layers identified as possible marker beds within each borehole can then be stored as a PDC-Prolog fact-list of the form:

pos_mar_bed(Bor_no, [L_i, L_j,.....])

where L_i and L_j represent the layers numbered i and j within borehole numbered Bor_no. In the case of boreholes with no possible marker beds identified, the list containing the layer numbers is empty.

b.   Consistency

The descriptive terms for consistency correspond to a Standard Penetration Test N-value, following the uniform representation scheme which was described in the previous chapter. Using the same numerical estimation as above for soil type, the extreme values of the N-values population can be determined, as those that fall more than one standard deviation from the mean, and are considered as indicative of possible marker beds.

c.   Colour

According with the representation scheme used in the previous chapter, colour can be quantitatively described as a set of three values corresponding to Hue, Saturation and Luminence. By observing the extreme values of the Hue and Luminence values population, initial trends about possible marker beds can be made with respect to colour (Saturation is not considered because most colours have the same value, 240). Although for Luminence values the same method as for soil type and consistency can be applied, for Hue values care needs to be taken because of the circular scale that these values follow.

By determining the sum of differences between each Hue value and all the others within a borehole a number can be produced, called Total Hue Difference (THD), indicating the total numerical difference in Hue of one colour with all the rest. In the

Hue scale the value 120 represents the maximum difference and hence before summing differences one must ensure that this constraint is satisfied.

Thus, by calculating the mean and standard deviation of the THDs, one can observe the ones that fall more than one standard deviation from the arithmetic mean. Therefore, every Hue value that is defined from such a THD is considered to be indicative of a possible marker bed.

For example, for a borehole consisting of six layers the colour descriptions and their corresponding Hue and Luminence values are shown in Table 5.2 :

Layer	Colour Description	Hue	Luminence
1	Yellowish brown	25	75
2	Brown	20	60
3	Brown	20	60
4	Blue	160	120
5	Red	0	120
6	Reddish brown	15	75

Table 5.2   Colour descriptions in a borehole consisting of six layers.

For the Luminence values above the arithmetic mean is 85 and the corrected standard deviation is 36, thus all values are included in the range (85-36 = 49) up to (85+36 = 121) and no trend about possible marker beds can be made in terms of Luminence.

Then, the sum of differences (THD) between each Hue value and the rest is calculated.

For the first Hue value (25) the differences are:

Difference between 1 and 2 :   25 - 20 = 5

Difference between 1 and 3 :   25 - 20 = 5

Difference between 1 and 4 :   160 - 25 = 135 > 120 so it becomes

                                      240 - 135 = 105

Difference between 1 and 5 :   25 - 0 = 25

Difference between 1 and 6 :   25 - 15 = 10

Hence the first THD is (5+5+105+25+10) = 150.

Using the same procedure the THDs for all the colours can be calculated and are
shown below:

THD_1 =150, THD_2 =130, THD_3 =130, THD_4 =480, THD_5 =160 and THD_6 =130.

For the above THDs the arithmetic mean is 197 and the standard deviation is 140.

Therefore the accepted range of values is between (197-140)=57 and (197+140)=337.

So the THD of 480 is falling outside the accepted range and hence, the layer featuring

the Hue value of 160 is highlighted as a possible marker bed.

(ii)   Application

In the example, the possible marker beds in terms of soil type, which are presented as
highlighted layers in Figures 5.2a and 5.2b, can be stored by the following set of fact-
lists.

pos_mar_bed(1, [1,3]).
pos_mar_bed(2, [1,4]).
pos_mar_bed(3, [1,4]).
pos_mar_bed(4, [1,5]).
pos_mar_bed(5, [1,5,7]).
pos_mar_bed(6, [1,3,5]).
pos_mar_bed(7, [1]).
pos_mar_bed(8, [1,3,7]).
pos_mar_bed(9, [1]).
pos_mar_bed(10, [1]).

An example calculation for the possible marker beds of borehole 1 is given below.

The AINs for the layers of borehole 1 are:

AIN_2 = 515, AIN_3 = 400, AIN_4 = 500, AIN_5 = 520, AIN_6 = 510. Layer 1 is not

included as it is Peat. Therefore the arithmetic mean is 489 and the standard

deviation is 50. Hence the accepted range of values is between (489-50)=439 and

(489+50)=539. So the value 400 in the third layer of borehole 1 is the only one which falls outside the range. This layer is highlighted as a possible marker bed and together with the Peat layer make up the first of the fact-lists given above:

$$pos\_mar\_bed(1, [1,3]).$$

5.2.2 Configuration of Triangles

(i) Procedure

Following the identification of the possible marker beds, the site is divided into triangles having as vertices the borehole locations.

An automated method is proposed for that purpose based on automatic mesh generation, [23]. Its primary objective is to interconnect the borehole locations to form triangular elements in such a way that no elements overlap and all boreholes participate in one or more triangles. The procedure starts by selecting the node (borehole location) having the lowest x coordinate, assuming that the x-axis has an East-West direction (e.g. Borehole i). Next the nearest node is found (e.g. Borehole j) and the side ij is established. Then, a third node (e.g. Borehole k) is searched for, such that the angle $i\hat{k}j$ is maximum and ijk is an anticlockwise sequence of nodes. If no such node can be found, the next nearest node to i is found and the procedure is repeated.

To determine the largest angle $i\hat{k}j$, for angles in the range of 0-180°, it is sufficient to find which angle has the smallest cosine. This can be calculated using the following expression:

$$\cos\left(i\hat{k}j\right) = \frac{(ik)^2 + (jk)^2 - (ij)^2}{(ik).(jk)}$$

where (ik) indicates the distance between boreholes i and k.

95

The anticlockwise sequence of the boreholes i, j and k with coordinates (x_i,y_i), (x_j,y_j) and (x_k,y_k) respectively can be ensured by confirming that:

$$\begin{vmatrix} x_i & y_i & 1 \\ x_j & y_j & 1 \\ x_k & y_k & 1 \end{vmatrix} > 0$$

and is used in order to disallow overlapping triangles.

After configuring the first triangle (ijk), side ik is established and a new triangle is looked for, following the two constraints that were described above. When all possible triangles containing borehole i have been identified, borehole j is to be fully or partially surrounded by triangular elements. When all boreholes have been examined, a set of different and non-overlapping triangles form the created triangular configuration of the site.

Next, the triangular elements are checked in terms of their quality index. According to Lindholm, [35], every triangle with side lengths ik, jk and ij features a quality index (q) that can be calculated as follows:

$$q = \frac{8.[s-(ik)].[s-(jk)].[s-(ij)]}{(ik).(jk).(ij)} \ , \ \text{where} \ \ s = \frac{(ik)+(jk)+(ij)}{2}$$

and hence q is the ratio of the diameter of the inscribing circle to the radius of the circumscribing circle. Thus, for an equilateral triangle q=1, whereas a triangle which tends to a straight line has q=0.

Considering that correlation of boreholes inside triangles of poor geometry may cause misinterpretation of the ground conditions, such triangles that are defined as having a value of quality index less than 0.05 are highlighted when found. If, after excluding the 'poor' triangle from the final site configuration the boreholes are still all

interconnected through the remaining triangles, then the 'poor' triangular element is discarded. Otherwise, the 'poor triangle' is maintained in the final triangular configuration but any inference based on that part of the site should be treated with care.

After completing the triangular configuration of the site, the resulting triangular elements can be stored as a set of fact-lists having the form:

triangle(Triangle_no, Qual_index, [Bor_i, Bor_j, Bor_k]).

(ii)   Application

Following the procedure that was described above the triangular configuration of the site can be represented through the set of fact-lists:

triangle(1,0.706,[1,9,2]).
triangle(2,0.027,[1,2,4]).
triangle(3,0.140,[9,3,2]).
triangle(4,0.150,[9,10,3]).
triangle(5,0.069,[2,3,4]).
triangle(6,0.908,[4,3,10]).
triangle(7,0.625,[4,10,5]).
triangle(8,0.007,[4,5,6]).
triangle(9,0.008,[10,7,8]).
triangle(10,0.598,[10,8,5]).
triangle(11,0.956,[5,8,6]).
triangle(12,0.840,[6,8,7]).

It can be observed that triangles 2, 8 and 9 have a low quality index (less than 0.05) and are thus considered triangles of poor geometry. At the same time, if these triangles are discarded, the remaining ones include all the boreholes on the site.

As a result these three triangles are dropped from the final configuration and this is shown in Figure 5.4, where the dotted lines represent sides of the triangles which were discarded.
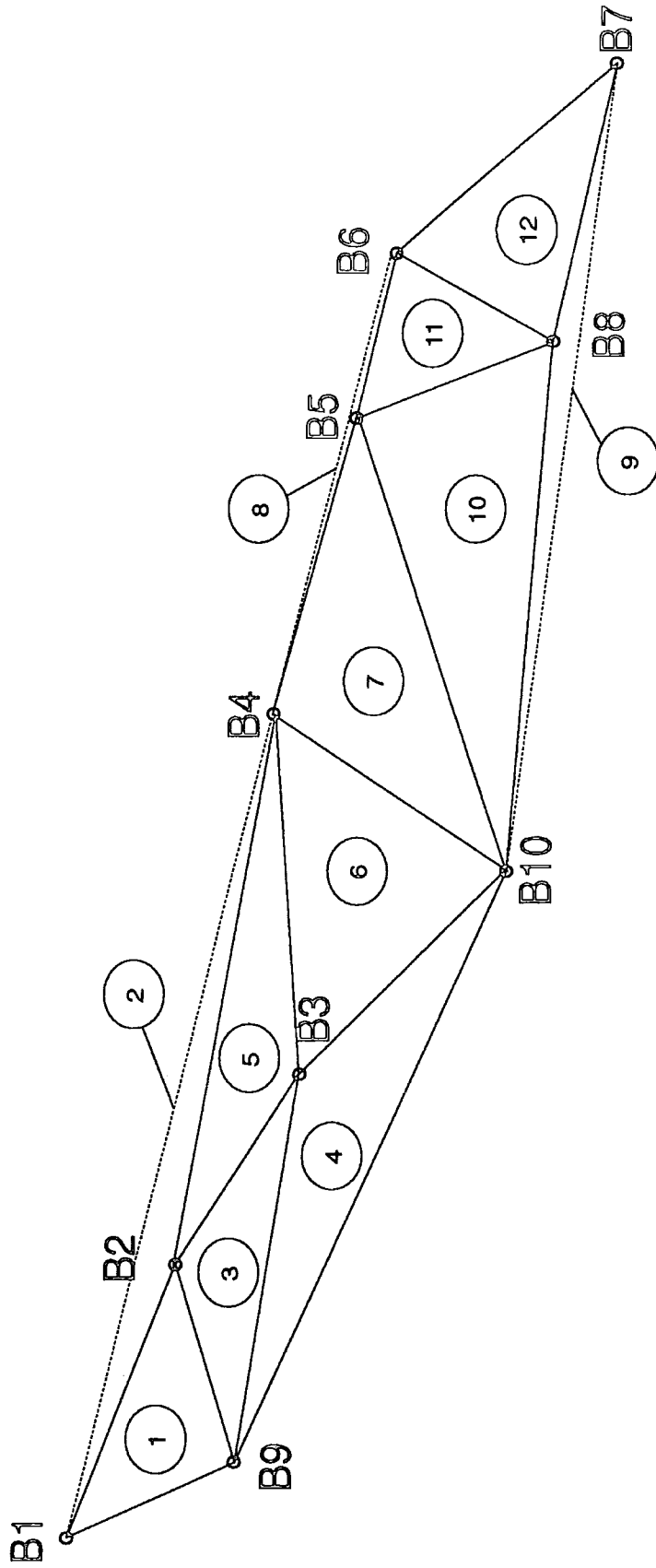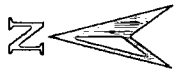
Figure 5.4   Configuration and numbering of triangles.

5.2.3   Connection between Pairs of Marker Beds

(i)   Procedure

During this step, the continuity of the possible marker beds is examined between pairs of boreholes that belong to the same triangle.

Assuming that ijk is an anticlockwise sequence of boreholes which are vertices of the same triangle, the continuity of the possible marker beds between boreholes i and j, j and k, k and i is assessed.

First the fact-lists *pos_mar_bed* for these boreholes are called and comparison takes place between these layers, using the procedure for the calculation of similarity that was described in the previous chapter. The continuity is established for the layers that appear to have Similarity Numbers greater than 90. Every time that such a number is observed a fact-list is created having the form:

$$connect(\text{Triangle\_no, [Bor\_i, } L_i, \text{ Bor\_j, } L_j])$$

where Bor_i and Bor_j are the pair of boreholes under examination and L_i and L_j are the corresponding layers of these boreholes between which a Similarity Number greater than 90 is calculated.

It must be noted that, if at least one of the three boreholes within a triangle has one or more marker beds identified, which do not link with marker beds (if any) in the other two, the two boreholes are searched for layers that are similar to the possible marker bed (Similarity Number > 90). If such layers are found, these are stored as possible marker beds and their continuity is examined. Hence, using this methodology, layers within a borehole which were not identified as possible marker beds (either because there are few layers in the borehole or there is an almost equal distribution of differing layers and thus the calculation of the arithmetic mean and the standard deviation can not provide for layers that 'stand out' from the general ground conditions in that borehole) are eventually observed using the information obtained from neighbouring boreholes. If however, in all three boreholes no possible marker

beds were identified previously (meaning that for these boreholes the *pos_mar_bed* fact-lists are empty) then it becomes evident that no conclusion can be reached concerning marker beds for this triangle. The triangle is left for processing until a later stage, when the marker beds present in the neighbouring triangles have been established.

(ii)   Application

The continuous marker beds between pairs of boreholes that belong to the same triangle are given below together with the possible marker beds, if any, that were not identified in the earlier section:

- ○   Triangle 1

  connect(1,[1,1,9,1]).
  connect(1,[9,1,2,1]).
  connect(1,[2,1,1,1]).
  connect(1,[1,3,9,3]) and *pos_mar_bed*(9,[3]).
  connect(1,[9,3,2,4]).
  connect(1,[2,4,1,3]).

- ○   Triangle 3

  connect(3,[9,1,3,1]).
  connect(3,[3,1,2,1]).
  connect(3,[2,1,9,1]).
  connect(3,[9,3,3,4]).
  connect(3,[3,4,2,4]).
  connect(3,[2,4,9,3]).

- ○   Triangle 4

  connect(4,[9,1,10,1]).
  connect(4,[10,1,3,1]).
  connect(4,[3,1,9,1]).
  connect(4,[9,3,10,3]) and *pos_mar_bed*(10,[3]).
  connect(4,[10,3,3,4]).
  connect(4,[3,4,9,3]).

- ○   Triangle 5

  connect(5,[2,1,3,1]).
  connect(5,[3,1,4,1]).

connect(5,[4,1,2,1]).

connect(5,[2,4,3,4]).

connect(5,[3,4,4,5]).

connect(5,[4,5,2,4]).

o   Triangle 6

connect(6,[4,1,3,1]).

connect(6,[3,1,10,1]).

connect(6,[10,1,4,1)].

connect(6,[4,5,3,4]).

connect(6,[3,4,10,3]).

connect(6,[10,3,4,5)].

o   Triangle 7

connect(7,[4,1,10,1]).

connect(7,[10,1,5,1]).

connect(7,[5,1,4,1]).

connect(7,[4,5,10,3]).

connect(7,[10,3,5,5]).

connect(7,[10,3,5,7]).

connect(7,[5,5,4,5]).

connect(7,[5,7,4,5]).

o   Triangle 10

connect(10,[10,1,8,1]).

connect(10,[8,1,5,1]).

connect(10,[5,1,10,1]).

connect(10,[10,3,8,3]).

connect(10,[10,3,8,7]).

connect(10,[8,3,5,7]).

connect(10,[8,7,5,5]).

connect(10,[5,5,10,3]).

connect(10,[5,7,10,3)].

⊕   Triangle 11

connect(11,[5,1,8,1]).

connect(11,[8,1,6,1]).

connect(11,[6,1,5,1]).

connect(11,[5,5,8,7]).

connect(11,[5,7,8,3]).

connect(11,[8,3,6,3]).

connect(11,[8,7,6,5]).

connect(11,[6,3,5,7]).

connect(11,[6,5,5,7]).

o   Triangle 12

connect(12,[6,1,8,1]).

connect(12,[8,1,7,1])

connect(12,[7,1,6,1]).

connect(12,[6,3,8,3]).

connect(12,[6,5,8,7]).

connect(12,[8,3,7,3]) and *pos_mar_bed*(7,[3]).

connect(12,[8,7,7,5]) and *pos_mar_bed*(7,[5]).

connect(12,[7,3,6,3]).

connect(12,[7,5,6,5]).

In the example, the first three fact-lists for every triangle refer to PEAT. The possible marker beds identified in boreholes 7, 9 and 10 could not be recognised earlier because in these boreholes there was an almost equal distribution of fine and coarse grained soil layers and thus it was not possible to highlight the coarse grained soils as extreme values of the layers' population.

5.2.4   Assessment of Planar Marker Beds

(i)   *Procedure*

In the previous step, the continuity of marker beds between pairs of boreholes was established and stored as *connect* fact-lists. The main objective of assessing planar marker beds is the expansion of these fact-lists by considering continuity of marker beds within triangles defined by three boreholes, in accordance with the triangular site configuration

.

Each triangle is searched for planar marker beds by observing whether the layer numbers contained in the *connect* fact-lists form a 'closed loop'. Assuming that the first *connect* fact-list of triangle *ijk* indicates that layer L_i of borehole B_i and layer L_j of borehole B_j correlate, a search is made in the remaining fact-lists of that triangle to assess whether layer L_j of borehole B_j correlates with layer L_k of borehole B_k. If

102

that is so, the procedure is iterated once more in order to establish that layer L_k of borehole B_k correlates back to layer L_i of borehole B_i, thus closing the loop. In that case, layers L_i, L_j and L_k are considered observations of the same layer at three discrete points and it is established that within triangle *ijk* a continuous bed exists.

A similar technique for handling correlation problems is used in SCHIKORRE, [5], where continuity of coal seams is considered by performing local correlation of seams (between pairs of boreholes) and combining them in order to achieve a global correlation inside polygons that are defined by sets of four boreholes.

When assessing the existence of a marker bed inside triangle *ijk*, the planar bed can be stored as a fact-list of the form:

$$closed\_loop(\text{Triangle\_no},[L_i,L_j,L_k])$$

where the layers present in the fact-list correspond to the three boreholes that define the triangle that was stored earlier as a *triangle* fact-list. Following this procedure, a set of *closed_loop* fact-lists can be created, identifying planar marker beds within every triangle for which this is possible. If more than one marker bed exist inside the same triangle, a check of compatibility between them is performed. At this moment, the only constraint that can be applied is relative to whether the two beds cross-over. Therefore, if a second marker bed is observed in triangle *ijk* and it is stored as:

$$closed\_loop(\text{Triangle\_no},[L_{ii},L_{jj},L_{kk}])$$

the limitation that must be satisfied for the pair of marker beds to be compatible is:

$$L_i > L_{ii} \text{ and } L_j > L_{jj} \text{ and } L_k > L_{kk}$$

$$or$$

$$L_i < L_{ii} \text{ and } L_j < L_{jj} \text{ and } L_k < L_{kk}$$

If this limitation is satisfied, the two planar marker beds can be combined and stored as one fact-list of the form:

$$closed\_loop(\text{Triangle\_no},[L_i,L_j,L_k,L_{ii},L_{jj},L_{kk}]).$$

Hence, after completing this step, a set of planar marker beds is identified and if more than one appears to exist within the same triangle, an initial check is made to ensure that the two beds do not cross-over.

(ii)   *Application*

The planar marker beds, identified in the site under consideration, are numerically expressed below for every triangle.

- o   Triangle 1
        closed_loop(1,[1,1,1,3,3,4]).
- o   Triangle 3
        closed_loop(3,[1,1,1,3,4,4]).
- o   Triangle 4
        closed_loop(4,[1,1,1,3,3,4]).
- o   Triangle 5
        closed_loop(5,[1,1,1,4,4,5]).
- o   Triangle 6
        closed_loop(6,[1,1,1,5,4,3]).
- o   Triangle 7
        closed_loop(7,[1,1,1,5,3,5])
        closed_loop(7,[1,1,1,5,3,7]).
- o   Triangle 10
        closed_loop(10,[1,1,1,3,3,7])
        closed_loop(10,[1,1,1,3,8,5]).
- o   Triangle 11
        closed_loop(11,[1,1,1,5,7,5])
        closed_loop(11,[1,1,1,7,3,3]).
- o   Triangle 12
        closed_loop(12,[1,1,1,3,3,3,5,7,5])

which indicates the presence of three compatible marker beds within triangle 12.

5.2.5 Calculation of Dip Angle and Dip Orientation

(i) Procedure

The next step after defining the planar marker beds is to determine the attitude of the bed, meaning its dip angle and dip orientation. Having as input data the elevations of the top of each marker bed and the coordinates of the ten boreholes, the attitude of the bed can be determined provided that the sets of three boreholes are not on a straight line (this was ensured earlier by calculating the quality index of every triangle) and that the top of the bed is assumed planar (applicable for layers that are mainly sedimentary).

The dip angle and dip orientation can be determined using the 'three-point method', [4, 37], and is based on a linear combination of the elevations. The dip angle can be calculated from two points along the dip direction of the plane using the expression:

$$\tan(\text{ Dip Angle }) = (\text{elevation difference}) / (\text{horizontal distance}).$$

The strike direction of the plane passing through three layers observed at three boreholes i, j and k with coordinates (x_i, y_i), (x_j, y_j), (x_k, y_k) and layer elevations e_i, e_j, e_k respectively with $e_i > e_j > e_k$ is given as an angle relative to the y-axis (North) such that:

$$\tan(\text{ Strike Direction })=\dfrac{x_2 - x_1 - \dfrac{(e_1 - e_2)(x_3 - x_1)}{(e_1 - e_3)}}{y_2 - y_1 - \dfrac{(e_1 - e_2)(y_3 - y_1)}{(e_1 - e_3)}}$$

and

$$\text{Orientation Angle} = \text{Strike Direction} \pm 90^{\circ}$$

where the + or - signs are used accordingly, in order to produce a positive orientation angle. Hence, for every planar marker bed identified previously, a dip angle and a dip orientation relative to the North (both in degrees) can be calculated and then stored as:

$$mar\_bed(\text{Triangle\_no},[L_i,L_j,L_k],[\text{Dip},\text{Orientation}])$$

where Dip and Orientation define geometrically the plane of the marker bed. When both Dip and Orientation are equal to 0, it is assumed that the corresponding layer is horizontal.

(ii)   Application

The corresponding fact-lists after calculating the dip angle and dip orientation for every marker bed within each triangle are:

○   Triangle 1
          mar_bed(1,[1,1,1,3,3,4],[0.14,5,1.5,43]).
○   Triangle 3
          mar_bed(3,[1,1,1,3,4,4],[0.1,118,0.6,15]).
○   Triangle 4
          mar_bed(4,[1,1,1,3,3,4],[0,0,0.22,25]).
○   Triangle 5
          mar_bed(5,[1,1,1,4,4,5],[0.2,9,0.44,0]).
○   Triangle 6
          mar_bed(6,[1,1,1,5,4,3],[0.1,90,0.2,11]).
○   Triangle 7
          mar_bed(7,[1,1,1,5,3,5],[0.1,138,0.3,78])
          mar_bed(7,[1,1,1,5,3,7],[0.1,138,4,119]).
○   Triangle 10
          mar_bed(10,[1,1,1,3,3,7],[0,0,5.3,3])
          mar_bed(10,[1,1,1,3,8,5],[0,0,1.3,144]).
○   Triangle 11
          mar_bed(11,[1,1,1,5,7,5],[0,0,1.4,40])
          mar_bed(11,[1,1,1,7,3,3],[0,0,12,298]).
○   Triangle 12
          mar_bed(12,[1,1,1,3,3,3,5,7,5],[0,0,0.25,177,1.7,15]).

5.2.6   Continuity of Marker Beds Across the Site

(i)   Procedure

After identifying continuity of marker beds within each triangle and determining the geometric properties of their plane in the form of dip angle and dip orientation, their continuity across the whole site (or a significant part of it) must be assessed in order to construct a site-wide model of the ground conditions. The procedure involves correlation of marker beds belonging to triangles around common nodes (boreholes). Each node is examined and if the continuity is established between the first pair of triangles (having a common node), a third triangle (passing through the same node) is considered and its continuity with each one of the previously examined triangles is assessed. Iteration of this procedure until every node is examined (and thus every triangle is considered) can lead to a set of trends about the ground conditions.

By calling the fact-lists *mar_bed* for two triangles having a common node and thus one common borehole, a check is made to determine whether the marker bed(s) defined for this borehole is the same for both triangles. This can be done by observing the same pair of layer numbers in both fact-lists. In that case the continuity is established by examining compatibility of their geometric properties. The geometric constraints to be satisfied in order to achieve correlation are relative to the observed differences in the values of dip angle and orientation. If the dip angles for the triangles under consideration are less than 5°, the site is considered as 'near-level' and thus if the same bed appears in the boreholes participating, its continuity is established without examining whether the dip (angle and orientation) values follow an acceptable pattern.

For dip angles greater than 5°, the pattern of dip angles and dip orientations is considered as acceptable if the pair of beds appear to have differences in these values that conform with the following:

$$\text{Dip Angle Difference} < 10°$$

and

$$\text{Dip Orientation Difference} < 45°.$$

These criteria are provisional and could be adjusted in the light of greater experience in the use of the methodology, or could be individually specified by the user. Each time that the continuity of a marker bed is established in an additional triangle, its representation (*mar_bed* fact-list) is appended in a list of fact-lists that contains the already identified correlations. That list will have the form:

trend(No,[*mar_bed*(.....), *mar_bed*(......),.......]).

When the pair of triangles under consideration either do not appear to have the same marker bed or the compatibility of their geometric properties fails, then a new *trend* list is created containing that *mar_bed* fact-list and compatibility is examined between that trend and other marker beds.

Therefore a set of possible models about the ground conditions is created which can assist the engineer in the identification of continuous marker beds across the site and/or highlight to him/her areas where the ground conditions appear to be complex and hence additional investigation might be required.

(ii)   Application

For the site under consideration, three trends were identified concerning the presence of marker beds. The dip angles and dip orientation of these are schematically shown in Figures 5.5a to 5.5c, whereas the resulting cross sections (after Figure 5.1, cross-section AA') are shown in Figures 5.6a to 5.6c. Their representation within the methodology is shown below.

trend(1,[*mar_bed*(1,[1,1,1,3,3,4],[0.14,5,1.5,43]),
mar_bed(3,[1,1,1,3,4,4],[0.1,118,0.6,15]),
mar_bed(4,[1,1,1,3,3,4],[0,0,0.22,25]),*mar_bed*(5,[1,1,1,4,4,5],[0.2,9,0.44,0]),
mar_bed(6,[1,1,1,5,4,3],[0.1,90,0.2,11]),*mar_bed*(7,[1,1,1,5,3,5],[0.1,138,0.3,78]),
mar_bed(10,[1,1,1,3,8,5],[0,0,1.3,144]),*mar_bed*(11,[1,1,1,5,7,5],[0,0,1.4,40]),
mar_bed(12,[1,1,1,3,3,3,5,7,5],[0,0,0.25,177,1.7,15])]).

trend(2,[*mar_bed*(1,[1,1,1,3,3,4],[0.14,5,1.5,43]),
mar_bed(3,[1,1,1,3,4,4],[0.1,118,0.6,15]),
mar_bed(4,[1,1,1,3,3,4],[0,0,0.22,25]),*mar_bed*(5,[1,1,1,4,4,5],[0.2,9,0.22,25]),
mar_bed(6,[1,1,1,5,4,3],[0.1,90,0.2,11]),*mar_bed*(7,[1,1,1,5,3,7],[0.1,138,4,119]),
mar_bed(10,[1,1,1,3,3,7],[0,0,5.3,3]),*mar_bed*(11,[1,1,1],[0,0])
mar_bed(12,[1,1,1,3,3,3,5,7,5],[0,0,0.25,177,1.7,15])]).

trend(3,[*mar_bed*(1,[1,1,1],[0.14,5]),*mar_bed*(3,[1,1,1],[0.1,118]),
mar_bed(4,[1,1,1],[0,0]),*mar_bed*(5,[1,1,1],[0.2,9]),*mar_bed*(6,[1,1,1],[0.1,90]),
mar_bed(7,[1,1,1],[0.1,138]),*mar_bed*(10,[1,1,1],[0,0]),
mar_bed(11,[1,1,1,7,3,3],[0,0,12,298]),*mar_bed*(12,[1,1,1],[0,0])]).

The trends were listed in order of importance. The first one is the strongest one because all triangles participate and two continuous marker beds over the whole site appear consistent as these are 'near-level'. Also in one triangle (*triangle*(12,0.840,[6,8,7])) a third 'near-level' marker bed seems to exist.

The second trend is indicative of two 'near-level' continuous marker beds again, but one triangle (*triangle*(11,0.956,[5,8,6])) does not fully participate in the configuration (only the PEAT marker bed exists) and another (*triangle*(10,0.598,[10,8,5])) appears to have a dip angle greater than $5°$ and thus seems to deviate from the other low dips ($<5°$) of the hypothesis.

The third trend is the weakest because the continuity of the inorganic soils is established only within one triangle (*triangle*(11,0.956,[5,8,6])) and at the same time the dip angle of the bed is relatively high ($12°$), possibly indicating two different horizons.

Figure 5.5a   Schematic representation of the first trend.

Figure 5.5b   Schematic representation of the second trend.

KEY

→ : Coarse grained marker bed
↑ : PEAT marker bed
↑L : Level PEAT marker bed

111

N

KEY
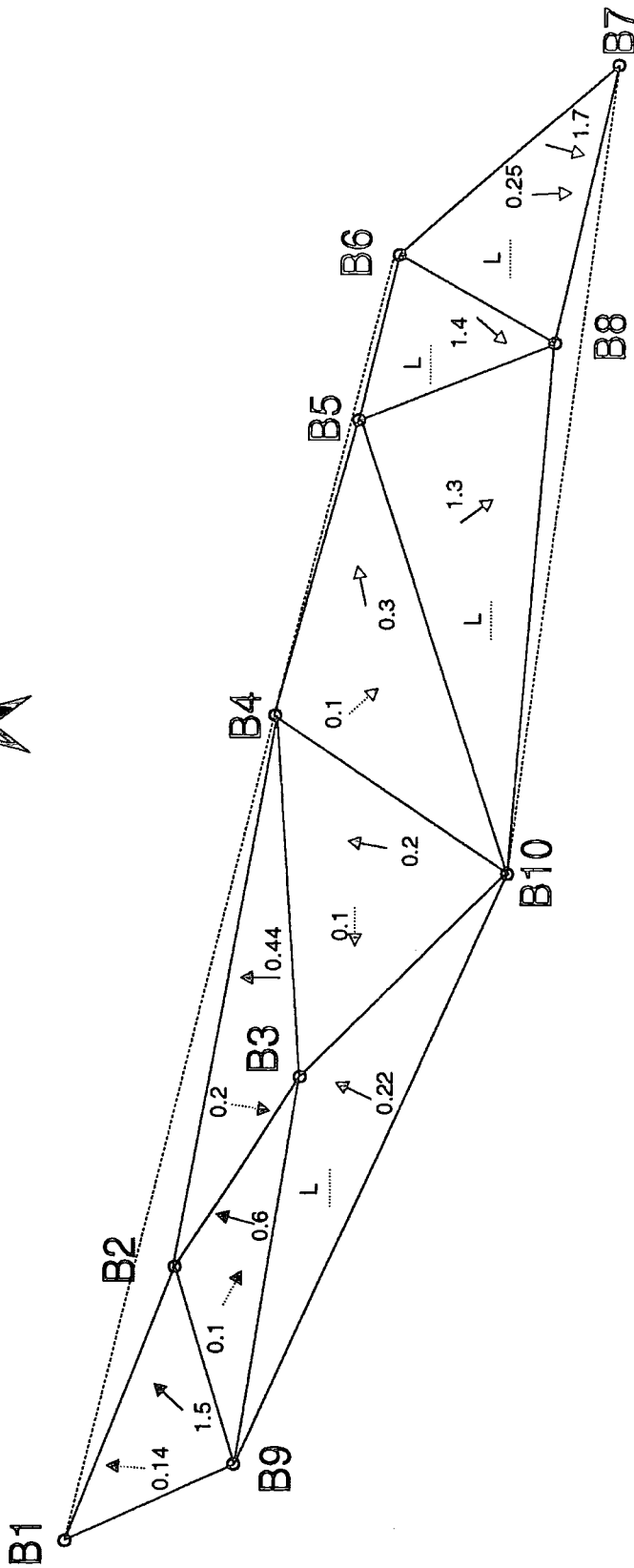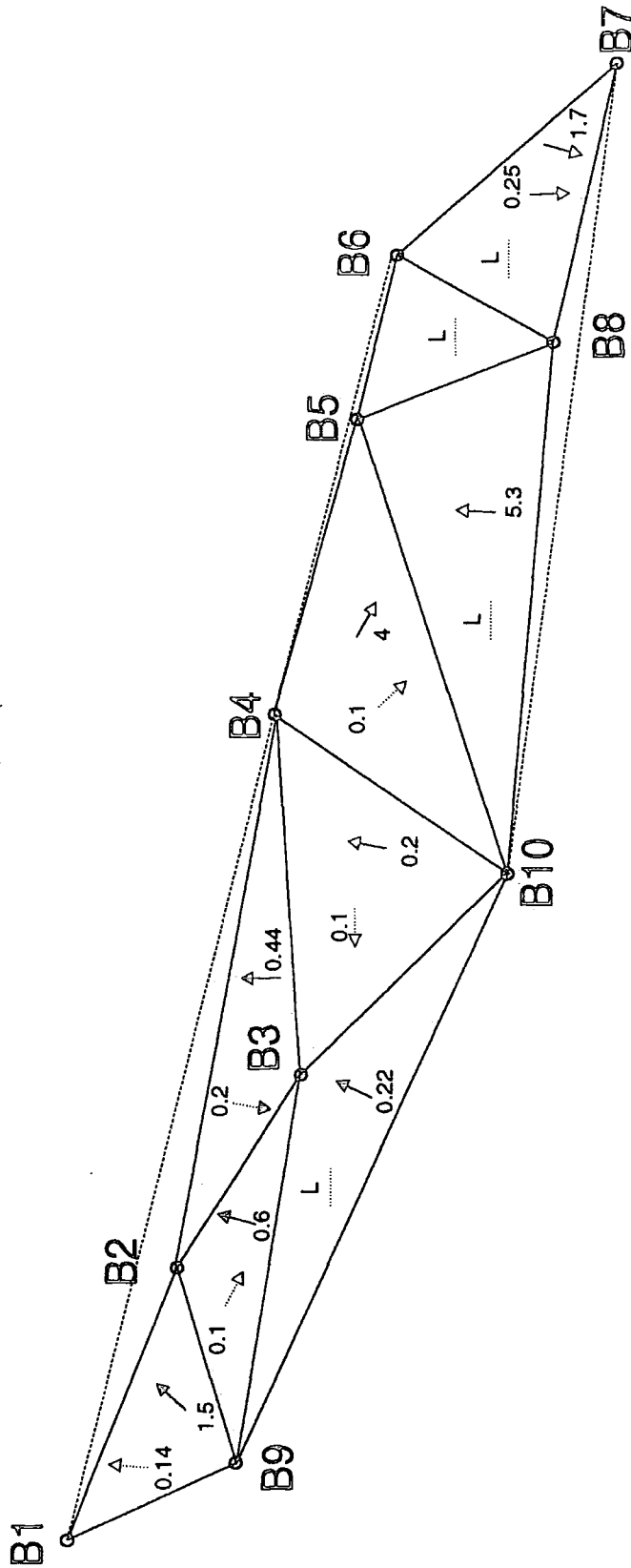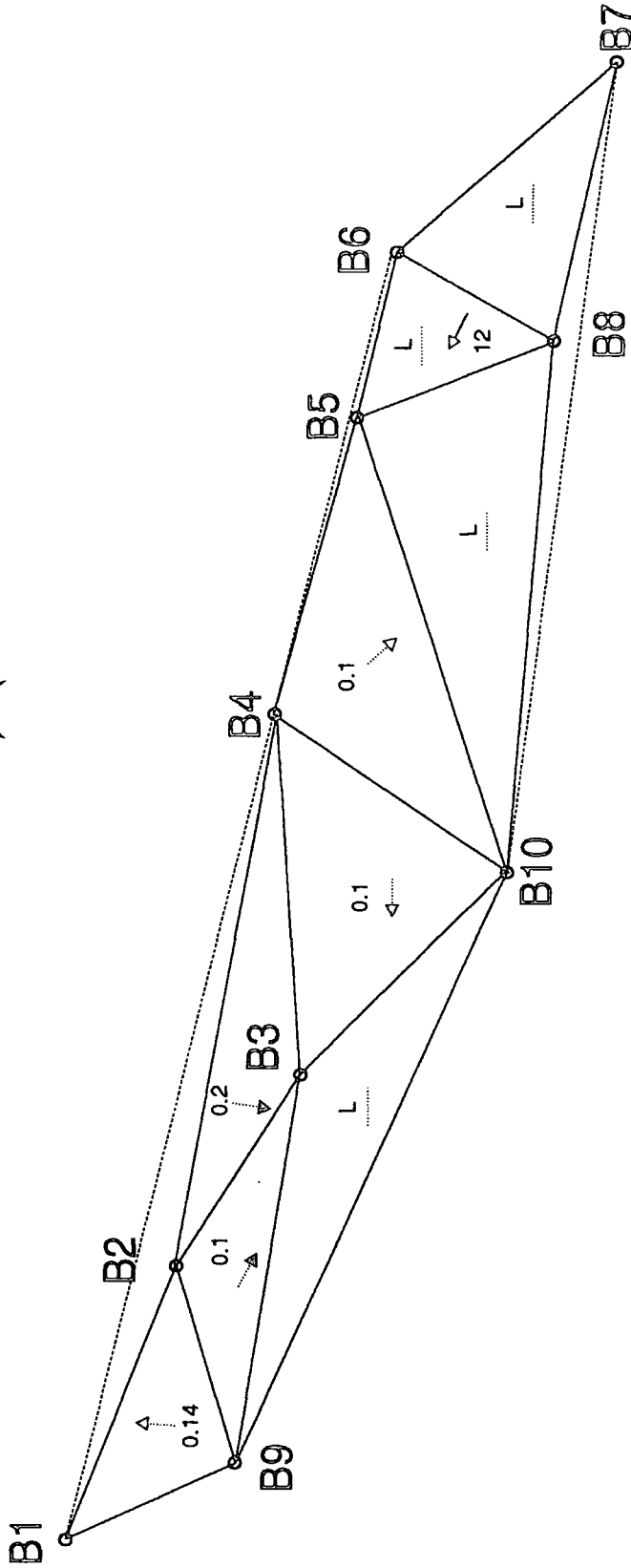
→ : Coarse grained marker bed

⊳ : PEAT marker bed

L : Level PEAT marker bed

Figure 5.5c   Schematic representation of the third trend.
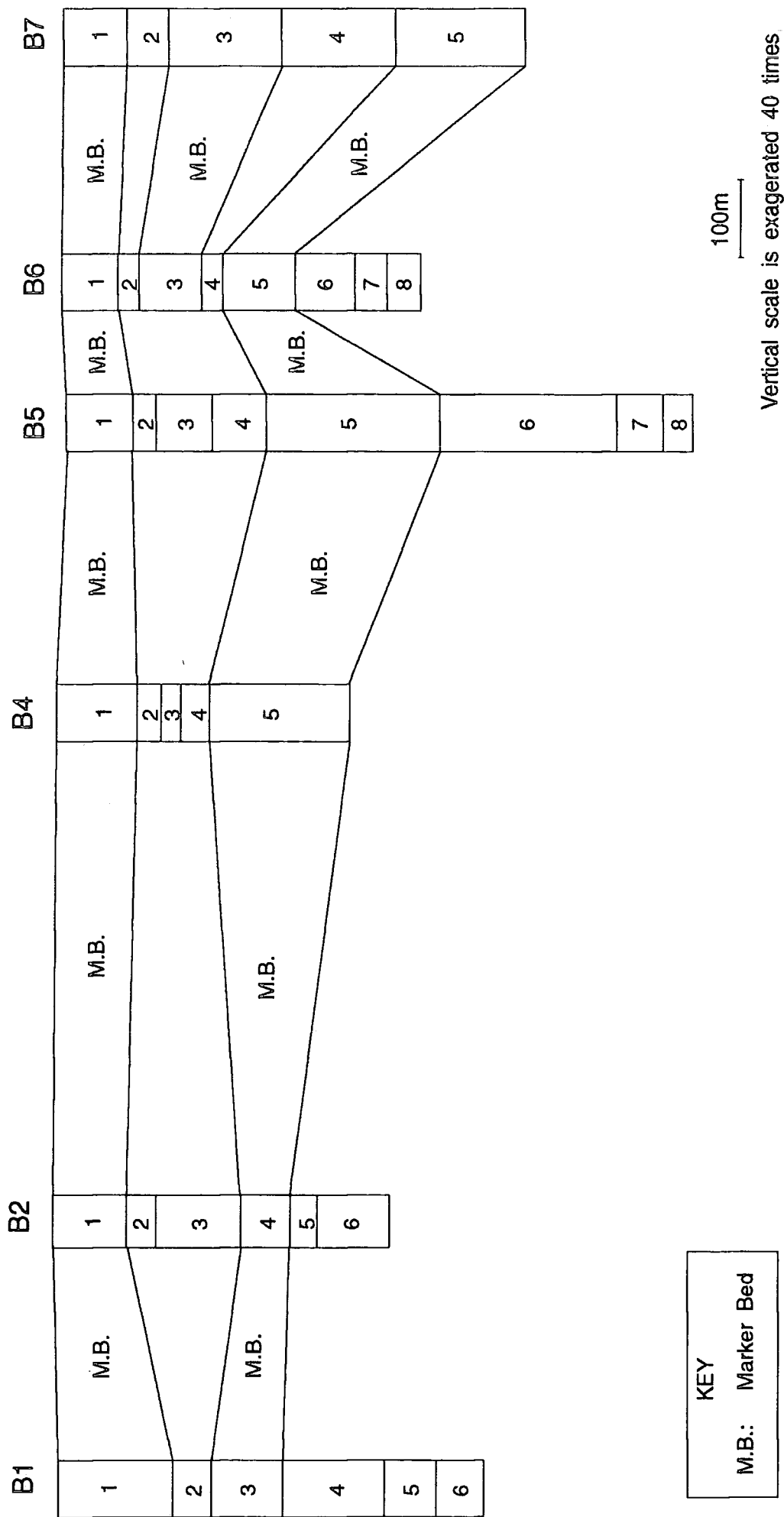
Figure 5.6a   Schematic representation of the cross-section AA', shown in Figure 5.1, in accordance with the first trend.

KEY

M.B.:   Marker Bed

100m

Vertical scale is exagerated 40 times

113
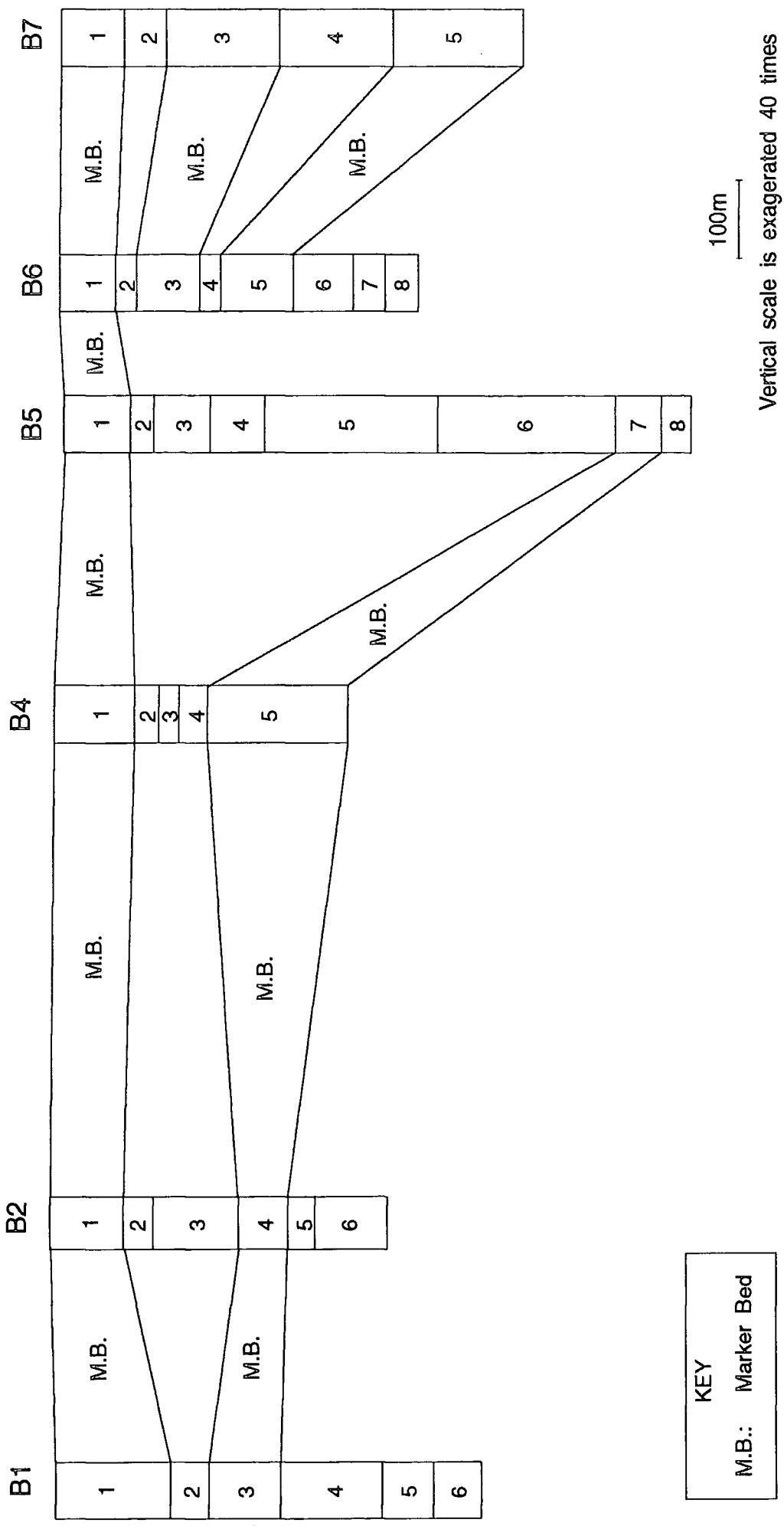
Figure 5.6b   Schematic representation of the cross-section AA', shown in Figure 5.1, in accordance with the second trend.

KEY

M.B.:   Marker Bed

100m

Vertical scale is exagerated 40 times

114
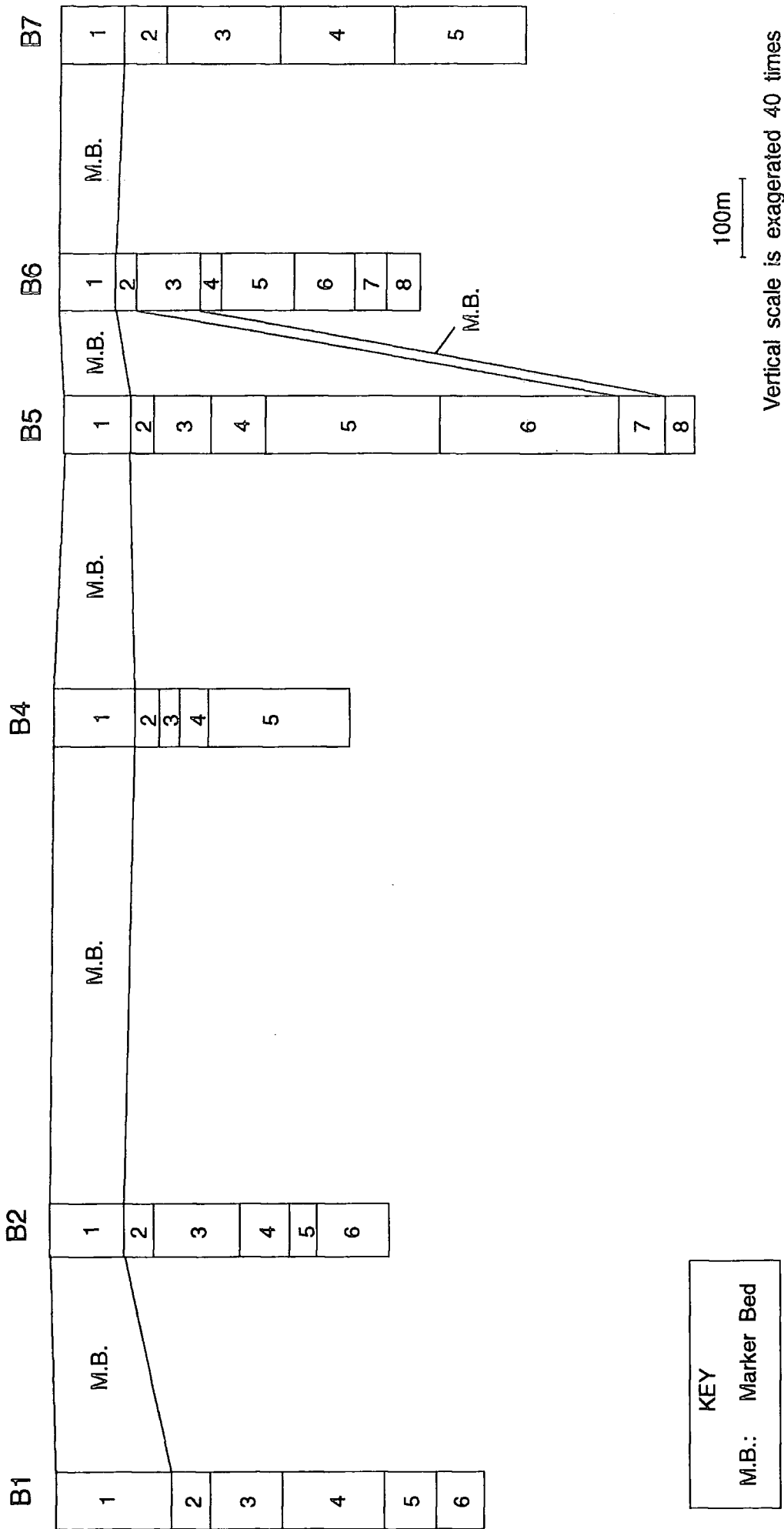
Figure 5.6c   Schematic representation of the cross-section AA', shown in Figure 5.1, in accordance with the third trend.

KEY

M.B.:   Marker Bed

100m

Vertical scale is exagerated 40 times

115

5.2.7 Borehole-to-Borehole Correlation

(i) Procedure

After completing the site-wide approach to the correlation problem, a model of the ground conditions is constructed in terms of marker beds. However, if the ground conditions are very complex, it may not be possible to infer site-wide correlations. Therefore, it becomes necessary to examine the detailed ground conditions on a borehole-to-borehole basis. At this level, a detailed examination of the ground conditions is made between pairs of neighbouring boreholes.

Using the methodology for comparing soils that was introduced in the previous Chapter, a Similarity Number is calculated between each layer of the first borehole and all of the layers of the other borehole. Hence for every pair of neighbouring boreholes a matrix of Similarity Numbers can be produced having the general form:

		Borehole 2				
		Layer 1	Layer 2	Layer 3	....	Layer k
	Layer 1	S_{11}	S_{12}	S_{13}	....	S_{1k}
Borehole	Layer 2	S_{21}	S_{22}	S_{23}	....	S_{2k}
1	Layer 3	S_{31}	S_{32}	S_{33}	....	S_{3k}
	.	.	.	.	....	.
	.	.	.	.	....	.
	Layer n	S_{n1}	S_{n2}	S_{n3}	....	S_{nk}

where 1, 2, 3,....,n are the layers of Borehole 1 and 1, 2, 3,.....,k are the layers of Borehole 2 and S_{nk} represents the Similarity Number between layers n and k of these boreholes. A potential link between layer n of Borehole 1 and layer k of Borehole 2 is identified if the Similarity Number S_{nk} is maximum for either the n-th row of the matrix or the k-th column of it and that number is greater than 90. Once such links have been identified for all layers (if possible), they are then used to construct valid

hypotheses about the soil profile between the two boreholes. These hypotheses comprise a set of compatible links, where compatibility between links is established by observing that layers do not 'cross over'. The fact that layers could die-out between the borehole pairs can be allowed for in the hypotheses. Hence, a set of valid hypotheses are generated which are ranked in order of importance. The higher the number of the participating layers and the larger the average of the observed Similarity Numbers, the better the hypothesis is considered to be. If a particularly strong site-wide trend has been identified, this can be used to fix, one or more, of the links, thus limiting the number of hypotheses that the system can generate.

The borehole-to-borehole approach has been implemented in PDC-Prolog. The user is queried about the borehole numbers to be examined and a set of hypotheses is generated based on the observed similarities between soil layers in terms of soil type. The user is also queried about the existence of fixed links. Each generated hypothesis is presented as a list of integer pairs referring to the pair of boreholes under consideration. Thus, if the entered boreholes are A and B, a hypothesis of the form [1,1,2,2,3,3] indicates that layer 1 of borehole A correlates to layer 1 of borehole B, layer 2 of borehole A correlates to layer 2 of borehole B and layer 3 of borehole A correlates to layer 3 of borehole B. The listing of the program is shown in Appendix B.

(ii)   Application

In the application three pairs of boreholes are examined to demonstrate the methodology, each one belonging to different triangles.

o   *triangle*(1,0.706,[1,9,2])

If boreholes 1 and 2 are entered in the borehole-to-borehole correlation program the generated matrix of Similarity Numbers will have the form :

Borehole   2

		Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6
	Layer 1	100	0	0	0	0	0
Borehole	Layer 2	0	100	94	73	93	98
1	Layer 3	0	82	76	91	85	83
	Layer 4	0	93	92	75	100	95
	Layer 5	0	98	93	73	95	100
	Layer 6	0	99	93	73	93	98

The highlighted numbers are the identified potential links because they are the maximums of each row or column and all are higher than 90. Thus, based on these links the generated hypotheses about the soil profile between boreholes 1 and 2 are :

o   Hypothesis 1 :   [1,1,2,2,2,3,3,4,4,5,5,6]

o   Hypothesis 2 :   [1,1,2,2,6,2]

Link [6,2] cannot be incorporated in Hypothesis 1 as it would 'cross over' links [2,3], [3,4], [4,5] and [5,6]. Therefore a second hypothesis is generated, containing the links that are compatible with [6,2]. However, in the first and second *trend* lists the *mar_bed*(1,[1,1,1,3,3,4]) is present. This corresponds to *triangle*(1,0.706,[1,9,2]), indicating that layer 1 of borehole 1 links with layer 1 of borehole 9 and layer 1 of borehole 2, and that layer 3 of borehole 1 links with layer 3 of borehole 9 and layer 4

of borehole 2. Hence the marker beds identified at the site-wide approach can be established in the form of the fixed links [1,1,3,4] for boreholes 1 and 2. If these fixed links are imposed, only Hypothesis 1 is the output of the program. Therefore, Hypothesis 1 seems to be the strongest one because it is in accordance with the first site-wide trend and further more layers are participating. A schematic representation of Hypothesis 1 is shown in Figure 5.7.

     o   *triangle*(11,0.956,[5,8,6])

If boreholes 5 and 6 are examined, both belonging to triangle 11 where two conflicting marker beds consisting of inorganic soil types appear to exist (identified earlier in the first and third *trend* lists). By considering the fixed links [1,1,5,5], that were identified in the first trend, one hypothesis is generated by the program having the form:

     o   Hypothesis 1 :   [1,1,3,2,3,4,5,5,6,6,6,7,8,7].

If, instead, the fixed links are [1,1,7,3], identified in the third trend, the resulting hypothesis is:

     o   Hypothesis 1 :   [1,1,3,2,7,3,8,7].

These correlations are shown in Figures 5.8a and 5.8b. Thus, at the borehole-to-borehole level, the first Hypothesis seems stronger (more layers participating) and therefore the marker bed defined by layer 5 of borehole 5 and layer 5 of borehole 6 (identified in the first trend) is more likely to exist than the marker bed defined by layer 7 of borehole 5 and layer 3 of borehole 6 (identified in the third trend). In this case it can be seen that the borehole-to-borehole correlation can assist in the re-assessment of the site-wide model.

119

BOREHOLE 1

BOREHOLE 2

Layer 1   PEAT

Layer 2   Very silty slightly sandy CLAY

Layer 3   Silty SAND with a little gravel

Layer 4   Sandy CLAY

Layer 5   Silty sandy CLAY

Layer 6   Very silty sandy CLAY

PEAT   Layer 1

Very silty slightly sandy CLAY   Layer 2

Silty CLAY   Layer 3

SAND and GRAVEL   Layer 4

Sandy CLAY   Layer 5

Silty sandy CLAY   Layer 6

KEY
——▷ : Fixed links
——▷ : Links

Figure 5.7   Strongest generated hypothesis for boreholes 1 and 2.

BOREHOLE 5

BOREHOLE 6

Layer 1 — PEAT

Layer 2 — V. silty sandy CLAY

Layer 3 — Very silty slightly sandy CLAY

Layer 4 — Silty very sandy CLAY

Layer 5 — Clayey very sandy GRAVEL

Layer 6 — Slightly silty sandy CLAY

Layer 7 — Slightly gravelly SAND

Layer 8 — Very silty sandy CLAY

PEAT — Layer 1

V.silty s.sandy CLAY — Layer 2

SAND with a trace of gravel — Layer 3

V.silty s.sandy CLAY — Layer 4

GRAVEL with a little sand — Layer 5

Very sandy silty CLAY — Layer 6

V. silty sandy CLAY — Layer 7

V.silty s.sandy CLAY — Layer 8

KEY

⟶ : Fixed links

⟶ : Links

Figure 5.8a   Strongest generated hypothesis for boreholes 5 and 6 for the

shown fixed links, identified in the first trend

121

BOREHOLE 5

BOREHOLE 6

Layer 1   PEAT

PEAT   Layer 1

V.silty s.sandy CLAY   Layer 2

Layer 2   V. silty sandy CLAY

SAND with a
trace of gravel   Layer 3

Layer 3   Very silty slightly
sandy CLAY

V.silty s.sandy CLAY   Layer 4

Layer 4   Silty very sandy
CLAY

GRAVEL with
a little sand   Layer 5

Layer 5   Clayey
very sandy
GRAVEL

Very sandy
silty CLAY   Layer 6

V. silty sandy CLAY   Layer 7

V.silty s.sandy CLAY   Layer 8

Layer 6   Slightly silty
sandy CLAY

KEY

⟶▷ : Fixed links

⟶▷ : Links

Layer 7   Slightly gravelly
SAND

Layer 8   Very silty
sandy CLAY

Figure 5.8b   Strongest generated hypothesis for boreholes 5 and 6 for the

fixed links, identified in the third trend

122

o   *triangle*(12,0.840,[6,8,7]

Boreholes 6 and 7 are examined, both belonging to triangle 12, where the presence of three marker beds was assessed at the site-wide level, in the first and second trends. If only the PEAT marker bed is established ([1,1]) the borehole-to-borehole correlation generates the following hypotheses :

o   Hypothesis 1 :   [1,1,2,2,2,4,4,4,6,4,7,4,8,4].

o   Hypothesis 2 :   [1,1,2,2,3,3,4,4,6,4,7,4,8,4].

o   Hypothesis 3 :   [1,1,2,2,4,2,4,4,6,4,7,4,8,4].

o   Hypothesis 4 :   [1,1,2,2,4,2,6,2,6,4,7,4,8,4].

o   Hypothesis 5 :   [1,1,2,2,4,2,6,2,7,2,7,4,8,4].

o   Hypothesis 6 :   [1,1,2,2,4,2,6,2,7,2,8,2,8,4].

o   Hypothesis 7 :   [1,1,2,2,2,4,4,4,5,5].

o   Hypothesis 8 :   [1,1,2,2,3,3,4,4,5,5].

o   Hypothesis 9 :   [1,1,2,2,4,2,4,4,5,5].

The number of the resulting hypotheses can be reduced if the site-wide inferences for triangle 12 are entered in the program in the form of the fixed links [1,1,3,3,5,5]. In that case the program returns only Hypothesis 8, and thus it becomes evident that even if other hypotheses seemed stronger because more layers were participating, the site-wide approach assisted in order to identify the one that appears to be the most correct interpretation of the ground conditions. This hypothesis is shown in Figure 5.9.

Finally, for the strongest site-wide trend, both site-wide and borehole-to-borehole inferences are shown in Figures 5.10, in accordance with the cross-section AA' shown in Figure 5.1.

BOREHOLE 6

BOREHOLE 7

Layer 1	PEAT
Layer 2	V.silty s.sandy CLAY
Layer 3	SAND with a trace of gravel
Layer 4	V.silty s.sandy CLAY
Layer 5	GRAVEL with a little sand
Layer 6	Very sandy silty CLAY
Layer 7	V. silty sandy CLAY
Layer 8	V.silty s.sandy CLAY

PEAT	Layer 1
Very silty slightly sandy CLAY	Layer 2
Silty SAND	Layer 3
Very silty slightly sandy CLAY	Layer 4
GRAVEL with some sand	Layer 5

KEY

⟶ : Fixed links

⟶ : Links

Figure 5.9   Strongest generated hypothesis for boreholes 6 and 7.
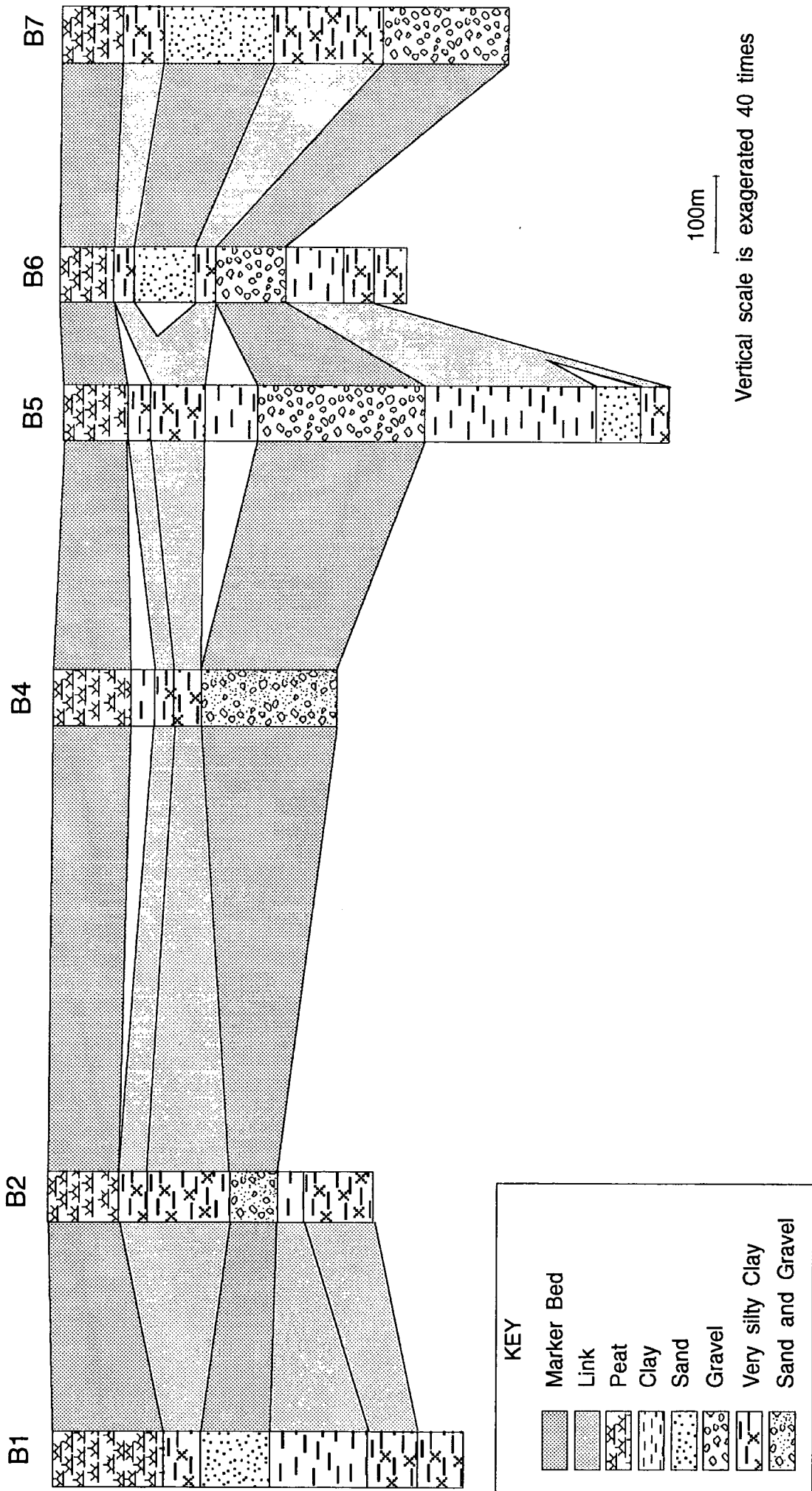
Figure 5.10   Schematic representation of the cross-section AA', shown in Figure 5.1, in accordance with the first trend, for both site-wide and borehole-to-borehole inferences.

5.3 Discussion

The proposed methodology for correlating borehole information makes use of soil type observations in order to assess the presence and continuity of marker beds. In addition to that, colour and consistency are important features for extracting trends about layers that 'stand out' from the general ground conditions. Using the quantitative representation showed earlier in this chapter, the same methodology can be used in order to infer the existence of marker beds and perform correlations where this is possible, based on colour and consistency.

Further, the inferences that are reached using the methodology are mainly based on observations of compatibility about the geometric properties of the layers encountered. Knowledge about geology should be included in the methodology, allowing for trends to be made in areas where the dip angles and dip orientations do not follow an acceptable pattern and to re-assess inferences based on geometric compatibility.

Finally, the proposed methodology is not intended to provide the geotechnical engineer with fixed solutions, but to assist him/her in order to achieve an outline of the ground conditions and to indicate areas within a site where complex conditions exist. The judgement of the engineer should be the key factor for validating the inferences reached and for selecting the most appropriate to be indicative of the existing conditions.

CHAPTER 6

DISCUSSION AND CONCLUSIONS

6.1 Discussion

Site investigation aims at determining depositional geometry and properties that are relevant to the construction of an engineering project and to indicate areas where additional information is needed.

In this work has been presented a way of dealing with this problem which could aid the engineer by producing a preliminary assessment of the ground conditions, having as input the borehole arrangement of a site, together with the engineering soil descriptions and the elevations of every layer within each borehole. A modular decision-support system is proposed for use in the site characterization procedure. The modules, as they have been implemented are discussed below, and their role in a final knowledge-based system which is under development at Durham University is considered.

The parser module described in Chapter 4 is essential for breaking down and identifying the contributing parts of a soil description. In this work, the parser feeds the required information directly to the VASIC module for processing. However, in the final system, the parser will be utilised as a critical part of the front end of a proposed site investigation database, thus allowing for the storage of the qualitative information obtained from borehole records, [71].

127

The Values Assignment and Similarity Calculation (VASIC) module (currently used in parallel with the parser module), provides a means of producing a quantitative representation of the recovered descriptive information, and therefore a means of making comparisons between soil layers having complex descriptions. This forms the core of the site-wide and borehole-to-borehole interpretation procedures.

Next, according to the engineer's requirements, a site-wide examination of the whole site (or part of it) can take place. After processing the relevant data, a list of trends can be produced, indicating possible marker beds present and their continuity over the site, based on assessment of their geometrical properties in the form of dip angle and dip orientation. The resulting trends can be listed in order of importance, but the engineer should judge the significance of every trend, based on his/her knowledge and experience about the geology and geomorphology of the area under consideration and according to the nature of the project to be completed.

In areas where more detailed information is required, the engineer can select to proceed with the borehole-to-borehole correlation which can either strengthen or weaken the trends obtained from the site-wide consideration. Again, it is the engineer's responsibility to consider which marker beds to establish (fixed links), either by selecting one of the resulting trends (which would create fixed links for several boreholes over the site) or by establishing links for pairs of boreholes. Then, he/she can proceed to correlate boreholes and eventually to select which of the generated hypotheses is the most likely to represent the existing ground conditions.

The proposed system is not intended to be a tool able to produce definite solutions about the subsurface conditions, but as a decision-support system to assist the engineer. The main advantages of the system over existing ones are its ability to handle complex soil descriptions, to treat sites with varying ground conditions and to perform the site characterization process in a way similar to that of an engineer.

Both SITECHAR, [47, 55], and CONE, [44, 45], cannot handle soil descriptions containing more than a main and a secondary soil type, in contrast with the proposed system that can perform quantitative analysis of qualitative data containing complex engineering soil descriptions. In addition, its ability to examine sites which are different in character, means that it does not have the problem that is present in LOGS, [1], of being a very site-specific system. Finally, the two levels approach (site-wide and borehole-to-borehole) used for the interpretation of the ground conditions seems to be in good agreement with the methodology that an engineer follows in order to characterize a site, as is described by Norkin, [47], where it is stressed that a site-wide approach can result to inferences about layers that 'stand out' from the general ground conditions (marker beds) and can then assist in the identification of the detailed soil profile.

Possible improvements of the system, by incorporating knowledge about geology in the form of rules, can aid in producing more refined inferences about the site, but still the engineer is to play the major part in evaluating the output of the system. Interaction between the engineer and the system is another area that needs addressing. The system is envisaged as highly interactive, and a graphical interface seems to be the best solution. By presenting every trend in terms of two-dimensional cross-sections (shown in Figures 5.6a to 5.6c) together with the schematic dip angle and dip orientation configuration over the site (shown in Figures 5.5a to 5.5c), the engineer will be able to visualize the ground conditions and to assess the importance of each trend both graphically and numerically (from *trend* lists).

To conclude, the proposed system is viewed as an intelligent assistant to the engineer. He/she must always direct the system, to avoid misinterpretations, towards the 'correct' inferences about the ground conditions that can lead to a safe and economic design.

6.2   Conclusions

Geotechnical design requires the interpretation of ground conditions from site investigation information. This normally requires interpolating or extrapolating from observations at discrete points, such as boreholes. If a knowledge-based system is to assist in this interpretation it must be able to compare soils observed at two or more locations, in order to identify whether the soils observed belong to the same horizon.

A methodology has been developed whereby the similarity of two soils can be calculated, based on engineering soil descriptions. The qualitative terms are converted into quantitative representations from which a Similarity Number can be derived.

Individual Similarity Numbers can be calculated for each of the factors identified : soil type, consistency, structure and colour. These are normalised to give values between 0 and 100 (with 100 indicating identical features) and combined using appropriate weighting factors to give an Overall Similarity Number which represents a comparison based on these features. A consistent set of weighting factors was determined from a small knowledge elicitation exercise. These preliminary weighting factors should be modified when the knowledge elicitation exercise is complete.

Using the quantitative representation of the soil descriptions, a preliminary assessment of the ground conditions can be made. The correlation of the borehole information is approached at two levels. At the site-wide level, an attempt is made to identify marker beds, considering that these are soil layers which 'stand out' from the general existing ground conditions.

A search for possible marker beds is first made at each borehole. The search is then extended to pairs of boreholes and further, the continuity of marker beds is established inside triangles which were formed having the boreholes as vertices. Where continuous layers are observed, the dip angle and dip orientation are calculated, to form the geometrical parameters on which preliminary conclusions are based. Compatibility (or not) of these parameters between neighbouring triangles is the key factor for assessing continuity of the marker beds.

Finally, the detailed ground conditions are examined on a borehole-to-borehole level. At this level, a set of hypotheses about the ground conditions is constructed by looking at pairs of adjacent boreholes. Hence, a set of hypotheses is produced, even for areas for which the site-wide level approach has been unable to establish trends.

The proposed system is to be part of a bigger Knowledge-Based System which is under development at the University of Durham. The development of this system is being done in a modular manner, operating around a central database of site investigation information. The main objective of the system under development is the interpretation of the ground conditions from borehole logs and the interpretation of design parameters from laboratory and field test results.

REFERENCES

1. Adams T.M., Christiano P. and Hendrickson C. "Some expert system applications in geotechnical engineering", in *Foundation Engineering: Current principles and practices*, ASCE, New York, 1989.

2. Alim S. and Munro J. "PROLOG-Based Expert Systems in Civil Engineering", *Proc. Institution of Civil Engineers*, Part 2, Vol. 83, pp. 1-14, 1987.

3. Arockiasamy M., Radhakrishnan N., Sreenivasan G. and Lee S. "KBES Applications to the Selection and Design of Retaining Structures", Proceedings of the Geotechnical Engineering Congress, in *Geotechnical Special Publication*, Vol. 1, No. 27, (ed. McLean F.G., Campbell DW. A. and Harris D.W.), ASCE, pp. 391-402, Boulder, Colorado, 1991.

4. Badgley P. C. *Structural Methods for the Exploration Geologist*, Harper and Brothers Publishers, New York, 1970.

5. Baumbach U. and Plumer L. "Schikorre : A Knowledge Based System for the Correlation of Stratigraphic Sequences in Prolog", in *Proceedings of the International Conference for the Practical Applications of Prolog*, Vol. 1, London, 1992.

6. British Standard 5930. *Code of Practice for Site Investigations*, British Standards Institution, London, 1981.

7. Burland J.B. "The teaching of soil mechanics - a personal view". The Nash lecture, *Proc. 9th European Conf. Soil Mechanics and Foundation Engineering*, Vol 3, pp 1427-1447, Balkema, Rotterdam, 1987.

8. Carpaneto R. and Cremomini M.G. "Evaluation of Geotechnical Design Parameters by Expert System Techniques", Proceedings of the Geotechnical Engineering Congress, in *Geotechnical Special Publication*, Vol. 1, No. 27, (ed.

McLean F.G., Campbell DW. A. and Harris D.W.), ASCE, pp. 422-433, Boulder, Colorado, 1991.

9.   Chahine J.R. and Janson B.N. "Interfacing Databases with Expert Systems: a Retaining Wall Management Application", *Microcomputers in Civil Engineering*,Vol. 2, No. 1, pp. 19-38, 1987.

10.   Chameau J-L. and Santamarina J.C. "Knowledge-Based System for Soil Improvement", *Journal of Computing in Civil Engineering*, ASCE, Vol. 3, No. 3, pp. 253-267, 1989.

11.   Child G. H. "Soil Descriptions - Quo Vadis?", in *Site Investigation Practice : Assessing BS5930*, Geological Society, Engineering Geology Special Publication No.2, London, pp 73-81,1986.

12.   Clayton C. R. I., Simons N. E. and Matthews M. C. *Site Investigation*, Granada Publishing Ltd, London, 1982.

13.   Craig R. F. *Soil Mechanics*, Van Nostrand Reinhold Co. Ltd, UK, 1987.

14.   Davey-Wilson I.E.G. "Development of a Prolog Based Expert System for Groundwater Control", in *Application of A.I. techniques to Civil and Structural Engineering* (ed. Topping B.H.V.), CIVIL-COMP Press, Edinburgh, pp. 263-268, 1989.

15.   Davey-Wilson I.E.G. "Geotechnical Laboratory Test Simulation using AI Techniques",in *Artificial Intelligence and Civil Engineering* (ed. Topping B.H.V.), CIVIL-COMP Press, Edinburgh, pp. 119-124, 1991.

16.   Davey-Wilson I.E.G. and May I.M. "Development of a knowledge-based system for the selection of groundwater control methods", *Computers and Geotechnics*, 7, pp. 189-203, 1989.

17.   Davey-Wilson I.E.G., May I.M. and Tizani W. "Discussion on PROLOG-Based Expert Systems in Civil Engineering", *Proc. Institution of Civil Engineers*, Part 2, Vol. 85, pp. 185-187, 1988.

18.   Dimmick K., Bhatia S.K. and Hassett J. "Geotextile Edge Drain Design and Specification by Expert System", Proceedings of the Geotechnical Engineering

Congress, in *Geotechnical Special Publication*, Vol. 1, No. 27, (ed. McLean F.G., Campbell DW. A. and Harris D.W.), ASCE, pp. 288-297, Boulder, Colorado, 1991.

19.   Dumbleton M. J. and West G. *Guidance on Planning, Directing and Reporting Site Investigations*, Report LR625, Transport and Road Research Laboratory, Crowthorne, 1974.

20.   Elton D.L. and Brown D.A. "Expert System for Driven Pile Selection", Proceedings of the Geotechnical Engineering Congress, in *Geotechnical Special Publication*, Vol. 1, No. 27, (ed. McLean F.G., Campbell DW. A. and Harris D.W.), ASCE, pp. 253-263, Boulder, Colorado, 1991.

21.   Faure R.M., Mascarelli D., Zelfani M., Charveriat L., Gandar J. and Mosuro O. "XPENT: An Expert System for Slope Stability", in *Artificial Intelligence and Civil Engineering* (ed. Topping B.H.V.), CIVIL-COMP Press, pp. 143-147, Edinburgh, 1991.

22.   Feigenbaum E. "Expert Systems in the 80's", in *Machine Intelligence*, A. Bond, 1981.

23.   Frederick C. O., Wong Y. C. and Edge F. W. "Two Dimensional Automatic Mesh Generation for Structural Analysis", in *International Journal of Numerical Methods in Engineering*, Vol. 2, pp. 133-144, 1970.

24.   Gillette D.R. "An Expert System for Estimating Soil Strength Parameters", Proceedings of the Geotechnical Engineering Congress, in *Geotechnical Special Publication*, Vol. 1, No. 27, (ed. McLean F.G., Campbell DW. A. and Harris D.W.), ASCE, pp. 276-287, Boulder, Colorado, 1991.

25.   Gordon J. and Abramov I. "Scaling Procedures for Specifying Colour Appearance", Colour Research and Application, Vol./Part 13 (3), pp. 146-152, 1988.

26.   Hadipriono F.C., Diaz C.F. and Wolfe W.E. "Toward the Development of a Knowledge Base Expert System for Determining the Causes of Foundation Failures", *Proc. of Conf. on Computational Structures Technology*, CIVIL-COMP Press, Herriot-Watt Univ., Edinbourgh, 1991.

27.   Hadipriono F.C. and Wolfe W.E. "Repairability Assessment of Damaged Foundations", in *Artificial Intelligence and Civil Engineering* (ed. Topping B.H.V.), CIVIL-COMP Press, pp. 137-141, Edinburgh, 1991.

28.   Halim I.S., Tang W.H. and Garrett J.H.Jr. "Knowledge-Assisted Interactive Probabilistic Site Characterization", Proceedings of the Geotechnical Engineering Congress, in *Geotechnical Special Publication*, Vol. 1, No. 27, (ed. McLean F.G., Campbell DW. A. and Harris D.W.), ASCE, pp. 264-275, Boulder, Colorado, 1991.

29.   Head J. M. "Planning and Design of Site Investigations", in *Site Investigation Practice : Assessing BS5930*, Geological Society, Engineering Geology Special Publication No.2, London, pp 1-8,1986.

30.   Hutchinson P.J., Rosenman M.A. and Gero J.S. "RETWALL: An Expert System for the Selection and Preliminary Design of Earth Retaining Structures", in *Knowledge-based systems*, Vol. 1, No. 1, pp. 11-23, 1987.

31.   Institution of Civil Engineers, *Inadequate Site Investigation*, Thomas Telford, London, 1991.

32.   Jennings J.E. and Brink A.B.A. "Application of geotechnics to the solution of engineering problems". *Proc. Institution of Civil Engineers*, Part 1, 64, pp. 571-589, 1978.

33.   Joyce M. D. *Site Investigation Practice*, E. and F. Spon, New York, U.S.A., 1982.

34.   Juang C.H. and Lee D.H. "Development of an Expert System for Rock Mass Classification", *Civil Engineering Systems*, Vol. 6, pp. 147-156, 1989.

35.   Lindholm D. A. "Automatic Triangular Mesh Generation on Surfaces of Polyhedra", in *IEEE Transactions on Magnetics*, Vol. MAG-19 Wo. 6, pp. 2539-2542, 1983.

36.   Maher M.H. and Williams T.P. "A Hybrid Expert System for Design with Geosynthetics", Proceedings of the Geotechnical Engineering Congress, in *Geotechnical Special Publication*, Vol. 1, No. 27, (ed. McLean F.G., Campbell DW. A. and Harris D.W.), ASCE, pp. 241-252, Boulder, Colorado, 1991.

37.   Maltman A. *Geological Maps: An Introduction*, Open University Press, Milton Keynes, 1990.

38.   Meyer S. *Preliminary Foundation Design Using EDESYN*, NATO Advanced Study Institute: Optimisation and Decision Support Systems in Civil Engineering, Heriot-Watt University, June 1989.

39.   Mi Z. and Jieliang P. "An Expert System of Predicting and Preventing Surface Settlement Caused by Shield-Driven Tunneling in City", *Proceedings of International Conference '89 on Expert Systems in Engineering Applications*, Huazhong University of Science and Technology Press, pp. 466-472, 1989.

40.   Microsoft Corporation *Microsoft Windows 3 Users Guide*, U.S.A., 1990

41.   Motamed F., Salazar G. and D'Andrea R. "An Expert System for Preliminary Ground Improvement Selection", Proceedings of the Geotechnical Engineering Congress, in *Geotechnical Special Publication*, Vol. 1, No. 27, (ed. McLean F.G., Campbell DW. A. and Harris D.W.), ASCE, pp. 379-390, Boulder, Colorado, 1991.

42.   Moula M. A Knowledge-Based System to Assist in the Selection of the Appropriate Geotechnical Field Tests, Phd Thesis, University of Durham, (in preparation), 1992.

43.   Moula M. and Vaptismas N. "Knowledge-Based System Technology in Geotechnical Engineering", *Proc. 2nd Panhellenic Conference in Geotechnical Engineering*, Thessaloniki, October, 1992.

44.   Mullarkey P.W. "A Geotechnical KBS Using Fuzzy Logic", in *Applications of A.I. in Engineering Problems*, 1st Int. Conf., Southampton University (ed. Sriram D. and Adey R.), Springer-Verlag, Vol. 2, pp. 847-859, 1986.

45.   Mullarkey P.W. and Fenves S.J. "Fuzzy logic in a geotechnical knowledge based system : CONE", *Civil Engineering Systems*, Vol 3, No. 2, pp. 58-81, 1986.

46.   Norbury D. R., Child G. H. and Spink T. W. "A Critical Review of Section 8 (BS5930)-Soil and Rock Description", in *Site Investigation Practice : Assessing*

BS5930, Geological Society, Engineering Geology Special Publication No.2, London, pp 331-342,1986.

47.   Norkin D.D. *Expert System for Geotechnical Site Characterisation*, MSc dissertation, Carnegie-Mellon University, Dept. of Civil Engineering, 1985.

48.   Northumbrian Water Authority, Mainsforth Stell Land Drainage Scheme, *Report on Site Investigation*, Report No. 006-701-296, April, 1982.

49.   Oliphant J. and Blockley D.I. "Knowledge Based System: Advisor on the Selection of Earth Retaining Structures", *Proc. 4th Int. Conf. A.I. techniques and applications for Civil and Structural Engineering Computing*, CIVIL-COMP Press, Edinburgh, 1989.

50.   Parikh S.A. and Kameswara Rao N.S.V. "An Expert System for Civil Engineering Applications", Proceedings of the Geotechnical Engineering Congress, in *Geotechnical Special Publication*, Vol. 1, No. 27, (ed. McLean F.G., Campbell DW. A. and Harris D.W.), ASCE, pp. 413-421, Colorado, 1991.

51.   Prolog Development Center *PDC Prolog Version 3.20 User's Guide*, Denmark, 1990.

52.   Prolog Development Center *PDC Prolog Version 3.20 Reference Guide*, Denmark, 1990.

53.   Prolog Development Center *PDC Prolog Version 3.20 Toolbox*, Denmark, 1990.

54.   Rashad M.M., Yehia N.A.B., Bazaraa A.S. and Dessouki A.I. "Foundcon : A Conceptual Model for the Integrated Knowledge-Based CAD Systems for Foundation Design", in *Artificial Intelligence and Civil Engineering* (ed. Topping B.H.V.), CIVIL-COMP Press, Edinburgh, pp. 125-135, 1991.

55.   Rehak D.R., Christiano P.P. and Norkin D.D. "SITECHAR: An Expert System Component of a Geotechnical Site Characterization Workbench", in *Applications of Knowledge-Based Systems to Engineering Analysis and Design* (ed. Dym C.L.), Am. Soc. Mech. Eng., New York, 1985.

56.   Rethati L. "Probabilistic Solutions in Geotechnics", in *Developments in Geotechnical Engineering*, Vol. 46, Budapest, 1988.

57.   Righetti G.A. and Cremonini M.G. "The DAISY Environment and the Expert System GUESS", in *Artificial Intelligence in Engineering : Diagnosis and Learning* (ed. Gero J.S.), Elsevier, Amsterdam, 1988.

58.   Rowlinson S. "Knowledge Based Systems: Potential in Design and Management", *Proc. 4th Int. Conf. A.I. techniques and applications for Civil and Structural Engineering Computing*, CIVIL-COMP Press, Edinburgh, pp. 21-26, 1989.

59.   Santamarina J.C. and Chameau J.L. "Expert Systems for Geotechnical Engineers", *Journal of Computing in Civil Engineering*, Vol. 1, 4, pp. 241-252, 1987.

60.   Shyu G.C. and Hryciw R.D. "SOLES: A Knowledge-Based Soil Liquefaction Potential Evaluation System", Proceedings of the Geotechnical Engineering Congress, in *Geotechnical Special Publication*, Vol. 1, No. 27, (ed. McLean F.G., Campbell DW. A. and Harris D.W.), ASCE, pp. 403-412, Boulder, Colorado, 1991.

61.   Siller J.T. "Expert Systems in Geotechnical Engineering", in *Expert Systems for Civil Engineers: Technology and Applications*, (ed. Maher M.L.), ASCE, New York, 1987.

62.   Smith I.G.N. and Oliphant J. "The Use of a Knowledge-Based System for Civil Engineering Site Investigations", in *Artificial Intelligence and Civil Engineering* (ed. Topping B.H.V.), CIVIL-COMP Press, Edinburgh, pp. 105-112, 1991.

63.   Smith I.G.N. and Oliphant J. "The Use of a Knowledge-Based System for Civil Engineering Site Investigations", (extended) in *Computer Systems in Engineering*, (to be published), 1992.

64.   Smith R.G. and Baker J.D. "DIPMETER ADVISOR System", *Proc. 8th Int. Joint Conf. A.I.*, pp. 122-129, Karlsruhe, Germany, 1983.

65.   Stuckrath L.A. and Grivas D.A. "A Knowledge-Based System for Bridge Foundation Selection", OECD Workshop on Knowledge-Based Expert Systems in Transportation Part 1, in *VTT Symposium (Valtion Teknillinen Tutkimuskeskus)*, Vol. 1, No. 116, pp. 283-297, Technical Research Center of Finland, Espoo, 1990.

66.   Terzaghi K. and Peck R.B. *Soil Mechanics in Engineering Practice*, Wiley and Sons, New York, 1967.

67.   Thomas P.R. *Geological Maps and Sections for Civil Engineers*, Blackie, Glascow and London, 1991.

68.   Toll D.G. "Representing the Ground", *Proc. NATO Advanced Study Institute: Optimisation and Decision Support Systems in Civil Engineering*, Heriot-Watt University, Vol II, 1989.

69.   Toll D. G. "Do Geotechnical Engineers Need Expert Systems", in *Ground Engineering*, pp. 32-36, April, 1990.

70.   Toll D.G. and Attewell P.B. "Use of an Expert System in Site Investigation", in *Proc. 6th International Congress of International Association Engineering Geologists*, Vol. 1, pp. 767-773, Amsterdam, August, 1990.

71.   Toll D.G., Moula M., Oliver A. and Vaptismas N. "A Knowledge-Based System for Interpreting Site Investigation Information", *Proc. International Conference in Geotechnics and Computers*, pp. 607-614, Paris, 1992.

72.   Toll D.G., Moula M. and Vaptismas N. "Representing the Engineering Description of Soils in Knowledge Based Systems", AIENG 6, *Proc. 6th Int. Conf. on Applications of Artificial Intelligence in Engineering*, University of Oxford, July 1991.

73.   Weltman A. J. and Head J. M. *Site Investigation Manual*, Construction Industry Research and Information Association, London, 1983.

74.   Wislocki A.P. and Bentley S.P. "An Expert System for Landslide Hazard and Risk Assessment", *Proc. 4th Int. Conf. A.I. techniques and applications for Civil and Structural Engineering Computing*, CIVIL-COMP Press, pp. 249-252, Edinburgh, 1989.

75.   Wong K.C., Poulos H.G. and Thorne C.P. "Development of Expert Systems for Pile Foundation Design", *Trans. Inst. of Engineers*, Vol. CE33, No. 2, pp. 119-127, 1991.

76.   Wong K.C., Poulos H.G. and Thorne C.P. "Site Classification by Expert Systems", *Computers and Geotechnics*, 8, pp. 133-156, 1989.

77.   Yehia N.A.B. and El-Hajj A.H. "A Knowledge Based Approach for the Design of Spread Footings", in *Application of A.I. techniques to Civil and Structural Engineering* (ed. Topping B.H.V.), CIVIL-COMP Press, pp. 119-124, Edinburgh, 1987.

78.   Zheng H., Mikroudis G.K., Pamukcu S. and Hu Z.X. "BABE: An Expert System for Structural Design of Bridge Abutments and Piers", in *Computer Utilization in Structural Engineering*, Proceedings of the sessions at Structures Congress '89, ASCE, pp. 372-381, New York, 1989.

APPENDIX A

PARSER and VASIC MODULES

code =5000

include "\\pharpro\\programs\\tdoms.pro"
include "\\pharpro\\programs\\tpreds.pro"
include "\\pharpro\\programs\\menu2.pro"

domains

list   = symbol*
reallist   = real*

predicates

allow(string,string,list)
append(integerlist,integerlist,integerlist)
append(list,list,list)
append(reallist,reallist,reallist)
assign_col(list,list)
assign_cons(list)
assign_soil(list,list,integerlist,integerlist)
assign_values
becoming(list)
bed_or_spacing(integerlist,integerlist,integerlist)
bed_spacing(list)
bedding(list)
calc_col(integerlist,integerlist,integerlist,integerlist,integerlist,integerlist,integerlist)
calc_hue(integerlist,integerlist)
calc_lum(integerlist,integerlist)
calc_sat(integerlist,integerlist)
calc_sat_lum_hue(integerlist,integerlist)
calc_sim(integerlist,integerlist,reallist)
choose(integer)
col_similarity(reallist,integerlist,reallist,reallist)
colour(list)
colour_hue(symbol,integerlist)
combine(string,string,list,list,list,symbol,symbol)
compare
compare_bedding(list,list,integerlist)
compare_colour(integerlist,integerlist,reallist)
compare_cons(integerlist,integerlist,integerlist)

compare_hue(reallist,reallist,reallist)
compare_lum(reallist,reallist,reallist)
compare_sat(integerlist,integerlist,integerlist)
compare_soils(list,list,list,list,reallist)
compare_spacing(list,list,integerlist)
cons(symbol,integer)
consist(string,string,symbol,symbol)
consistency(symbol,list)
continuation(list)
create_sim_list(integerlist,integerlist,integerlist,integerlist)
derivative(symbol,list,symbol)
discontinuity(list)
distribution(list)
ending(symbol,list)
enter(list,list,list,list,list,list,list,list)
final_sim(reallist,reallist,reallist,reallist,reallist)
find_max(real,real,real)
find_similarity(integerlist,integer,integer)
first(list,symbol)
give_values(list,integerlist)
grading(list)
insert(symbol,symbol,symbol)
instr(string,string)
instr(symbol,symbol)
last(list,symbol)
last_elem(list,list,list)
layer_structure(List)
length_list(integerlist,integer,integer)
length_list(list,integer,integer)
make_list(list,list,list,list,list,list,list,list,list)
member(integer,integerlist)
member(symbol,list)
mod_lum(integerlist,list,integerlist)
modifier(symbol,symbol,list)
moisture(list)
more(list)
mottle(list)
overall_sim(reallist,reallist,integerlist,integerlist)
parse
parse(symbol,string,string,list,list,list,list,list,list,list,list,list,list,list,list,list,list)
per_clay(integerlist,integerlist)
percents(integerlist,integerlist)
plasticity(list)
process_col(list,list,integerlist)
process_cons(list,integerlist)
process_other(list)
process_soil(list,integerlist)
process_structure(list,list)
psd(list,list,list,list,integerlist)
psd1(list,list,list,list,integerlist,integerlist)

psd2(list,list,list,list,integerlist,integerlist)
put_percent(list,integer,list,integerlist,integerlist,integerlist)
range(list)
require(list,string,string,list,symbol)
reverse(integerlist,integerlist)
reverse(list,list)
separate(list,list,list,list,list)
shape(list)
size(list)
soil(string,string,list,list,list,list,list,list,list,list)
soil_type(symbol,list)
spacing(list)
split_str(string,char,string,string,string)
str_spacing(symbol,symbol,integer)
structure(list)
sum_percents(integerlist,integer,integerlist,integerlist)
texture(list)
weathering(list)
write_list(symbol,symbol,list)
write_list(symbol,symbol,integerlist)
write_list(symbol,symbol,reallist)
write_out_list(list,integerlist)
write_soil(symbol,symbol,symbol)
write_soil(symbol,symbol,integer)

/*******STANDARD CLAUSES*******/

clauses

instr(Item,Itemstring) :-
        concat(Item,_,Itemstring).
instr(Item,Itemstring) :-
        concat(_,Item,Itemstring).

member(Name,[Name|_]).
member(Name,[_|Tail]) :-
        member(Name,Tail).

append([],List,List).
append([X|L1],List2,[X|L3]) :-
        append(L1,List2,L3).

first([First|_],First).

last([Last],Last).
last([_|Tail],Last) :-
        last(Tail,Last).

A3

```
reverse([],[]).
reverse([Head|Tail],List) :-
        reverse(Tail,Result),
        append(Result,[Head],List).


split_str(String,Char,Work,Front,Back) :-
        frontchar(Work,Char,Back),
        concat(Front,Work,String).
split_str(String,Char,Work,Front,Back) :-
        frontchar(Work,_,Rest),
        split_str(String,Char,Rest,Front,Back).



/*******DEFINITIONS*******/



becoming([becoming]).
bed_spacing([thick,thin]).
bedding([bedded,laminated]).
colour([red,pink,purple,yellow,brown,green,blue,white,grey,black]).
colour_hue(red,[240,120,0]).
colour_hue(pink,[240,180,230]).
colour_hue(purple,[240,60,200]).
colour_hue(yellow,[240,120,40]).
colour_hue(brown,[240,60,20]).
colour_hue(green,[240,120,80]).
colour_hue(blue,[240,120,160]).
colour_hue(white,[-240,240,-240]).
colour_hue(grey,[0,180,-240]).
colour_hue(black,[-240,0,-240]).
consistency(cohesive,[soft,firm,stiff,hard]).
consistency(silt,[loose,compact]).
consistency(granular,[loose,dense]).
consistency(organic,[spongy,firm]).
cons("very loose",2).
cons("loose",7).
cons("medium dense",20).
cons("dense",40).
cons("very dense",52).
cons("moderately compact",12).
cons("compact",22).
cons("very compact",40).
cons("very soft",1).
cons("soft",3).
cons("firm",6).
cons("stiff",12).
cons("very stiff",20).
continuation(["and",",",with,of,to]).
discontinuity([fissures,joints]).
distribution([poor,well,gap,uniform]).
```

ending(adjective,["ish","y","ly"]).
ending(verb,["d"]).
ending(noun,["ing","ion","ity"]).
grading([grade]).
layer_structure([interbedded,interstratified,inclusions,lenses,pockets]).
modifier(none, plasticity,[none,no,non,un]).
modifier(low, plasticity,[low]).
modifier(medium, plasticity,[medium,intermediate]).
modifier(high, plasticity,[high]).
modifier(minor, soil_before,[slightly]).
modifier(secondary, soil_before,[]).
modifier(major, soil_before,[very]).
modifier(minor, soil_after,[occasional,little,trace,scatter,infrequent,isolate]).
modifier(secondary, soil_after,[some,few]).
modifier(major, soil_after,[numerous,many,frequent]).
modifier(low, colour,[pale,light]).
modifier(high, colour,[dark]).
modifier(extreme,colour,[bright]).
moisture([dry,dessicated,moist,damp,wet]).
more([more]).
mottle([mottle,patch,blotch]).
percents([1,0,0,0],[100,0,0,0]).
percents([2,0,0,0],[50,0,0,0]).
percents([3,0,0,0],[33,0,0,0]).
percents([4,0,0,0],[25,0,0,0]).
percents([1,1,0,0],[75,25,0,0]).
percents([1,0,1,0],[90,0,10,0]).
percents([1,0,0,1],[97,0,0,3]).
percents([1,1,1,0],[65,25,10,0]).
percents([1,1,0,1],[72,25,0,3]).
percents([1,1,2,0],[55,25,10,0]).
percents([2,0,0,1],[48,0,0,4]).
percents([2,0,1,0],[45,0,10,0]).
percents([2,1,0,0],[37,26,0,0]).
percents([1,0,0,2],[94,0,0,3]).
percents([1,0,1,1],[87,0,10,3]).
percents([1,0,2,0],[80,0,10,0]).
percents([1,2,0,0],[50,25,0,0]).
percents([1,0,0,3],[91,0,0,3]).
percents([1,0,1,2],[84,0,10,3]).
percents([1,1,0,2],[69,25,0,3]).
percents([1,0,3,0],[70,0,10,0]).
percents([1,0,2,1],[77,0,10,3]).
percents([1,3,0,0],[40,20,0,0]).
percents([1,2,0,1],[47,25,0,3]).
percents([1,2,1,0],[40,25,10,0]).
percents([1,1,1,1],[62,25,10,3]).
per_clay([1,0,0,0],[100,0,0,0]).
per_clay([2,0,0,0],[50,0,0,0]).
per_clay([1,1,0,0],[35,65,0,0]).

per_clay([1,0,1,0],[50,0,50,0]).
per_clay([1,0,0,1],[65,0,0,35]).
per_clay([1,1,1,0],[35,40,25,0]).
per_clay([1,1,0,1],[35,45,0,20]).
per_clay([1,1,2,0],[35,25,20,0]).
per_clay([2,0,0,1],[40,0,0,20]).
per_clay([2,0,1,0],[35,0,30,0]).
per_clay([2,1,0,0],[35,30,0,0]).
per_clay([1,0,0,2],[50,0,0,25]).
per_clay([1,0,1,1],[40,0,35,25]).
per_clay([1,0,2,0],[40,0,30,0]).
per_clay([1,2,0,0],[36,32,0,0]).
per_clay([1,0,1,2],[40,0,30,15]).
per_clay([1,1,0,2],[35,35,0,15]).
per_clay([1,0,2,1],[35,0,25,15]).
per_clay([1,2,0,1],[35,30,0,5]).
per_clay([1,2,1,0],[35,25,15,0]).
per_clay([1,1,1,1],[35,35,20,10]).
plasticity([plastic,cohesive,plasticity,lean,fat,light,heavy]).
range(["and","-",to,becoming]).
shape([rounded,subrounded,subangular,angular,elongate,flat,spherical,tabular]).
size([fine,medium,coarse]).
soil_type(inorganic,[clay,silt,sand,gravel,cobbles,boulders]).
soil_type(organic,[peat]).
spacing([wide,close]).
str_spacing("very thickly bedded","over 2000",100).
str_spacing("thickly bedded","2000 - 600",83).
str_spacing("medium bedded","600 - 200",66).
str_spacing("thinly bedded","200 - 60",49).
str_spacing("very thinly bedded","60 - 20",32).
str_spacing("thickly laminated","20 - 6",15).
str_spacing("thinly laminated","under 6",0).
str_spacing("very widely spaced","over 2000",100).
str_spacing("widely spaced","2000 - 600",80).
str_spacing("medium spaced","600 - 200",60).
str_spacing("closely spaced","200 - 60",40).
str_spacing("very closely spaced ","60 - 20",20).
str_spacing("extremely closely spaced","under 20",0).
structure([interbedded,interstratified]).
texture([rough,smooth,polished]).
weathering([fresh,weathered]).

```
/*******PARSER MODULE*******/


parse:-
        enter(Soil_list,Cons_list,Col_list,Bed_list,Spacing_list,Mod_list,
                            Layer_str_list,Other_list),
        write(" Soil list : ",Soil_list),nl,
        write(" Layer Structure : ",Layer_str_list),nl,
        write(" Consistency list : ",Cons_list),nl,
        write(" Colour list : ",Col_list),nl,
        write(" Colour modifier : ",Mod_list),nl,
        write(" Bedding list : ",Bed_list),nl,
        write(" Spacing list : ",Spacing_list),nl,nl,
        write(" Additional Info : ",Other_list),nl,nl.


enter(Soil_list,Cons_list,Number_list,Bed_list,Spacing_list,Mod_list,
                            Layer_str_list,Other_list):-
        write("Enter soil description :"),
        readln(String),nl,
        soil(String,"",Soil_list,Cons_list,Number_list,Bed_list,
                            Spacing_list,Mod_list,Layer_str_list,Other_list).


soil(Stringin,Stringout,Soil_list,Cons_list,Col_list,Bed_list,
                            Spacing_list,Mod_list,Layer_str_list,Other_list):-!,
        parse(main,Stringin,String1,[],Soil_list,[],Cons_list,[],Col_list,
                            [],Bed_list,[],Spacing_list,[],Mod_list,[],Other_list),
        parse(second_main,String1,Stringout,[],Layer_str_list,[],
                            [],[],[],[],[],[],[],[],[],[]).


consist(Stringin,Stringout,Type,Consist):-
        consistency(Type,List),
        combine(Stringin,Stringout,[],List,[],_,Consist).
consist(Stringin,Stringout,Type,Consist):-
        consistency(Type,List),
        combine(Stringin,Stringout,[very,medium,moderately],List,[],Token1,
                            Token2),
        concat(Token1," ",X),
        concat(X,Token2,Consist).


parse(_,"","",List_out,List_out,Cons_out,Cons_out,Number_list_out,
                            Number_list_out,Bed_out,Bed_out,Spacing_out,
                                    Spacing_out,Mod_out,Mod_out,Other_out,
                                            Other_out).
parse(_,Stringin,Stringin,List_out,List_out,Cons_out,Cons_out,Number_list_out,
                            Number_list_out,Bed_out,Bed_out,Spacing_out,
                                    Spacing_out,Mod_out,Mod_out,Other_out,
                                            Other_out):-
        layer_structure(List),
        fronttoken(Stringin,Token,_),
```

A7

```prolog
        member(Token,List),!.
parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                    Number_temp_list,Number_list_out,Bed_temp,
                    Bed_out,Spacing_temp,Spacing_out,
                    Mod_temp,Mod_out,Other_temp,
                    Other_out):-
        continuation(List),
        require([],Stringin,String1,List,_),
        parse(Level,String1,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                    Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                    Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                    Other_temp,Other_out).
parse(Level,Stringin,Stringout,List_in,List_out,Cons_temp,Cons_out,
                    Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                    Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                    Other_temp,Other_out):-
        layer_structure(List),
        require([],Stringin,String1,List,Layer_structure),
        append([Layer_structure],List_in,List_temp),
        parse(Level,String1,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                    Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                    Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                    Other_temp,Other_out).
parse(Level,Stringin,Stringout,List_in,List_out,Cons_temp,Cons_out,
                    Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                    Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                    Other_temp,Other_out):-
        soil_type(_,List),
        require([],Stringin,String1,List,Soil_type),
        append([Level,Soil_type],List_in,List_temp),
        parse(Level,String1,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                    Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                    Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                    Other_temp,Other_out).
parse(Level,Stringin,Stringout,List_in,List_out,Cons_temp,Cons_out,
                    Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                    Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                    Other_temp,Other_out):-
        modifier(Amount,soil_before,List1),
        soil_type(_,List2),
        combine(Stringin,String1,List1,List2,[y],_,Soil_type),
        append([Amount,Soil_type],List_in,List_temp),
        parse(Level,String1,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                    Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                    Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                    Other_temp,Other_out).
parse(_,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                    Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                    Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                    Other_temp,Other_out):-
```

```
                allow(Stringin,String1,[]),
                modifier(Amount,soil_after,List2),
                require([],String1,String2,List2,_),
                parse(Amount,String2,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                        Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                                Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                        Other_temp,Other_out).
parse(Level,Stringin,Stringout,List_temp,List_out,Cons_in,Cons_out,
                        Number_list_in,Number_list_out,Bed_temp,Bed_out,
                                Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                        Other_temp,Other_out):-
                consist(Stringin,String1,Type,Consist1),
                range(List2),
                require([],String1,String2,List2,_),
                consist(String2,String3,Type,Consist2),
                cons(Consist1,_),
                cons(Consist2,_),
                append([Consist1,Consist2],Cons_in,Cons_temp),
                parse(Level,String3,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                        Number_list_in,Number_list_out,Bed_temp,Bed_out,
                                Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                        Other_temp,Other_out).
parse(Level,Stringin,Stringout,List_temp,List_out,Cons_in,Cons_out,
                        Number_list_in,Number_list_out,Bed_temp,Bed_out,
                                Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                        Other_temp,Other_out):-
                consist(Stringin,String1,_,Consist),
                cons(Consist,_),
                append([Consist],Cons_in,Cons_temp),
                parse(Level,String1,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                        Number_list_in,Number_list_out,Bed_temp,Bed_out,
                                Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                        Other_temp,Other_out).
parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                        Number_list_in,Number_list_out,Bed_temp,Bed_out,
                                Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                        Other_temp,Other_out):-
                colour(List1),
                require([],Stringin,String1,List1,Colour1),
                range(List2),
                require([],String1,String2,List2,_),
                require([],String2,String3,List1,Colour2),
                append([Level,main,Colour1,Colour2],Number_list_in,Number_list_temp),
                parse(Level,String3,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                        Number_list_temp,Number_list_out,Bed_temp,Bed_out,
                                Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                        Other_temp,Other_out).
```

```
parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                Number_list_in,Number_list_out,Bed_temp,Bed_out,
                        Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_temp,Other_out):-
        colour(List),
        require([],Stringin,String1,List,Colour),
        append([Level,main,Colour],Number_list_in,Number_list_temp),
        parse(Level,String1,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                Number_list_temp,Number_list_out,Bed_temp,Bed_out,
                        Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_temp,Other_out).
parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                Number_list_temp,Number_list_out,Bed_temp,Bed_out,
                        Spacing_temp,Spacing_out,Mod_in,Mod_out,
                                Other_temp,Other_out):-
        modifier(_,colour,List),
        require([],Stringin,String1,List,Mod),
        append([Level,Mod],Mod_in,Mod_temp),
        parse(Level,String1,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                Number_list_temp,Number_list_out,Bed_temp,Bed_out,
                        Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_temp,Other_out).
parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                Number_list_temp,Number_list_out,Bed_temp,Bed_out,
                        Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_temp,Other_out):-
        colour(List),
        require([ish,y],Stringin,String1,List,Colour),
        append([Level,secondary,Colour],Number_list_temp,Number_temp_list),
        parse(Level,String1,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                        Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_temp,Other_out).
parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                Number_temp_list,Number_list_out,Bed_in,Bed_out,
                        Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_temp,Other_out):-
        bed_spacing(List1),
        combine(Stringin,String1,[very],List1,[ly],Token1,Token2),
        concat(Token1," ",X),
        concat(Token2,"ly ",Bed),
        concat(X,Bed,Bed1),
        bedding(List2),
        require([],String1,String2,List2,Token3),
        concat(Bed1,Token3,Bedding),
        append([Bedding],Bed_in,Bed_temp),
        parse(Level,String2,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                        Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_temp,Other_out).
```

```prolog
parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                Number_temp_list,Number_list_out,Bed_in,Bed_out,
                    Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                        Other_temp,Other_out):-
        bed_spacing(List1),
        require([ly],Stringin,String1,List1,Bed),
        concat(Bed,"ly ",Bed1),
        bedding(List2),
        require([],String1,String2,List2,Token),
        concat(Bed1,Token,Bedding),
        append([Bedding],Bed_in,Bed_temp),
        parse(Level,String2,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                    Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                        Other_temp,Other_out).
parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                Number_temp_list,Number_list_out,Bed_in,Bed_out,
                    Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                        Other_temp,Other_out):-
        require([],Stringin,String1,[medium],Bed),
        concat(Bed," ",Bed1),
        bedding(List2),
        require([],String1,String2,List2,Token),
        concat(Bed1,Token,Bedding),
        append([Bedding],Bed_in,Bed_temp),
        parse(Level,String2,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                    Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                        Other_temp,Other_out).
parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                    Spacing_in,Spacing_out,Mod_temp,Mod_out,
                        Other_temp,Other_out):-
        spacing(List1),
        discontinuity(List2),
        combine(Stringin,String1,[very,extremely],List1,[ly],Token1,Token2),
        concat(Token1," ",X),
        concat(Token2,"ly ",Space),
        concat(X,Space,Space1),
        require([],String1,String2,[spaced],Token3),
        concat(Space1,Token3,Spacing),
        require([],String2,String3,List2,Token4),
        append([Token4,Spacing],Spacing_in,Spacing_temp),
        parse(Level,String3,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                    Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                        Other_temp,Other_out).
```

parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                    Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                        Spacing_in,Spacing_out,Mod_temp,Mod_out,
                            Other_temp,Other_out):-
            spacing(List1),
            discontinuity(List2),
            require([ly],Stringin,String1,List1,Space),
            concat(Space,"ly ",Space1),
            require([],String1,String2,[spaced],Token),
            concat(Space1,Token,Spacing),
            require([],String2,String3,List2,Token1),
            append([Token1,Spacing],Spacing_in,Spacing_temp),
            parse(Level,String3,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                        Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                            Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_temp,Other_out).
parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                    Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                        Spacing_in,Spacing_out,Mod_temp,Mod_out,
                            Other_temp,Other_out):-
            discontinuity(List),
            require([],Stringin,String1,[medium],Space),
            concat(Space," ",Space1),
            require([],String1,String2,[spaced],Token),
            concat(Space1,Token,Spacing),
            require([],String2,String3,List,Token1),
            append([Token1,Spacing],Spacing_in,Spacing_temp),
            parse(Level,String3,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                        Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                            Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_temp,Other_out).
parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                    Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                        Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                            Other_in,Other_out):-
            size(List1),
            require([],Stringin,String1,List1,Size1),
            range(List2),
            require([],String1,String2,List2,_),
            require([],String2,String3,List1,Size2),
            append([Level,lower_size,Size1,upper_size,Size2],Other_in,Other_temp),
            parse(Level,String3,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                        Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                            Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_temp,Other_out).
parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                    Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                        Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                            Other_in,Other_out):-
            size(List),

```
                require([],Stringin,String1,List,Size),
                append([Level,size,Size],Other_in,Other_temp),
                parse(Level,String1,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                        Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                            Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_temp,Other_out).
        parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                        Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                            Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_in,Other_out):-
                modifier(Plasticity,plasticity,List1),
                require([],Stringin,String1,List1,_),
                plasticity(List2),
                require([],String1,String2,List2,_),
                append([Level,plasticity,Plasticity],Other_in,Other_temp),
                parse(Level,String2,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                        Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                            Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_temp,Other_out).
        parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                        Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                            Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_in,Other_out):-
                plasticity(List),
                require([],Stringin,String1,List,_),
                append([Level,plasticity,medium],Other_in,Other_temp),
                parse(Level,String1,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                        Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                            Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_temp,Other_out).
        parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                        Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                            Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_in,Other_out):-
                shape(List),
                require([],Stringin,String1,List,Shape),
                append([Level,shape,Shape],Other_in,Other_temp),
                parse(Level,String1,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                        Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                            Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_temp,Other_out).
        parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                        Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                            Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_in,Other_out):-
                texture(List),
                require([],Stringin,String1,List,Texture),
                append([Level,texture,Texture],Other_in,Other_temp),
                parse(Level,String1,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                        Number_temp_list,Number_list_out,Bed_temp,Bed_out,
```

```prolog
                                    Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                    Other_temp,Other_out).
parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
            Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                    Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                            Other_in,Other_out):-
        distribution(List1),
        require(["",ly],Stringin,String1,List1,Distribution),
        grading(List2),
        allow(String1,String2,List2),
        append([Level,grading,Distribution],Other_in,Other_temp),
        parse(Level,String2,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                        Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                                Other_temp,Other_out).
parse(Level,Stringin,Stringout,List_temp,List_out,Cons_temp,Cons_out,
            Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                    Spacing_temp,Spacing_out,Mod_temp,Mod_out,
                            Other_temp,Other_out):-
        fronttoken(Stringin,Token,String1),
        write("Ignoring : ",Token),nl,
        parse(Level,String1,Stringout,List_temp,List_out,Cons_temp,Cons_out,
                Number_temp_list,Number_list_out,Bed_temp,Bed_out,
                        Spacing_temp,Spacing_out,Mod_temp,
                                Mod_out,Other_temp,Other_out).


write_soil(Level,Type,Soil_type):-
        write(Level," : ",Type," : ",Soil_type),nl.
write_list(Level,Type,List_out):-
        write(Level," : ",Type," : ",List_out),nl.


require(Ending,Stringin,Stringout,List,Member):-
        fronttoken(Stringin,Token,Stringout),
        member(Member,List),
        derivative(Member,Ending,Token).


allow(Stringin,Stringout,List):-
        ending(_,End_List),
        require(End_List,Stringin,Stringout,List,_).
allow(Stringin,Stringout,[]):-
        fronttoken(Stringin,_,Stringout).
allow(String,String,_).


combine(Stringin,Stringout,List1,List2,Ending,Token1,Token2):-
        require([],Stringin,String1,List1,Token1),
        require(Ending,String1,Stringout,List2,Token2).
combine(Stringin,Stringout,[],List2,Ending,"",Token2):-
        require(Ending,Stringin,Stringout,List2,Token2).
```

```prolog
derivative(Noun,Ending,Result):-
        member(End,Ending),
        insert(Noun,End,Result).
derivative(Noun,[],Noun).

insert(Noun,Ending,Result):-
        member(X,[d,l]),
        concat(_,X,Noun),
        concat(Noun,X,New),
        concat(New,Ending,Result).
insert(Noun,Ending,Result):-
        concat(Noun,"e",New),
        concat(New,Ending,Result).
insert(Noun,Ending,Result):-
        concat(Noun,Ending,Result).
```

/****VALUES ASSIGNMENT and SIMILARITY CALCULATION MODULE****/

```prolog
assign_values:-
        enter(Soil_list,Cons_list,Col_list,Bed_list,Spacing_list,Mod_list,
                                Layer_str_list,Other_list),
        assign_soil(Soil_list,Layer_str_list,_,_),
        assign_cons(Cons_list),
        assign_col(Col_list,Mod_list),
        process_structure(Bed_list,Spacing_list),nl,
        process_other(Other_list),nl.

assign_soil(Soil_list,[],Psd,[]):-!,
        process_soil(Soil_list,Psd),
        write_out_list([clay,silt,sand,gravel,cobbles,boulders],Psd),nl.
assign_soil(Soil_list,Layer_str_list,Psd,Psdd):-
        last_elem(Layer_str_list,[Element],Layer_list),
        process_soil(Soil_list,Psd),
        process_soil(Layer_list,Psdd),
        write_out_list([clay,silt,sand,gravel,cobbles,boulders],Psd),nl,
        write("Layer Structure term : ",Element),nl,nl,
        write_out_list([clay,silt,sand,gravel,cobbles,boulders],Psdd),nl.

last_elem([],[],[]).
last_elem(List,Last,List1):-
        append(List1,Last,List),
        length_list(Last,0,1).


assign_cons([]):-!.
assign_cons(Cons_list):-
        process_cons(Cons_list,Consistency),
```

```
                write("Estimated N value : ",Consistency),nl.

assign_col([],[]):-!.
assign_col(Col_list,Mod_list):-
        process_col(Col_list,Mod_list,Final_list),
        write("Colour values : ",Final_list),nl.

process_soil([],[]).
process_soil(List,Psd):-
        make_list(List,[],Main_list,[],Major_list,[],Sec_list,[],Minor_list),
        psd(Main_list,Major_list,Sec_list,Minor_list,Psd).

process_col([],[],[]).
process_col(List,Mod_list,Final_list):-
        separate(List,[],Mainout,[],_),
        give_values(Mainout,Val_list),
        mod_lum(Val_list,Mod_list,Final_list).

mod_lum(List,[],List).
mod_lum(List,[A,_],List):-
        not(A=main).
mod_lum([A,B,C],[main,dark],[A,B1,C]):-
        B1=round(0.8*B).
mod_lum([A,B,C],[main,light],[A,B1,C]):-
        B2=1.1*B,B1=round(B2).

give_values([main,Colour],List1):-
        colour_hue(Colour,List),
        reverse(List,List1).
give_values([main,Colour1,main,Colour2],List):-
        colour_hue(Colour1,[A1,A2,A3]),
        colour_hue(Colour2,[B1,B2,B3]),
        C1=(A1+B1)/2,C2=(A2+B2)/2,C3=(A3+B3)/2,
        List=[C1,C2,C3].
give_values([secondary,Colour2,main,Colour1],List):-
        colour_hue(Colour1,[A1,A2,A3]),
        colour_hue(Colour2,[B1,B2,B3]),
        calc_sat_lum_hue([A1,A2,A3,B1,B2,B3],List).

calc_col(Number_list,Sat_list_temp,Sat_list_out,Lum_list_temp,Lum_list_out,
                            Hue_list_temp,Hue_list_out):-
        Number_list=[Head|Tail],
        append([Head],Sat_list_temp,Sat_list_in),
        Tail=[H|T],
        append([H],Lum_list_temp,Lum_list_in),
        T=[H1|T1],
        append([H1],Hue_list_temp,Hue_list_in),
        calc_col(T1,Sat_list_in,Sat_list_out,Lum_list_in,Lum_list_out,
                            Hue_list_in,Hue_list_out).
calc_col([],Sat_list_out,Sat_list_out,Lum_list_out,Lum_list_out,
```

```
calc_sat([],Saturation):-
        Saturation=[].
calc_sat(Sat_list,Saturation):-
        Sat_list=[A],Saturation=[A];
        Sat_list=[240,240],Saturation=[240];
        Sat_list=[0,0],Saturation=[0];
        Sat_list=[0,240],Saturation=[168];
        Sat_list=[240,0],Saturation=[72];
        Sat_list=[A,-240],Saturation=[A];
        Sat_list=[-240,A],Saturation=[A].


calc_lum([],[]).
calc_lum([0,A],Luminence):-
        Lum=round(0.5*A),Luminence=[Lum].
calc_lum([240,A],Luminence):-
        Lum=round(0.5*(240+A)),Luminence=[Lum].
calc_lum(Lum_list,Luminence):-
        Lum_list=[A,B],
        Lum=round(0.25*A+0.75*B),Luminence=[Lum];
        Lum_list=[A],Luminence=[A].


calc_hue([],Hue):-
        Hue=[].
calc_hue(Hue_list,Hue):-
        Hue_list=[A],Hue=[A];
        Hue_list=[-240,A],Hue=[A];
        Hue_list=[A,-240],Hue=[A];
        Hue_list=[A,B],
        abs(A-B)<=120,
        H=round(0.25*A+0.75*B),Hue=[H];
        Hue_list=[A,B],
        A-B>120,C=240-A+B,
        Hue0=round(A+0.75*C),Hue0<=240,
        Hue=[Hue0];
        Hue_list=[A,B],
        A-B>120,C=240-A+B,
        Hue0=A+0.75*C,Hue0>240,
        H=round(Hue0-240),Hue=[H];
        Hue_list=[B,A],
        A-B>120,C=240-A+B,
        Hue0=round(A+0.75*C),Hue0<=240,
        Hue=[Hue0];
        Hue_list=[B,A],
        A-B>120,C=240-A+B,
        Hue0=A+0.75*C,Hue0>240,
        H=round(Hue0-240),Hue=[H].


calc_sat_lum_hue([],[]).
```

A17

```prolog
calc_sat_lum_hue(Number_list,List):-
        calc_col(Number_list,[],Sat_list,[],Lum_list,[],Hue_list),
        calc_sat(Sat_list,[Saturation]),
        calc_lum(Lum_list,[Luminance]),
        calc_hue(Hue_list,[Hue]),
        List=[Hue,Luminance,Saturation].

separate([],Mainout,Mainout,Lesserout,Lesserout).
separate([medium,Amount,Colour|T],Mainin,Mainout,Lesserin,Lesserout):-
        append([Amount,Colour],Lesserin,Lessertemp),
        separate(T,Mainin,Mainout,Lessertemp,Lesserout).
separate([main,Amount,Colour|T],Mainin,Mainout,Lesserin,Lesserout):-
        append([Amount,Colour],Mainin,Maintemp),
        separate(T,Maintemp,Mainout,Lesserin,Lesserout).

process_cons([],[]).
process_cons([A],[Value]):-
        cons(A,Value).
process_cons([A,B],[Value]):-
        cons(A,Value1),
        cons(B,Value2),
        Value=abs(Value1+Value2)/2.

process_structure([],[]).
process_structure([A],[]):-
        str_spacing(A,Range,_),
        write("Bedding spacing range (mm): ",Range).
process_structure([],[A,B]):-
        str_spacing(B,Range,_),
        write("Spacing range of ",A,"(mm): ",Range).

process_other([]).
process_other(List):-
        write("Additional Info : ",List),nl,nl.

length_list([],Length,Length).
length_list([_|Tail],Len,Length):-
        Len1=Len+1,
        length_list(Tail,Len1,Length).

psd(Main,Major,Sec,Minor,Psd):-
        member(clay,Main),!,
        psd2(Main,Major,Sec,Minor,_,Psd).

psd(Main,Major,Sec,Minor,Psd):-
        member(silt,Main),!,
        psd2(Main,Major,Sec,Minor,_,Psd).
psd(Main,Major,Sec,Minor,Psd):-
        psd1(Main,Major,Sec,Minor,_,Psd).
```

```prolog
psd1(Main,Major,Sec,Minor,Particle_size,Psd):-
        Definitions=[clay,silt,sand,gravel,cobbles,boulders],
        Init_percent=[0,0,0,0,0,0],
        Particle_size=[0,150,300,450,600,750,900],
        length_list(Main,0,Len_main),
        length_list(Major,0,Len_major),
        length_list(Sec,0,Len_sec),
        length_list(Minor,0,Len_minor),
        percents([Len_main,Len_major,Len_sec,Len_minor],
                        [Per_main,Per_major,Per_sec,Per_minor]),
        put_percent(Main,Per_main,Definitions,Init_percent,[],Psd1),
        put_percent(Major,Per_major,Definitions,Psd1,[],Psd2),
        put_percent(Sec,Per_sec,Definitions,Psd2,[],Psd3),
        put_percent(Minor,Per_minor,Definitions,Psd3,[],Psd).

psd2(Main,Major,Sec,Minor,Particle_size,Psd):-
        Definitions=[clay,silt,sand,gravel,cobbles,boulders],
        Init_percent=[0,0,0,0,0,0],
        Particle_size=[0,150,300,450,600,750,900],
        length_list(Main,0,Len_main),
        length_list(Major,0,Len_major),
        length_list(Sec,0,Len_sec),
        length_list(Minor,0,Len_minor),
        per_clay([Len_main,Len_major,Len_sec,Len_minor],
                        [Per_main,Per_major,Per_sec,Per_minor]),
        put_percent(Main,Per_main,Definitions,Init_percent,[],Psd1),
        put_percent(Major,Per_major,Definitions,Psd1,[],Psd2),
        put_percent(Sec,Per_sec,Definitions,Psd2,[],Psd3),
        put_percent(Minor,Per_minor,Definitions,Psd3,[],Psd).

write_out_list([],[]).
write_out_list([Head|Definitions],[Value|Psd]):-
        write(Head,": ",Value),nl,
        write_out_list(Definitions,Psd).

put_percent(_,_,[],[],Percent_temp,Percent_out):-
        reverse(Percent_temp,Percent_out).
put_percent(Soil_types,Percentage,[Definition|Def_tail],[_|Per_tail],
                                        Percent_temp,Percent_out):-
        member(Definition,Soil_types),
        append([Percentage],Percent_temp,Percent_new),
        put_percent(Soil_types,Percentage,Def_tail,Per_tail,Percent_new,
                        Percent_out).
put_percent(Soil_types,Percentage,[Definition|Def_tail],[Percent|Per_tail],
                                        Percent_temp,Percent_out):-
        not(member(Definition,Soil_types)),
        append([Percent],Percent_temp,Percent_new),
        put_percent(Soil_types,Percentage,Def_tail,Per_tail,Percent_new,
                        Percent_out).
```

```prolog
make_list([],A,A,B,B,C,C,D,D).
make_list(List_out,Main_list_in,Main_list,High_list_temp,High_list,
                    Medium_list_temp,Medium_list,Low_list_temp,Low_list):-
      List_out=["main",Soil_type|Tail],
      append([Soil_type],Main_list_in,Main_list_temp),
      make_list(Tail,Main_list_temp,Main_list,High_list_temp,High_list,
                    Medium_list_temp,Medium_list,Low_list_temp,Low_list).
make_list(List_out,Main_list_temp,Main_list,High_list_in,High_list,
                    Medium_list_temp,Medium_list,Low_list_temp,Low_list):-
      List_out=["major",Soil_type|Tail],
      append([Soil_type],High_list_in,High_list_temp),
      make_list(Tail,Main_list_temp,Main_list,High_list_temp,High_list,
                    Medium_list_temp,Medium_list,Low_list_temp,Low_list).
make_list(List_out,Main_list_temp,Main_list,High_list_temp,High_list,
                    Medium_list_in,Medium_list,Low_list_temp,Low_list):-
      List_out=["secondary",Soil_type|Tail],
      append([Soil_type],Medium_list_in,Medium_list_temp),
      make_list(Tail,Main_list_temp,Main_list,High_list_temp,High_list,
                    Medium_list_temp,Medium_list,Low_list_temp,Low_list).
make_list(List_out,Main_list_temp,Main_list,High_list_temp,High_list,
                    Medium_list_temp,Medium_list,Low_list_in,Low_list):-
      List_out=["minor",Soil_type|Tail],
      append([Soil_type],Low_list_in,Low_list_temp),
      make_list(Tail,Main_list_temp,Main_list,High_list_temp,High_list,
                    Medium_list_temp,Medium_list,Low_list_temp,Low_list).
make_list(List_out,Main_list_in,Main_list,High_list_temp,High_list,
                    Medium_list_temp,Medium_list,Low_list_temp,Low_list):-
      List_out=["second_main",Soil_type|Tail],
      append([Soil_type],Main_list_in,Main_list_temp),
      make_list(Tail,Main_list_temp,Main_list,High_list_temp,High_list,
                    Medium_list_temp,Medium_list,Low_list_temp,Low_list).


                      /*******COMPARISON*******/


compare:-
      enter(Soil_list1,Cons_list1,Col_list1,Bed_list1,Spacing_list1,Mod_list1,
                                  Layer_str_list1,_),
      enter(Soil_list2,Cons_list2,Col_list2,Bed_list2,Spacing_list2,Mod_list2,
                                  Layer_str_list2,_),
      compare_soils(Soil_list1,Layer_str_list1,Soil_list2,Layer_str_list2,
                                  Sim_list1),
      process_col(Col_list1,Mod_list1,List1),
      process_col(Col_list2,Mod_list2,List2),
      compare_colour(List1,List2,List),
      process_cons(Cons_list1,Cons1),
      process_cons(Cons_list2,Cons2),
      compare_cons(Cons1,Cons2,Cons_list),nl,
      compare_bedding(Bed_list1,Bed_list2,Bed_list),nl,
```

A20

```prolog
        compare_spacing(Spacing_list1,Spacing_list2,Spacing_list),nl,nl,
        bed_or_spacing(Bed_list,Spacing_list,Structure),
        overall_sim(Sim_list1,List,Cons_list,Structure).

compare_soils(Soil_list1,Layer_str_list1,Soil_list2,Layer_str_list2,Final_list):-
        last_elem(Layer_str_list1,_,Layer_list1),
        last_elem(Layer_str_list2,_,Layer_list2),
        process_soil(Soil_list1,Psd1),
        process_soil(Layer_list1,Psd2),
        process_soil(Soil_list2,Psd3),
        process_soil(Layer_list2,Psd4),
        calc_sim(Psd1,Psd3,Sim_list1),
        calc_sim(Psd1,Psd4,Sim_list2),
        calc_sim(Psd2,Psd3,Sim_list3),
        calc_sim(Psd2,Psd4,Sim_list4),
        final_sim(Sim_list1,Sim_list2,Sim_list3,Sim_list4,Final_list).

bed_or_spacing(List,[],List).
bed_or_spacing([],List,List).

compare_bedding([],[],[]):-!.
compare_bedding([Bed1],[Bed2],[Bed_Sim]):-
        str_spacing(Bed1,_,N1),
        str_spacing(Bed2,_,N2),
        Bed_Sim=100-abs(N1-N2),!,
        write("Bedding Similarity = ",Bed_Sim).
compare_bedding(_,_,[]):-
        write("Bedding Similarity : ",unknown).

compare_spacing([],[],[]):-!.
compare_spacing([_,Spacing1],[_,Spacing2],[Spac_Sim]):-
        str_spacing(Spacing1,_,Sp1),
        str_spacing(Spacing2,_,Sp2),
        Spac_Sim=100-abs(Sp1-Sp2),!,
        write("Spacing Similarity = ",Spac_Sim).
compare_spacing(_,_,[]):-
        write("Spacing Similarity : ",unknown).

compare_cons([],[],[]).
compare_cons([Cons_number1],[Cons_number2],[Number]):-
        Number=100-2*abs(Cons_number1-Cons_number2),
        write("N value Similarity = ",Number).
compare_cons(_,_,[]):-
        write("N value Similarity : ",unknown).

compare_colour([],[],[]).
compare_colour([],_,[]):-!,
        write("Unknown Colour Similarity "),nl.
compare_colour(_,[],[]):-!,
        write("Unknown Colour Similarity "),nl.
```

```prolog
compare_colour([Hue1,Lum1,Sat1],[Hue2,Lum2,Sat2],List):-
    compare_hue([Hue1],[Hue2],Hue),
    compare_lum([Lum1],[Lum2],Lum),
    compare_sat([Sat1],[Sat2],Sat),
    col_similarity(Hue,Sat,Lum,List).

col_similarity([Hue],[Lum],[Sat],[Col_sim]):-
    Hue_sim=0.4*(100-(Hue/120)*100),
    Lum_sim=0.3*(100-(Lum/240)*100),
    Sat_sim=0.3*(100-(Sat/240)*100),
    Col_sim=(Hue_sim+Lum_sim+Sat_sim),
    Col_simil=round(Col_sim),
    write("Colour Similarity  = ",Col_simil),nl.

compare_hue([],[],[]):-!.
compare_hue(_,[Hue],[0]):-
    Hue=-240,!.
compare_hue([Hue],_,[0]):-
    Hue=-240,!.
compare_hue([Hue1],[Hue2],[Hue_difference]):-
    Hue_difference=abs(Hue1-Hue2),
    Hue_difference<=120,!.
compare_hue([Hue1],[Hue2],[Hue_difference]):-
    Hue_dif=abs(Hue1-Hue2),Hue_dif>120,!,
    Hue_difference=240-Hue_dif.
compare_hue(_,_,[]).

compare_sat([],[],[]):-!.
compare_sat([-240],[0],[0]):-!.
compare_sat([0],[-240],[0]):-!.
compare_sat([Sat1],[Sat2],[Sat_difference]):-
    abs(Sat1-Sat2)=480,
    Sat_difference=0,!.
compare_sat([Sat1],[Sat2],[Sat_difference]):-
    abs(Sat1-Sat2)=Sat_difference,!.
compare_sat(_,_,[]).

compare_lum([],[],[]):-!.
compare_lum([Lum1],[Lum2],[Lum_difference]):-
    Lum_difference=abs(Lum1-Lum2),!.
compare_lum(_,_,[]).

calc_sim([],[],[]).
calc_sim([],_,[]).
calc_sim(_,[],[]).
calc_sim(Psd,Psdd,Sim_number_list):-
    sum_percents(Psd,0,[],Psd_new),
    sum_percents(Psdd,0,[],Psdd_new),
    create_sim_list(Psd_new,Psdd_new,[],Sim_list),
    find_similarity(Sim_list,0,Simil),
```

```prolog
        Similar=100-(Simil/6),Similarity=round(Similar),
        Sim_number_list=[Similarity],
        write_list("","Psd1=",Psd),
        write_list("","Psd2=",Psdd),
        write("Soil type Similarity = ",Similarity),nl,nl.

find_similarity([],Similarity,Similarity):-!.
find_similarity(Psd_new,A,Similarity):-
        Psd_new=[H1|Tail],
        Sum=H1+A,
        find_similarity(Tail,Sum,Similarity).


create_sim_list([],[],Psd_new,Psd_new):-!.
create_sim_list(Psd1,Psd2,Psd_in,Psd_new):-
        Psd1=[Head1|Tail1],
        Psd2=[Head2|Tail2],
        New=abs(Head1-Head2),
        append([New],Psd_in,Psd_temp),
        create_sim_list(Tail1,Tail2,Psd_temp,Psd_new).


sum_percents([],1,Psd_temp,Psd_new):-!,
        reverse(Psd_temp,Psd_new).
sum_percents(Psd,A,Psd_in,Psd_new):-
        Psd=[Head|Tail],A=0,
        append([Head,0],Psd_in,Psdnew),
        sum_percents(Tail,1,Psdnew,Psd_new).
sum_percents(Psd,1,Psdnew,Psd_new):-
        Psd=[Head|Rest],Psdnew=[H|_],
        Sum=Head+H,
        append([Sum],Psdnew,Psd_temp),
        sum_percents(Rest,1,Psd_temp,Psd_new).


final_sim(Similarity,[],[],[],Similarity).
final_sim([S1],[S2],[],[],[S]):-!,
        S=(S1+S2)/2.
final_sim([S1],[],[S3],[],[S]):-!,
        S=(S1+S3)/2.
final_sim([S1],[S2],[S3],[S4],[Max]):-
        N1=(S1+S4)/2,
        N2=(S2+S3)/2,
        find_max(N1,N2,Max).


find_max(A,B,Max):-
        A>=B,Max=A.
find_max(A,B,Max):-
        A<B,Max=B.


overall_sim([Soil],[Col],[Cons],[Structure]):-
        Sim=(0.65*Soil)+(0.15*Cons)+(0.1*Structure)+(0.1*Col),
        Simil=round(Sim),
```

```prolog
        write("Overall Similarity Number :",Simil),nl,nl.
overall_sim([Soil],[Col],[Cons],[]):-
        Sim=(0.65*Soil)+(0.20*Cons)+(0.15*Col),
        Simil=round(Sim),
        write("Overall Similarity Number :",Simil),nl,nl.
overall_sim([Soil],[Col],[],[Structure]):-
        Sim=(0.7*Soil)+(0.15*Col)+(0.15*Structure),
        Simil=round(Sim),
        write("Overall Similarity Number :",Simil),nl,nl.
overall_sim([Soil],[Col],[],[]):-
        Sim=(0.85*Soil)+(0.15*Col),
        Simil=round(Sim),
        write("Overall Similarity Number :",Simil),nl,nl.
overall_sim([Soil],[],[Cons],[Structure]):-
        Sim=(0.65*Soil)+(0.2*Cons)+(0.15*Structure),
        Simil=round(Sim),
        write("Overall Similarity Number :",Simil),nl,nl.
overall_sim([Soil],[],[Cons],[]):-
        Sim=(0.8*Soil)+(0.2*Cons),
        Simil=round(Sim),
        write("Overall Similarity Number :",Simil),nl,nl.
overall_sim([Soil],[],[],[Structure]):-
        Sim=(0.85*Soil)+(0.15*Structure),
        Simil=round(Sim),
        write("Overall Similarity Number :",Simil),nl,nl.
overall_sim([Soil],[],[],[]):-
        Sim=Soil,
        write("Overall Similarity Number :",Sim),nl,nl.


choose(1):-cursor(2,0),parse.
choose(2):-cursor(2,0),assign_values.
choose(3):-cursor(2,0),compare.



goal
clearwindow,
menu(5,5,15,1,[parse,assign_values,compare],"Select action",1,No),choose(No).
```

APPENDIX B

BOREHOLE-TO-BOREHOLE MODULE

domains

file=input

list=symbol*
reallist=real*
integerlist=integer*
stringlist = string*
llist = pair(integerlist)
result=llist*

database
layer(integer,integer,real,integerlist)
borehole(integer,real,integerlist)

predicates

append(integerlist,integerlist,integerlist)
append(list,list,list)
append(reallist,reallist,reallist)
append(result,result,result)
check_length(integer,integer)
check_sublist(result,result,result,result)
combine_link(integerlist,integerlist,integerlist)
comp_layer(integer,integer,integerlist,integerlist,integerlist,integerlist)
comp_psd(integer,integer,integer,integer,real)
compare_list(integerlist,integerlist,integerlist,integerlist,integerlist)
compound_sublist(llist,result)
create_sim_list(integerlist,integerlist,integerlist,integerlist)
delete_elem(integer,integer,integerlist,integerlist)
enter(integer,integer)
files(integer,string)
find_compound_length(llist,integer,integer)
find_element(integerlist,integerlist,integer,integer,integer)
find_list_max(integerlist,integer,integer)
find_sim(integerlist,integerlist,integer)
find_similarity(integerlist,integer,integer)
hyp_list(integerlist,integerlist,integerlist,integerlist,integerlist,integerlist)
hypothesis(result,integer,result,result)
identify_links(integerlist,integer,integer,integer,integerlist,integerlist)
incr_list(integerlist)

increasing_sublist(integerlist,integerlist,llist)
last(integerlist,integer)
last_check(result,result,result)
layer_list(integer,integer,integerlist,integerlist,integerlist,integerlist,integerlist,
                                    integerlist,integerlist,integerlist)
length_list(integerlist,integer,integer)
look_for(integerlist,integer,integer)
max(integer,integer,integer)
member(integer,integerlist)
member(symbol,list)
reverse(integerlist,integerlist)
reverse(list,list)
run
sort_lengths(result,Integer,result,result)
start(integer,integer,integerlist,integerlist,integer,integer,integerlist,integerlist)
sublist(integerlist,integerlist)
sublist1(integerlist,integerlist)
sum_percents(integerlist,integer,integerlist,integerlist)
transform_link_list(integerlist,integerlist,integerlist,integerlist)
valid_hyps(integerlist,result)
write_comp_list(integer,result)
write_list(symbol,symbol,list)
write_list(symbol,symbol,integerlist)
write_list(symbol,symbol,reallist)
write_list(symbol,symbol,result)

/*******STANDARD CLAUSES*******/

clauses

append([],List,List).
append([X|L1],List2,[X|L3]) :-
        append(L1,List2,L3).

reverse([],[]).
reverse([Head|Tail],List) :-
        reverse(Tail,Result),
        append(Result,[Head],List).

last([Last],Last).
last([_|Tail],Last) :-
        last(Tail,Last).

length_list([],Length,Length).
length_list([_|Tail],Len,Length):-
        Len1=Len+1,
        length_list(Tail,Len1,Length).

B2

```prolog
member(Name,[Name|_]).
member(Name,[_|Tail]) :-
  member(Name,Tail).
```

/********Borehole-to-Borehole Correlation********/

```prolog
run:-
        write("Which is the first file to consult :"),
        readln(File1),
        substring(File1,3,1,B1),str_int(B1,Bor_1),
        consult(File1),
        write("Which is the second file to consult :"),
        readln(File2),files(Bor_1,File2).


files(Bor_1,File2):-
        substring(File2,3,2,B2),str_int(B2,Bor_2),
        consult(File2),
        enter(Bor_1,Bor_2);
        substring(File2,3,1,B2),
        str_int(B2,Bor_2),
        consult(File2),
        enter(Bor_1,Bor_2).


enter(Bor_1,Bor_2):-
        borehole(Bor_1,_,LList1),
        borehole(Bor_2,_,LList2),
        layer_list(Bor_1,Bor_2,LList1,LList2,[],_,[],Link_list1,[],_),
        layer_list(Bor_2,Bor_1,LList2,LList1,[],_,[],Link_list2,[],_),
        combine_link(Link_list1,Link_list2,_).


layer_list(_,_,[],_,Sim_list_out,Sim_list_out,Link_list_out,
                        Link_list_out,Similarity_out,Similarity_out):-
        reverse(Link_list_out,Link_list),
        reverse(Similarity_out,Similarity_list),
        write_list("","Link List",Link_list),nl,
        write_list("","S",Similarity_list),nl,
        readchar(_),
        valid_hyps(Link_list,Hyps_list),
        write_comp_list(0,Hyps_list).
layer_list(Bor_1,Bor_2,LList1,[H2|T2],Sim_list_in,Sim_list_out,Link_list_in,
                        Link_list_out,Similarity_in,Similarity_out):-
        LList1=[H1|_],
        comp_psd(Bor_1,Bor_2,H1,H2,Similarity),
        append([Similarity],Sim_list_in,Sim_list_temp),
        layer_list(Bor_1,Bor_2,LList1,T2,Sim_list_temp,Sim_list_out,
                        Link_list_in,Link_list_out,Similarity_in,Similarity_out).
layer_list(Bor_1,Bor_2,LList1,[],Sim_list_out,Sim_list_out,Link_list_in,
                        Link_list_out,Similarity_in,Similarity_out):-
        borehole(Bor_2,_,LList2),
        LList1=[H|T],
```

```
        reverse(Sim_list_out,Sim_list),
        append(Sim_list_out,Similarity_in,Similarity_temp),
        find_list_max(Sim_list,0,Max),
        identify_links(Sim_list,H,0,Max,[],Link_list),
        append(Link_list,Link_list_in,Link_list_temp),
        layer_list(Bor_1,Bor_2,T,LList2,[],_,Link_list_temp,Link_list_out,
                Similarity_temp,Similarity_out).


combine_link(Link1,Link2,Link_list):-
        reverse(Link2,Link_2),
        compare_list(Link1,Link1,Link_2,[],Add_list),
        transform_link_list(Link1,Add_list,[],Link_list),
        valid_hyps(Link_list,Hyps_list),
        write_comp_list(0,Hyps_list).


transform_link_list(List,[],Link_in,Link_out):-
        reverse(Link_in,Link_temp),
        append(Link_temp,List,Link_out).
transform_link_list([],List,Link_in,Link_out):-
        reverse(Link_in,Link_temp),
        append(Link_temp,List,Link_out).
transform_link_list([H1,H2|T],[H3,H4|Tail],Link_in,Link_out):-
        H1<=H3,H2<=H4,
        append([H2,H1],Link_in,Link_temp),
        transform_link_list(T,[H3,H4|Tail],Link_temp,Link_out).
transform_link_list([H1,H2|T],[H3,H4|Tail],Link_in,Link_out):-
        H3<=H1,H4<=H2,
        append([H4,H3],Link_in,Link_temp),
        transform_link_list([H1,H2|T],Tail,Link_temp,Link_out).
transform_link_list([H1,H2|T],[H3,H4|Tail],Link_in,Link_out):-
        H1<=H3,
        append([H2,H1],Link_in,Link_temp),
        transform_link_list(T,[H3,H4|Tail],Link_temp,Link_out).
transform_link_list([H1,H2|T],[H3,H4|Tail],Link_in,Link_out):-
        H3<=H1,
        append([H4,H3],Link_in,Link_temp),
        transform_link_list([H1,H2|T],Tail,Link_temp,Link_out).


compare_list(_,_,[],Link_list,Link_list):-!.
compare_list(Link1,[],[H3,H4|T],List_in,List_out):-!,
        append([H3,H4],List_in,List_temp),
        compare_list(Link1,Link1,T,List_temp,List_out).
compare_list(Link1,[H1,H2|_],[H1,H2|Tail],List_in,List_out):-!,
        compare_list(Link1,Link1,Tail,List_in,List_out).
compare_list(Link1,[_,_|T],[H3,H4|Tail],List_in,List_out):-!,
        compare_list(Link1,T,[H3,H4|Tail],List_in,List_out).


comp_layer(_,_,_,[],_,[]).
comp_layer(Bor_1,Bor_2,Similarity_list,[H1|T1],List,[H|T]):-
        H1new=H1-1,H1new>0,
```

```prolog
                Hnew=H-1,Hnew>0,
                comp_layer(Bor_1,Bor_2,Similarity_list,[H1new|T1],List,[Hnew|T]).
comp_layer(Bor_1,Bor_2,Similarity_list,[H1|T1],List,[H|T]):-
                start(Bor_1,Bor_2,Similarity_list,List,H1,H,[],_),
                comp_layer(Bor_1,Bor_2,Similarity_list,T1,List,T).


start(_,B2,_,_,_,L2,Sim_out,Sim_out):-
                check_length(B2,L2),
                reverse(Sim_out,Sim_list),
                write_list("","Similarity List = ",Sim_list).
start(B1,_,_,_,L1,_,Sim_out,Sim_out):-
                check_length(B1,L1),
                reverse(Sim_out,Sim_list),
                write_list("","Similarity List = ",Sim_list).
start(B1,B2,Similarity_list,List,L1,L2,Sim_in,Sim_out):-
                find_element(Similarity_list,List,L1,L2,Sim),
                append([Sim,L2,L1],Sim_in,Sim_temp),
                L1new=L1+1,L2new=L2+1,
                start(B1,B2,Similarity_list,List,L1new,L2new,Sim_temp,Sim_out).


hyp_list(_,[],_,[],Hyp_list_temp,Hyp_list_out):-
                reverse(Hyp_list_temp,Hyp_list_out),
                write_list("","Hypothesis List",Hyp_list_out),nl.
hyp_list(Similarity_list,[H|T],LList2,[H1|T1],Hyp_list_in,Hyp_list_out):-
                find_element(Similarity_list,LList2,H,H1,Similarity),
                append([Similarity,H1,H],Hyp_list_in,Hyp_list_temp),
                hyp_list(Similarity_list,T,LList2,T1,Hyp_list_temp,Hyp_list_out).


find_element(_,_,_,0,0):-!.
find_element(Similarity_list,LList2,L1,L2,Similarity):-
                length_list(LList2,0,Len),
                Rank=((L1-1)*Len)+L2,
                look_for(Similarity_list,Rank,Similarity).


look_for(List,Rank,Similarity):-
                append(List1,_,List),
                length_list(List1,0,Rank),
                last(List1,Similarity).


check_length(Borehole,Layer):-
                borehole(Borehole,_,List),
                length_list(List,0,Length),
                Layer>Length.


comp_psd(Bor_1,Bor_2,L1,L2,Similarity):-
                layer(Bor_1,L1,_,List1),
                layer(Bor_2,L2,_,List2),
                find_sim(List1,List2,Similarity).


find_sim(Psd,Psdd,Similarity):-
```

```prolog
        sum_percents(Psd,0,[],Psd_new),
        sum_percents(Psdd,0,[],Psdd_new),
        create_sim_list(Psd_new,Psdd_new,[],Sim_list),
        find_similarity(Sim_list,0,Simil),
        Similar=100-(Simil/6),
        Similarity=round(Similar).


sum_percents([],1,Psd_temp,Psd_new):-!,
        reverse(Psd_temp,Psd_new).
sum_percents(Psd,A,Psd_in,Psd_new):-
        Psd=[Head|Tail],A=0,
        append([Head,0],Psd_in,Psdnew),
        sum_percents(Tail,1,Psdnew,Psd_new).
sum_percents(Psd,1,Psdnew,Psd_new):-
        Psd=[Head|Rest],Psdnew=[H|_],
        Sum=Head+H,
        append([Sum],Psdnew,Psd_temp),
        sum_percents(Rest,1,Psd_temp,Psd_new).


create_sim_list([],[],Psd_new,Psd_new):-!.
create_sim_list(Psd1,Psd2,Psd_in,Psd_new):-
        Psd1=[Head1|Tail1],
        Psd2=[Head2|Tail2],
        New=abs(Head1-Head2),
        append([New],Psd_in,Psd_temp),
        create_sim_list(Tail1,Tail2,Psd_temp,Psd_new).


find_similarity([],Similarity,Similarity):-!.
find_similarity(Psd_new,A,Similarity):-
        Psd_new=[H1|Tail],
        Sum=H1+A,
        find_similarity(Tail,Sum,Similarity).


identify_links([],_,_,_,List_out,List_out).
identify_links([H|T],Layer1,No,Max,List_in,List_out):-
        Max>H,
        No1=No+1,
        identify_links(T,Layer1,No1,Max,List_in,List_out).
identify_links([H|T],Layer1,No,Max,List_in,List_out):-
        Max=H,
        No1=No+1,
        append([No1,Layer1],List_in,List_temp),
        identify_links(T,Layer1,No1,Max,List_temp,List_out).


find_list_max([],Max,Max).
find_list_max(List,Temp_max,All_max):-
        List=[H|Tail],
        max(H,Temp_max,Max),
        find_list_max(Tail,Max,All_max).
```

```prolog
max(A,B,A):-A>=B.
max(A,B,B):-A<B.

write_list(Level,Type,List_out):-
        write(Level," ",Type," : ",List_out),nl.

find_compound_length(pair(List),0,Len):-
        length_list(List,0,Len).

delete_elem(X1,X2,[X1,X2|T],T).
delete_elem(X1,X2,[_,_|T],List):-
        delete_elem(X1,X2,T,List).

sublist(_,[]).
sublist(List1,[H1,H2|T]):-
        delete_elem(H1,H2,List1,List),
        sublist(List,T).

sublist1([],_).
sublist1([H1,H2|T],List1):-
        delete_elem(H1,H2,List1,List),
        sublist1(T,List).

compound_sublist(pair(List),[pair(List1)|_]):-
        sublist1(List,List1).
compound_sublist(pair(List),[pair(List1)|T]):-
        not(sublist1(List,List1)),
        compound_sublist(pair(List),T).

check_sublist([],_,List,List1):-
        sort_lengths(List,0,[],List1).
check_sublist([H|T],Result_list,List_in,List_out):-
        compound_sublist(H,Result_list),
        check_sublist(T,Result_list,List_in,List_out).
check_sublist([H|T],Result_list,List_in,List_out):-
        not(compound_sublist(H,Result_list)),
        append([H],List_in,List_temp),
        check_sublist(T,Result_list,List_temp,List_out).

increasing_sublist(List1,List2,pair(List2)):-
        write("Marker bed "),nl,
        readterm(integerlist,List3),nl,
        sublist(List1,List2),
        sublist(List2,List3),
        incr_list(List2),
        length_list(List2,0,Len),
        Len>4.

incr_list([_,_]).
incr_list([H1,H2,H3,H4|T]):-
```

```prolog
        H1<=H3,
        H2<=H4,
        incr_list([H3,H4|T]).

hypothesis([],_,Result,Result).
hypothesis([H|T],No,Res_in,Res_out):-
        find_compound_length(H,0,Len),
        Len<No,
        hypothesis(T,No,Res_in,Res_out).

hypothesis([H|T],No,_,Res_out):-
        find_compound_length(H,0,Len),
        Len>No,
        hypothesis(T,Len,[H],Res_out).

hypothesis([H|T],No,Res_in,Res_out):-
        find_compound_length(H,0,Len),
        Len=No,
        append([H],Res_in,Res_temp),
        hypothesis(T,No,Res_temp,Res_out).

sort_lengths([],_,List,List).
sort_lengths([H|T],No,List_in,List_out):-
        find_compound_length(H,0,Len),
        Len>=No,
        append(List_in,[H],List_temp),
        sort_lengths(T,Len,List_temp,List_out).
sort_lengths([H|T],No,List_in,List_out):-
        find_compound_length(H,0,Len),
        Len<No,
        append([H],List_in,List_temp),
        sort_lengths(T,No,List_temp,List_out).

last_check([],List,List).
last_check([H|T],Listin,Listout):-
        compound_sublist(H,T),!,
        last_check(T,Listin,Listout).
last_check([H|T],Listin,Listout):-
        not(compound_sublist(H,T)),
        append([H],Listin,Listemp),
        last_check(T,Listemp,Listout).

valid_hyps(List,List_out):-
        findall(Result,increasing_sublist(List,_,Result),List1),
        hypothesis(List1,0,[],List2),
        check_sublist(List1,List2,List2,Listout),
        last_check(Listout,[],List_out).

write_comp_list(_,[]):-readchar(_).
write_comp_list(No,List):-
```

```
New=No+1,
List=[pair(List1)|T],
write("Hypothesis ",New),nl,
write_list("","",List1),nl,
write_comp_list(New,T).
```

APPENDIX C

University of Durham
School of Engineering and Computer Science

Comparison of Soils to assist in the development of Geotechnical Knowledge Based Systems

Questionnaire

When interpreting ground conditions from borehole logs we look at the descriptions for two soils in adjacent boreholes and make a decision as to how similar they are, in order to identify whether they belong to the same soil layer.

To assist us in the development of a Knowledge Based System which will help with this interpretation, could you please circle the most appropriate similarity rating, in accordance with the terms listed below, for the following pairs of soil descriptions:-

A) Very similar
B) Similar
c) Slightly similar
D) Slightly dissimilar
E) Dissimilar
F) Very dissimilar

Comparison 1

Sandy CLAY

A	B	C	D	E	F

Clayey SAND

Comparison 2

SILT

A	B	C	D	E	F

Silty CLAY

Comparison 3

Gravelly SAND

A	B	C	D	E	F

Clayey GRAVEL

Comparison 4

Stiff sandy CLAY

A	B	C	D	E	F

Medium dense clayey SAND

Comparison 5

Loose SILT

A	B	C	D	E	F

Stiff silty CLAY

Comparison 6

Thinly laminated sandy CLAY

A	B	C	D	E	F

Very thinly bedded clayey SAND

Comparison 7

Brownish green gravelly SAND

A	B	C	D	E	F

Grey clayey GRAVEL

Comparison 8

Very loose clayey SAND

A	B	C	D	E	F

Firm sandy CLAY

Comparison 9

Thickly bedded gravelly SAND

A	B	C	D	E	F

Thickly laminated clayey GRAVEL

Comparison 10

Reddish brown SILT

A	B	C	D	E	F

Grey silty CLAY

Comparison 11

Loose thinly laminated SILT

A	B	C	D	E	F

Stiff thickly laminated silty CLAY

Comparison 12

Firm grey thickly bedded sandy CLAY

A	B	C	D	E	F

Very loose reddish brown very thinly bedded clayey SAND

Comparison 13

Yellow SILT with closely spaced fissures

A	B	C	D	E	F

Brown silty CLAY with extremely closely spaced fissures

Comparison 14

Very loose brownish green thickly bedded gravelly SAND

A	B	C	D	E	F

Medium dense grey thickly laminated clayey GRAVEL

Comparison 15

Silty CLAY interbedded with gravelly SAND

A	B	C	D	E	F

GRAVEL interbedded with silty CLAY

Please circle what you consider to be the relative importance for each component in a soil description, according to the scale given below.

    A) Extremely important
    B)Very important
    C)Important
    D)Unimportant
    E)Trivial

A	B	C	D	E

SOIL TYPE

A	B	C	D	E

CONSISTENCY

A	B	C	D	E

STRUCTURE

A	B	C	D	E

COLOUR

A	B	C	D	E

Would you consider any other factors when making such a comparison between two soils?

....
....
....
....
....

Do you have any other general comments on this exercise?

....
....
....
....
....

C3