Development of a 3D GIS database model for geotechnical analysis incorporating geostatistics

Gao, Shan

2007

Gao, S. (2007). Development of a 3D GIS database model for geotechnical analysis incorporating geostatistics. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/39359>

<https://doi.org/10.32657/10356/39359>

9530276

Development of a 3D GIS Database Model for Geotechnical Analysis Incorporating Geostatistics

Gao Shan

School of Civil and Environmental Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2007

To my parents
To my wife & daughter

ACKNOWLEDGEMENTS

I would like to express sincere thanks to my supervisor, Associate Professor Teh Cee Ing for his invaluable guidance and encouragement throughout the course of this study. His help in the preparation of this thesis is especially appreciated.

I am deeply grateful to Associate Professor Tor Yam Khoon and Associate Professor Goh Pong Chai who allow me to access the resources of the Spatial Information Laboratory. I wish to thank Professor Zhao Jian, Associate Professor Wong Kai Sin, and Associate Professor Goh Teck Chee, Anthony for their invaluable advice, discussion and encouragement.

I would like to thank Lim-Loh Mee Ching, Violet; Low Teh Yong, Eric from the Spatial Information Laboratory; Foo Shiang Kim and Fok Yew Seng, Anthony from the Hydraulic Laboratory for all the assistance and support during the course of this study.

I wish to express thanks to Selvarani Kathirvelu for the enduring support and help she gave through this study.

I must thank a lot of friends for the support, encouragement over the course of this study.

And the special thanks go to my family: my parents, my dear wife Gao Li and my lovely daughter YuNing, who's constant support make this happen.

TABLE OF CONTENTS

DEDICATION	ii
ACKNOWLEDGMENTS	iii
TABLE OF CONTENTS	iv
SUMMARY	viii
LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF PL/SQL PROCEDURES	xiv
Chapter 1 INTRODUCTION	1
1.1 GIS for Geotechnical Data	1
1.2 Motivation of This Research	5
1.3 Research Objective	6
1.4 Research Scope	6
1.5 Outlines of This Report	7
Chapter 2 FUNDAMENTALS OF DATABASE MODEL	10
2.1 Database Model and Modelling	10
2.2 Entity Relationship Model and Unified Modelling Language	13
2.2.1 Entity Relationship Model	13
2.2.2 UML	14
2.3 Criteria for a Database Model	19
2.4 Summary	20
Chapter 3 LITERATURE REVIEW	22
3.1 Geotechnical Database Models	22
3.2 Applicability of Reviewed Database Models for 3D GIS	48
3.3 Variogram of Geotechnical Data	50
3.3.1 Definition of Variogram	50

Table of Contents

3.3.2	Variogram Studies on Geotechnical Data	51
3.4	Summary	55
Chapter 4	GEOTECHNICAL DATA FOR THE PROPOSED DATABASE MODEL	56
4.1	Geotechnical Characterization Process	56
4.2	Borehole Test Data	59
4.3	Stratigraphy Data	61
4.3.1	Surface Data and Soil Layer	61
4.3.2	Stratigraphy Data	65
4.3.3	XML, GeotechML, GeosciML, AGSML and DIGGSML	66
4.3.4	XML for Stratigraphy Data	73
4.4	Characterizing Spatial Variation of Soil Parameters	77
4.4.1	Variogram	77
4.4.2	XML for Variogram Data	81
4.5	Summary	83
Chapter 5	HYBRID DATABASE MODEL	85
5.1	Worldview of Geosciences and NADM Model	85
5.2	Data Model Descriptions	88
5.2.1	Fact	92
5.2.2	Concept	92
5.2.3	Meta-data	103
5.3	Prototype Illustration	107
5.4	Database Model Evaluation	112
5.5	Summary	113
Chapter 6	CRYSTALGROUND: A DEMONSTRATION APPLICATION FOR THE PROPOSED DATABASE MODEL	115

Table of Contents

6.1	System Analysis	115
6.2	System Implementation	119
6.2.1	Geotechnical Database Server Implementation	119
6.2.2	Client Side Implementation	124
6.3	Summary	128
Chapter 7	CASE STUDY: STRATIGRAPHY MODELLING AND GEOSTATISTICAL KRIGING	130
7.1	The Synthetic Dataset	130
7.1.1	Stratigraphy Data	131
7.1.2	Simulation of Soil Parameter Values	134
7.2	Application of <i>CrystalGround</i>	136
7.2.1	Stratigraphy Study	136
7.2.2	Soil Parameter Estimation	150
7.3	Summary	158
Chapter 8	CASE STUDY: 3-D MODEL OF THE SUBSURFACE STRATTIGRAPHY AT A FAILED EXCAVATION PROJECT	160
8.1	Borehole Information	160
8.2	Application of <i>CrystalGround</i>	161
8.2.1	Data Input	161
8.2.2	Graphical Presentation of Soil Profiles	175
8.3	Summary	184
Chapter 9	CONCLUSIONS AND RECOMMANDATIONS	185
9.1	Conclusions	185
9.1.1	Geotechnical Database Model	185
9.1.2	Proof of Concepts	188
9.2	Recommendations for Further Study	189

Table of Contents

References	191
Appendix A An Example of DTD for Geotechnical Borehole at University of Durham	198
Appendix B SQL for Building an Oracle 8i Geotechnical Database	204
Appendix C Program Code for <i>CrystalGround</i>	208
Appendix D Datasets for Synthetic Stratigraphy Generation	248
Appendix E Soil Investigation Borehole Logs	256

SUMMARY

The application of 3D GIS and geostatistical tools in geotechnical fields enables geotechnical engineers to think spatially and make decision wisely. To support the 3D GIS applications, geotechnical data need to be managed based on their spatial components. The objective of this study is to develop a new geotechnical database model supporting the spatial analysis requirement of a 3D GIS. The result of this study will provide a foundation for the future development of a 3D GIS geotechnical database for Singapore Island.

A data requirement analysis has been conducted and based on the data requirements the new geotechnical database model is developed as a hybrid database model inheriting the advantage of existing geotechnical and geological models. The developed geotechnical database model combines the rational worldview of NADM model (North American Geologic-map Data Model), the rigorous data structure of the AGS format, and the flexibility of the XML schema.

The proposed geotechnical database model highlights the existence of different interpolation results based on the same entities by decoupling the geotechnical test data from the stratigraphy schema. It enables the geotechnical test data to be fitted into various stratigraphy schemas without any constraints. To handle the subsurface stratigraphy data, an XML schema is developed for the hybrid database model. Using the XML support from the database industry, the complex relationships interpolated for different soil subsurface can be stored in an XML database and retrieved for reuse. Rather than breaking the stratigraphy information into different segments just for storage purposes without any geotechnical meanings, XML database can manage stratigraphy data as an integrated unit.

Meanwhile, an XML schema for the variogram model is also proposed. By taking advantage of an XML database, the variogram for soil properties can be efficiently

Summary

stored and reused. Similar to the information for stratigraphy, the search and modification of variogram parameters can be quickly conducted due to the extensive availability of XML tools. The new geotechnical database model represents an early attempt to manage variogram data in a database, if not the first.

A demonstration application has been developed by implementing the new geotechnical database model. Two case studies have been conducted, one based on a synthetic dataset and the other based on the site investigation information involving the failure of a retaining wall system. Through the process of case studies, the applicability of the new database model is affirmed and its advantage for a 3D GIS in geotechnical data management is demonstrated. It can be concluded that the identified gap existing in the data analysis and distribution process could be filled or narrowed through the introduction of new geotechnical database model. The solution provided in this study gives geotechnical engineers a handy tool not only for sharing the geotechnical data but also, and most importantly, for sharing their knowledge with each other.

LIST OF TABLES

Table 2-1	Examples of UML diagrams (after Naiburg & Maksimchuk, 2001)	16
Table 3-1	Field description of the relationship database (after Ansary et al., 1999)	30-31
Table 3-2	List of reviewed database model	50
Table 4-1	Description of XML syntaxes	70
Table 4-2	Typical XML attribute types	70
Table 4-3	Typical XML attribute defaults	71
Table 5-1	Class attributes of <i>Fact</i> classes (Extracted from AGS format)	95-99
Table 5-2	Class attributes of <i>Concept</i> classes	101
Table 5-3	Class attributes of <i>Meta-data</i> classes (Extracted from AGS format)	106-107
Table 7-1	Soil layer information for sample boreholes	142

LIST OF FIGURES

Figure 1-1	Example of 3D-GIS visualization of proposed tunnel alignment (after Ozmutlu, 2001)	4
Figure 2-1	Example of ER diagram styles	15
Figure 2-2	Basic UML class diagram symbols	17
Figure 3-1	Schema for the GeoTech database structure (after Oliver & Toll, 1996)	24
Figure 3-2	Legend for the GeoTech database structure (after Oliver & Toll, 1996)	25
Figure 3-3	NGES database structure (after Satyanarayana & Benoit, 2001)	27
Figure 3-4	Relationship model for a geotechnical database proposed by Ansary et al. (1999)	29
Figure 3-5	The kernel USGS geologic site database (after Ponti, 2001)	33
Figure 3-6	Entity relationship diagram for distribute geotechnical data (after Real & Mcguire, 2001)	36
Figure 3-7	Tables and relationships between tables in ROSRINE database (after Nigbor et al., 2001)	40
Figure 3-8	Data model for site investigation (after Oloufa, et al., 1994)	44
Figure 3-9	Hierarchic representation of WELLHEAD Boring Log Database (after Adams et al., 1993)	46
Figure 3-10	GeSEP data model (after Suwanwiwattana et al., 2001)	47
Figure 4-1	Summary of the generic process of geotechnical characterization (after Houlding, 1994)	57
Figure 4-2	The relationships between soil layers	75
Figure 4-3	DTD for subsurface stratigraphy	76
Figure 4-4	An example of stratigraphy XML data	77
Figure 4-5	Characteristics for variogram description	78
Figure 4-6	DTD for variogram documentation	82
Figure 5-1	North America Data Model simplified (after Brodaric & Gahegan, 2000)	87

List of Figures

Figure 5-2	Diamond diagram of National Geologic Map Database Project (after Hastings & Brodaric, 2001)	88
Figure 5-3	The data application domain versus the data warehouse domain (after Stirling, 2002)	90
Figure 5-4	The hybrid data model for geotechnical database	91
Figure 5-5(a)	Class attributes of typical <i>Fact</i> classes for laboratory tests	93
Figure 5-5(b)	Class attributes of typical <i>Fact</i> classes for in-situ tests	94
Figure 5-6	Class attributes of <i>Concept</i> classes for soil layer category	100
Figure 5-7	Relationship types of strata in a connection	100
Figure 5-8	Class attributes of <i>Mata-data</i> classes for geotechnical tests	105
Figure 5-9	Database diagram for the prototype	109
Figure 5-10	Interpreted stratigraphy resulting from different judgments based on available borehole information	110
Figure 5-11	A SQL statement using the store procedure	111
Figure 6-1	Use case diagram of the demonstration application	117
Figure 6-2	Deployment diagram of <i>CrystalGround</i>	120
Figure 6-3	Static class model for <i>CrystalGround</i>	121
Figure 6-4	Attributes of point objects for representing geotechnical boreholes	123
Figure 6-5	Raster file in ASCII format	124
Figure 6-6	Snapshots of the GIS interface of <i>CrystalGround</i>	125
Figure 6-7	A template for borehole log report	127
Figure 7-1	Configuration of the synthetic study domain	132
Figure 7-2	Borehole plans for stratigraphy interpolation	138
Figure 7-3(a)	Borehole Logs (H1-H8)	139
Figure 7-3(b)	Borehole Logs (H9-H16)	140
Figure 7-3(c)	Borehole Logs (H17-H23)	141
Figure 7-4(a)	Section A-A for different sampling plans	146
Figure 7-4(b)	Section B-B for different sampling plans	147
Figure 7-4(c)	Section C-C for different sampling plans	148
Figure 7-4(d)	Section D-D for different sampling plans	149
Figure 7-5	The location plan for the sampled data and estimation data	151

List of Figures

Figure 7-6	The variogram model for the Marine Clay	152
Figure 7-7(a)	The comparison between the undrained shear strength estimation obtained by geostatistical kriging with the original defined data for BH-A	155
Figure 7-7(b)	The comparison between the undrained shear strength estimation obtained by geostatistical kriging with the original defined data for BH-B	156
Figure 7-7(c)	The comparison between the undrained shear strength estimation obtained by geostatistical kriging with the original defined data for BH-C	157
Figure 8-1	Layout of boreholes at the project site	162
Figure 8-2	Plan of boreholes and sections	176
Figure 8-3	Soil longitudinal section 1-3	177
Figure 8-4	A 3D diagram showing the existence of soft Marine Clay	178
Figure 8-5	Plan of assumed facility and sections	180
Figure 8-6(a)	The cross sections S1-S3	181
Figure 8-6(b)	The cross sections S4-S6	182
Figure 8-6(c)	The cross sections S7-S8	183

LIST OF PL/SQL PROCEDURES

List 7-1	PL/SQL procedure for storing the stratigraphy XML datagram	143
List 7-2	PL/SQL procedure for inserting variogram table	153
List 8-1	PL/SQL procedure for inserting project table	163
List 8-2	PL/SQL procedure for inserting borehole information table	164
List 8-3	PL/SQL procedure for inserting project borehole relation table	165
List 8-4	PL/SQL procedure for inserting SPT table	166
List 8-5	PL/SQL procedure of inserting source table	167
List 8-6	PL/SQL procedure for inserting sample table	167
List 8-7	PL/SQL procedure for inserting classification table	168
List 8-8	PL/SQL procedure for inserting stratum table	169
List 8-9	PL/SQL procedure for inserting connection table	171
List 8-10	PL/SQL procedure for inserting surface table	172
List 8-11	PL/SQL procedure for inserting soil layer table	173
List 8-12	PL/SQL procedure for inserting stratigraphy table	174

Chapter 1 Introduction

1.1 GIS for Geotechnical Data

Geotechnical data are numbers, characters, images or symbols from which geotechnical information can be derived. The information derived from geotechnical data may consist of geological history/geology, soil type, stratigraphy, soil mechanical properties, soil physical characteristic and water condition. The proper planning, design and construction of civil engineering projects rely heavily on a good understanding of geotechnical information. Analyzing and managing geotechnical data by applying engineering knowledge and making critical decisions is one major responsibility of geotechnical engineers.

Legacy geotechnical data is typically found in the form of hardcopy geotechnical reports that may be distributed around several organizations. These data are usually not readily accessible for use. Sometimes even their existence is unknown to the user. The information is often under-utilised and is undervalued because the data cannot be easily accessed or analysed. It is a difficult, laborious and time-consuming process to locate and retrieve these geotechnical data. Even when they are accessible, the existing data are often difficult to understand, identify, and integrate due to the various formats used. The data are also expensive to maintain. A Survey conducted in the 1980's by Cambridge Energy Research Associates (CERA) indicated that "geoscientists often spent upwards of 80% of their time simply trying to gather enough data to perform some useful work" (CERA, 1996). McBrayer et al. (1998) also reported that "in past practice, only \$1 of benefit is being achieved for every \$3 invested in environmental restoration. Much of this money has been spent in unsuccessful attempts to eliminate uncertainty in subsurface conditions". Very often, being unaware of the availability of usable existing geotechnical data, engineers carry out additional tests to obtain new data, inducing data deluge and wastage in resource utilization. The advancement of

technology may have contributed inadvertently to increases in disparate data rather than resolving the data deluge. Stirling (2002) described such a scenario called data application domain scenario, in which data are “locked in” by an application and cannot be accessed by other applications. This may also cause users to be trapped in vendor specific application software, which is not favoured by many users. This situation is more serious in today’s network environments because it represents a barrier to the flow of the information across the network.

The situation can be improved significantly with the help of computer information management systems. Many studies and projects have been conducted or are being carried out currently to develop such information management systems (Porsvig et al., 1989; Oloufa et al., 1992; O’Rourke & Pease, 1997; Parsons & Frost, 2000; Cheng et al., 2002). Two kernel components normally, if not always, exist for these information management systems: a database to manage geotechnical data and an application to analyse and to present geotechnical data. Through a geotechnical database, geotechnical data are no longer “locked in” via proprietary formats and are made more accessible to engineers interested in detailed information for a site. Such data will also be accessible to engineers interested in soil characteristics over a large region. Through the application, the geotechnical data in the geotechnical database are organized in a meaningful way and presented to the user in various formats: spread sheet form, graphic chart, log reports, cross section and 3-D view, just to mention a few examples.

Among the diversified information systems, Geographic Information Systems (GIS) have been shown to hold great promise. A GIS is “an organized collection of computer hardware, software, geographic data, and is designed to efficiently capture, store, update, manipulate, analyze, and display all forms of geographically referenced information” (ESRI 1997). Marble & Peuquet (1983) defined the GIS as a series of subsystems within a larger system:

- A data input subsystem that collects and reprocesses spatial data from various source.

- A data storage and retrieval subsystem that organizes the spatial data in a manner that allows retrieval, uploading and editing.
- A data manipulation and analysis subsystem that performs tasks on the data, aggregates and disaggregates, estimates parameters and constrains, and performs modelling functions
- A reporting subsystem that displays all or part of the database in tabular, graphic, or map form.

GIS not only provides a sophisticated spatial platform to handle complex spatial analysis, it also integrates the data management, analysis, and visualization functions that are typically performed independently and bridges the traditional “islands of automation”, it automates many time-consuming tasks and reduces man-hours in searching and retrieving information (Hellawell et al., 2001). The engineer can thus focus on using the data efficiently rather than spending great effort in just ‘preparing’ the data for various functions (Parsons & Frost, 2000). Through a comprehensive review, Toll (1998) pointed out that hybrid systems that combine different artificial intelligence techniques together with techniques such as database and GIS would probably be the most useful type of system for geotechnical site characterization. Figure 1-1 presents an example of 3D-GIS visualization of a proposed tunnel alignment together with planes showing the distribution of CPT values (Ozmutlu & Hack, 2001). Instead of producing a neatly sorted and summarized report, GIS, especially 3D GIS, can present geotechnical data through bright, attention grabbing, multi-dimensional graphs which reveal trends that may be hard to find in a text format.

Figure 1-1 Example of 3D-GIS visualization of proposed tunnel alignment (after Ozmutlu & Hack, 2001)

In using GIS for geotechnical data management, various degrees of customization of the geotechnical databases are normally needed. Such requirement for customization is generated from two factors:

- On the one hand, in most existing geotechnical databases the geotechnical data are not modelled spatially. Very often the spatial component of geotechnical data has been ignored due to the lack of spatial analysis tools.
- On the other hand, due to the relatively small market share the geotechnical industry commands in the overall GIS market place and the diversity of the requirements in the geotechnical field, an off-the-shelf GIS package seldom has the specialised features to fulfil all the requirements for geotechnical data management.

In order to exploit the full advantage of geotechnical data with a GIS, there is a need to understand the special characteristics of geotechnical data and the geotechnical logic residing in the data. A way must be found to gain control of geotechnical data, to avoid the proliferation of disparate data, and to develop a database that support the development of just-in-time information. The vision for integrating and sharing geotechnical data is predicated on a common data architecture within which all data are clearly defined so that they can be integrated into a data resource and readily shared.

1.2 Motivation of This Research

An integrated geotechnical GIS is particularly valuable in a well-developed urban setting. In Singapore's context, most areas are extensively developed. Island-wide construction activities have generated a huge amount of geotechnical investigation data. There is a potential to integrate the available geotechnical information from past projects in a single integrated framework to create a useful and intelligent database. Meanwhile, future city redevelopment and underground construction works are mostly conducted in the heavily built up area. The design, analysis, construction and the consequent impact of such construction works to nearby structures and the environment are critically dependent on an in-depth knowledge of the ground condition. The high stakes involved demand an information system to support the decision making process in an informed manner while consciously minimizing risk. Increasing geo-environmental concerns also call for a more information rich decision support system.

In response to the increasing information requirements, government agencies have started to invest in the development of GIS information systems. For the past years, two GIS information systems have been developed respectively by the Land Transport Authority (LTA) and the Housing and Development Board of Singapore (HDB). LTA's system is developed based on an earlier system for a specific project. With the extensive underground construction works spearheaded by the LTA, the early information management system GIS (Geotechnical Information System) for the North East and Changi Airport Line project has been shown to be a success (Kimmance, 1999). It has subsequently been evolved into the Borehole Information System (BIS) for providing daily support to other underground construction projects. Meanwhile, considering the huge amount of data accumulated from past projects and foreseeing the future upgrading needs, HDB has also developed its own information system to manage its geotechnical data. Although the two systems were developed to fulfill different requirements, one common feature for the two systems is that they were both built on a 2D GIS platform.

Managing the geotechnical data in a data management system is a great help for engineers who want to access the data. However, a 3D GIS decision support tool is more desirable for providing a subsurface view directly to support the users proposed judgments. The need for a 3D GIS tool has also risen from the lessons learned in the construction incidents caused by misinterpretation of subsurface conditions. For instance, in the Kranji Deep Tunnel Sewerage System project, the contractor has “encountered unanticipated terrain difficulties and have to provide S\$22 million for expected losses brought on by unforeseen complex ground conditions” (SIAR, 2002). Severe financial loss has been incurred due to the misinterpretation of subsurface conditions. It is desirable to minimize the possibility of this kind of misinterpretation through 3D GIS decision support tools which will enable the decision maker to think spatially and to decide wisely. As a 3D GIS requires extensive spatial analysis on geotechnical data, the first step in creating a 3D GIS application is to develop a new database model to support the spatial analysis.

1.3 Research Objective

The objective of this study is to develop a new geotechnical database model. The database model will support the spatial analysis requirement of a 3D GIS. The result of this study will provide a cornerstone for the future development of a 3D GIS geotechnical database for Singapore Island.

1.4 Research Scope

This study emphasizes the conceptual development of the database model and its validation by prototyping the key aspects of the proposed model. The scope of this study includes two stages.

- Development of a new database model

The data requirements for GIS applications are first identified at this stage,

where the gaps between existing database models are illustrated. Then a new database model is developed based on the identified data requirements. The new database model is further evaluated using a set of predefined criteria. The benchmark for this stage is that all the criteria are met by the proposed new database model.

- Proof of concept (POC) using a prototype

For a database model, especially a newly developed model, a prototype is the most effective way to communicate and verify the idea. At this stage, a prototype GIS application is developed together with a database constructed based on the proposed database model. Two case studies are conducted to verify the workability and the advantages of the proposed database model.

1.5 Outline of This Report

This report consists of 9 chapters. The current chapter gives the background for the initiation of this study, defines the study objectives and scope of the work.

In Chapter 2, the concept of database model in database theory is introduced. The goal of database model and the role of database model in the database design process are illustrated. Database model is represented using model diagrams. Two major methodologies for database modeling, Entity Relationship (ER) diagram and Unified Modelling Language (UML) diagram, are introduced. The evaluation criteria for database model are also covered in the final part of this chapter. This chapter provides the fundamental concepts for the subsequent literature review and the discussion on the development of the new database model.

In Chapter 3, a literature review of geotechnical database models for various applications, especially GIS applications in the geotechnical field, is presented. Through this review, a picture of the current state-of-art and development status for geotechnical database model is obtained. The existing gaps between GIS, database

and estimation tools that have been raised by some researchers are reviewed and the gaps need to be filled are illustrated. This chapter hence provides background support for the current study.

Chapter 4 identifies the key geotechnical data that should be included in a database model. The geotechnical data are categorized into three types: borehole test data, stratigraphy data and variogram data. The stratigraphy data and geostatistical variogram are represented using Extensible Markup Language (XML) and their data schemas are defined in this chapter.

Chapter 5 describes the proposed geotechnical database model. The model is a hybrid model created by merging the concept of the North American Geologic-map Data Model (NADM) with the traditional geotechnical database model. The proposed model decouples the geotechnical data and stratigraphy so that each of them can be managed separately. The data model is characterized by its data interoperability, user-centric paradigm, stratigraphy centralization and custodial orientation. By implementing this hybrid model in a database, a dataset can be easily created using the constraint of soil layer category. The new database model is then evaluated against the criteria set up in Chapter 2.

To demonstrate how the proposed database model can be implemented in a database and to illustrate the advantage of the proposed database model, a demonstration application, *CrystalGround*, has been developed. Chapter 6 describes the development of this prototype application.

Chapters 7 and 8 present two case studies using the application developed in this study. The applicability and advantage of the proposed database model are evaluated through the data analysis processes. Chapter 7 illustrates the data analysis process using a synthetic dataset developed in this study. Chapter 8 gives a case study involving the failure of a retaining wall system during construction.

Chapter 9 summarizes the findings obtained in this study and highlights the advantages of the developed database model. Some proposals for further study are also included in this chapter.

Chapter 2 Fundamentals of Database Model

Terms such as data model are widely used and have been given different meanings in different circumstances. This chapter addresses the basic concepts adopted in this study and aims to lay the fundamental understandings for the subsequent development. The chapter is organized into three sections. The first section defines the terms such as *model*, *modelling*, *database* and *database model* which are used in this study. The second part of this chapter introduces the two methodologies used for representing database model. The last section describes the evaluation criteria for a database model, which is adopted to evaluate the proposed new geotechnical database model.

2.1 Database Model and Modelling

Models are simplification of reality. On the basis of models, scientists can build and prove their hypothesis, make predictions, exchange ideas and gain knowledge. How models work is a curious question. Philosophy of science explores with the question. The process of defining and organizing data about real world into a consistent digital dataset that is useful and revealing information is called *data modelling*. (Bonham-Carter, 1996; Zlatanova, 2000).

A *database* is a collection of inter-related data that are managed by a database management system (Bisland, 1989, Garcia-Molina et al., 2002). The earlier commercial database systems are file-based systems which appeared in the late 1960's. Then the relational database systems are developed based on the relationship model proposed by Codd (1970). Although the relational database is the most popular technology for managing a collection of data, it has its disadvantages. For instance, it only supports its own type system and cannot handle complex objects. The object-relational database management system is developed to

extend the capabilities of the relational database by merging object and relational technologies. The core of object-relational is the relational store, objects still need to be composed into relation tables, rows and columns, for storage. However, the capability of adding extra types such as user defined data types, multi-valued data types and user-defined function provides greater flexibility to store business logic in the database level and hence promote reuse (Muller, 1999). The object-relational database is a compromise between both the object database and relational models and reflects the evolution of the database technology. Compared to a pure object-oriented database, the object-relational database management system is supported by most of the major database vendors in the market.

Data in relational/object-relational database system are stored in *tables*. A *table* consists of columns and rows. Each column contains a different type of attribute and each row corresponds to a single record. Databases keep all of the records through the use of keys. A *primary key* can be declared for one column or several columns in a table. A *primary key* means each row of the table must have a unique value in some column/columns. The *foreign key* enforces the referential integrity by creating relationships between tables. In a database, it is a field in one table that is indexed as a primary key in another. *Foreign keys* provide the building blocks for relating tables (Kifer et al., 2005). One feature supported by the database system is the *stored procedure*. A *stored procedure* is a group of Standard Query Language (SQL) statements that have been previously created and stored in the server database. *Stored procedures* accept input parameters so that a single procedure can be used over the network by several clients using different input data. *Stored procedures* reduce network traffic and improve performance. Additionally, *stored procedures* can be used to help ensure the integrity of the database (Kifer et al., 2005).

A database also needs to be created based on a model. A *database model* is a conceptual representation of the data structures that are required by a database. The data structures include the data objects, the associations between data objects, and

the rules which govern operations on the objects. As its name implies, the database model focuses on what data is required and how it should be organized rather than what operations will be performed on the data (Muller, 1999). To use a common analogy, the database model is equivalent to an architect's building plans. A database model is independent of hardware or software constraints. The database model focuses on representing the data as the user sees it in the "real world" rather than try to represent the data as a database would see it. A database model functions as a bridge between the concepts that make up real-world events and processes and the physical representation of those concepts in a database ((Muller, 1999; Kifer et al., 2005).

An architect represents the design idea through a set of drawings. The database model is often described using two types of documents. The first is a diagram which represents the data structures in a pictorial form. The diagram is a valuable tool to communicate the model to the end-users as it is easy to be understood. The second is a data document. This is a document that describes in detail the data objects, relationships, and the rules of operation required by the database. The goal of the database model is to make sure that the all data objects required by the database are completely and accurately represented. Because the database model uses easily understood notations and natural language, its correctness/meaning/purpose can be reviewed and verified by the end-users. The database model is usually detailed enough to be adopted by the database developers as a "blueprint" for building the physical database. The information contained in the database model will be used to define the relational tables, primary and foreign keys, and stored procedures (Muller, 1999, Simsion & Witt, 2005).

Database design is referred to the process of developing the logical and physical structure of one or more databases to accommodate the information needs for a defined set of applications. It encompasses the entire process from the creation of the requirements, the business process, logical analysis, and the actual setting up of the database to the deployment of the database (Naiburg & Maksimchuk, 2001). In

term of the scope of this study, the focus is on the logical database model rather than the physical implementation. In other words, the development of this study is part of the lifecycle of database design.

2.2 Entity Relationship Model and Unified Modelling Language

Two major methodologies currently exist to represent a database model: the *Entity-Relationship* (ER) diagram and the *Unified Modelling Language* (UML). The *ER* has been a dominant database modelling tool in the past, and it is still in use today. However, more recently, the UML has become the de-facto standard modelling language for database model.

2.2.1 Entity Relationship Model

The *ER* model is a conceptual database model that views the real world as entities and relationships between entities. In the *ER* model, the structure of data is graphically represented using following three principal element types (Thalheim, B. 2000, García-Molina et al., 2002):

- **Entities**
An entity is something which involves information. An entity is usually identifiable. A group of related entities become an entity set and each entity set is named. Entities are represented by boxes in the ER model.
- **Attributes**
Each entity has certain characteristics known as attributes. It is assumed that entity sets contain entities with same set of characteristics.
- **Relationships**
A Relationship is an association among entities. Relationships are classified in terms of *degree*, *connectivity*, *cardinality*, and *direction*.

There is no standard for graphically representing data objects in Entity-Relationship

and various styles exist. Each style uses a special set of symbols to represent the cardinality of a connection. Figure 2-1 gives a simple case showing the relationships of three entities presented in different ER diagram styles. Normally Entities are represented by labelled rectangles. The label is the name of the entity. Attributes of entity, when included, are listed inside the entity rectangle. Attributes which are identifiers of entity are underlined. Relationships are represented by a solid line connecting two entities. The name of the relationship is written above the line. Cardinality of many is represented by an asterisk (*) or N, the cardinality of one is represented as "1".

2.2.2 UML

The Unified Modelling Language (UML) has come into mainstream use through the pathway of object-oriented analysis and design. UML provides a set of diagrams used for capturing requirements, depicting deployment, and everything in between. Through these programs UML provides a mechanism for entering model elements into the model and showing overlapping sets of models, elements and their relationships. UML does not specify what diagrams should be created or what they should contain, only what they can contain and the rules for connecting the elements. Some different UML diagram types are listed in Table 2-1 (Naiburg & Maksimchuk, 2001).

Similar to entity and relationship defined in the ER model, UML defines class and relationship. UML defines class as a description of a set of objects that share the same attributes, operations, relationship and semantics. In other words, a class is an object's definition or template. While an object models a physical thing or an abstract concept, once modeled, the object can be manipulated in a natural way. A set of terms such as *association*, *inheritance*, *aggregation* and *composition*, are used to describe the relationship between classes and instances. Figure 2-2 shows the basic UML class diagram symbols and the notations are explained below (Page-Jones, M., 2000):

Figure 2-1 Example of ER diagram styles

Table 2-1 Examples of UML diagrams (after Naiburg & Maksimchuk, 2001)

Diagram	Description
Use case	The <i>use case</i> diagram is a model of the system’s intended functions and its environment that supports the business process. This method serves as a contract between the customer and developers
Interaction	<i>Interaction</i> diagrams are either sequence or collaboration diagram, both of which show the interaction of objects within the system. They can be used to understand queries that will affect the database and even help to build indexes based on the information modelled
Class	<i>Class</i> diagrams are logical models that show the basic structure of the system
Database	The <i>database</i> diagrams depict the structure of database including tables, columns, constraints, and so on

- **Class**

A *class* is the stencil from which objects are created. Class is illustrated with rectangles divided into three compartments. The topmost of the symbol’s three compartments shows the class name, the second partition shows the class attributes, and the bottom compartment presents the operations.

- **Associations**

Associations refer to the static relationships between classes. Associations are shown as a line between two classes with the association name appearing on that line. Roles representing the ways two classes see each other are placed near the end of an association.

- **Composition and Aggregation**

Composition and *Aggregation* are two whole/part associations. *Composition* represents a relationship between two classes where the “whole” class does not exist without its “part” class. In the aggregation association the “whole” class may exist without the “part” class. A association between the “whole” class to the “part” class is represented by a line with a diamond pointing

Figure 2-2 Basic UML class diagram symbols

toward the “whole” class. For the composition association, the diamond is filled with black; for the aggregation association, the open diamond is used.

- **Cardinality**

Cardinality (Multiplicity) refers to the number of objects in one class that can be related to one object in the related class. The cardinality notations can be used on both side of an association.

- **Generalization**

Generalization represents an inheritance hierarchy where a subclass has implicitly defined upon it each of the attributes and operation of a superclass, as if those attributes and operations had been defined upon the subclass itself. The generalization association is showed in UML as an arrow with an “open” head.

Both the ER model and UML can be adopted in the database design. The ER model is still a widely utilized method for relational database design. One specific feature of ER model is that it is very database centralized. It presumes that the database is the most important component and everything else should revolve with the database. On the other hand, UML is widely used in software analysis and design. One advantage for using UML in database modeling is that a common language can be adopted in the whole development process (Naiburg & Maksimchuk, 2001). As a model language, ER model is sourced from relational database and mainly implemented in the relational databases. UML model can not only be implemented in the relational database but also in object databases.

2.3 Criteria for a Database Model

A geotechnical design engineer uses design drawings and calculation reports to communicate with others. Similarly, the database model is expressed using model diagrams and documents. The database model needs to be validated based on established evaluation criteria. Oloufa et al. (1992) described that a database model should be correct, consistent, shareable and flexible. A correct model provides an accurate representation of the study domain being modeled. A consistent model contains no contradictions in the way the data elements are named, defined, related, and documented. A shareable model is accessible by a variety of applications and users to meet varying access requirement. A flexible model facilitates convenient additions and modifications to the existing structure. Zeiler (2000) proposed that the indications of a logical database model fulfilling the design requirements can be derived when the following questions are answered in the affirmative.

- Does the database model represent all data without duplication?
- Does the database model support an organization's business rule?
- Does the database model accommodate different views of data for distinct group of users?

Based on the criteria for database model evaluation proposed by Simsion & Witt (2005), in this study the new geotechnical database model is measured by the following factors:

- **Completeness**
The geotechnical database model is checked if it supports all the necessary data. Omissions of required data should be identified and amended.
- **Nonredundancy**
The redundancy of geotechnical data is checked to avoid recording the same fact more than once.

- **Enforcement of business rules**

The geotechnical database model should reflect and enforce the rules that apply to the geotechnical data.

- **Data reusability**

As far as possible, the geotechnical database model should be organized independent of any specific applications.

- **Stability and flexibility**

A database model is stable in the face of a change to requirements if it needs not be modified. A database model is flexible if it can be readily extended to accommodate likely new requirements with only minimal impact on the existing structure. The geotechnical database model is evaluated for the potential changes in requirements.

- **Elegance**

Elegant models are typically simple, consistent, and easily described and summarised. The geotechnical database model is evaluated to assess if it provides a reasonably neat and simple classification of the data.

- **Communication**

The geotechnical database model is checked to avoid the communication problems arising from high level of complexity, new concepts, and unfamiliar terminology.

2.4 Summary

Every database is constructed based on a database model, even if only an implicit one. A *database model* is a conceptual representation of the data structures that are required by a database. Database modelling is part of the conceptual design process. A database model can be evaluated against the criteria of correctness, consistency, accessibility and flexibility. The evaluation criteria for the proposed geotechnical database model are defined in the current chapter.

The database model is most efficiently described using model diagrams. The model diagrams can be created by ER diagrams or UML diagrams. In the subsequent chapter on literature review, most of the reviewed database models are described using ER diagrams. In this study, the proposed database model is described using UML diagram.

Chapter 3 Literature Review

The development of a new geotechnical database model requires a clear understanding of the geotechnical data and their internal relationships. A literature review on geotechnical database models is conducted in this chapter. The literature review is organized in the following sections: the first section goes through the different database models in the reported geotechnical databases, where the focus is to pinpoint the internal logic of geotechnical data identified by various researchers. Then an evaluation of the existing database models for supporting a 3D GIS application is conducted. One kernel component for a 3D GIS is the spatial analysis function (Houlding, 1994), how this function is supported by those database models is evaluated. A review on geostatistical analysis of geotechnical data is conducted in the final part of this chapter to obtain an understanding of the variogram data needed for spatial analysis. The state-of-the-art information derived from the review will provide the foundation and guidelines for the development of a new geotechnical database model.

3.1 Geotechnical Database Models

Reported geotechnical databases are comprehensively reviewed in this section. The database models are presented through model diagrams and documentations. These diagrams follow the general rules presented in Chapter 2. However, it should be noted that there may be minor variations in the notation used in these model diagrams.

dBase III for Changi South Bay (Lee et al., 1990)

The database system reported by Lee et al. (1990) is an example of file-based geotechnical database systems. The database is implemented using dBASE III running on a personnel computer. The data consists of more than 200 instruments of various types from the field instrumentation and test program of a land reclamation project in Changi South Bay, Singapore. The instrumentations include piezometers,

total stress cells, hydraulic settlement cells, inclinometers and about 200 in-situ tests. The data are organized into three components in the database: data files, a characteristic file and code files. All the readings obtained from measurements and instrumentations are stored in the data files and each data file contains readings only from one instrument or one test. The data describing the conditions under which the readings are collected are stored in a characteristic file. The code files contain the standard descriptor of attributes that can be used in the characteristic file and a logical flag indicating whether the descriptor has been used in the characteristic file.

The kernel of this logical database model is the characteristic file. Some important attributes identified in the model are: the transducer or equipment type, transducer or test identifier, associated transducer type and identifier, data organization, data types and units, location of device or test, test date and data file name. The data are accessed through these characteristic attributes. For instance, if the test identifier of a cone penetration test is known as DCP001, the data file containing its data can be identified by searching for the characteristic record using the test identifier as search key. Although the location of the device recorded the Cartesian coordinates relative to a predefined origin, they are only used as a supplemental search criterion for data retrieval. The data are output either in the form of a .DBF file (dBASE format) or exported to an ASCII text file; there is no tool to use data spatially and so naturally the spatial component of the data is generally not a major concern. As an early effort to manage geotechnical data using a database, the file-based database is now seldom adopted in new database developments.

GeoTec Database of SIGMA (Oliver & Toll, 1996)

A site investigation database GeoTec is developed using a relational database model as a core database to support the System for the Interpretation of Geotechnical Information (Toll, 1995; Oliver & Toll, 1996). Figure 3-1 shows the database structure of the GeoTec database, where the boxes represent tables in the relational database. The detailed descriptions of legends for the database tables are given in Figure 3-2.

Figure 3-1 Schema for the GeoTech database structure (after Oliver & Toll, 1996)

Legend			
proj	Project	drem	Depth Related Remarks
hole	Borehole/Trial Pit	core	Core
geol	Geology	frac	Fracture Detail
lay	Layer	ptim	Progress with Time
strt	Strata	wstk	Water Strike
stst	Stratum Structure	samp	Sample
cnst	Constituent	spdt	Specimen Detail
ctcl	Constituent Colour	isrf	In-Situ Test Index
tsrf	Test Reference	lbrf	Laboratory Test Index

Legend-Laboratory Tests			
cbrg	California Bearing Ratio-General	cbrt	California Bearing Ratio-Detail
cmpg	Compaction Tests-General	cmpt	Compaction Tests-Detail
mcvg	Moisture Condition Value- General	mcvt	Moisture Condition Value- Detail
shbg	Shear Box Testing- General	shbt	Shear Box Testing- Detail
cong	Consolidation Tests- General	cons	Consolidation Tests- Detail
trig	Triaxial Tests	trix	Triaxial Tests- Detail
clss	Classification Tests- General	grad	Particle Size Distribution
reld	Relative Density	chem	Chemical Properties
chlk	Chalk Crushing Value	ptst	Laboratory Permeability Tests
rock	Rock Properties		

Legend- In-Situ Tests			
icbr	In-Situ California Bearing Ratio	iden	In-Situ Density
iprm	In-Situ Permeability	irdx	In-Situ Redox
ires	In-Situ Resistivity	ispt	Standard Penetration Test
stcn	Static Cone Penetration	conc	Cone Calibration
pobs	Piezometer	pref	Piezometer- Detail

Figure 3-2 Legend for GeoTech database structure (after Oliver & Toll, 1996)

One specific characteristic of this data structure is introducing a geology table *geol* to store the identified geological horizons which could exist at the site. The information are obtained from a desk study at the feasibility stage of the project before any ground investigation has started, and can be referred by specific layers identified at a later stage, either during or after the ground investigation. Another characteristic of this database model is the structuring of the soil description in the relational table. The layer table *lay* contains a text field for the layer description. To more efficiently access this qualitative information, a data structure has been

defined in which the information contained in the description can be extracted and stored as separated field within the database. Four tables are introduced to structure the soil descriptions: stratum table *strt*, stratum structure table *stst*, colour table *ctcl* and constituent table *cnst*. This structure can handle the possibility of multiple strata within a layer, such as *sandstone interbedded with siltstone*, it can also deal with the case where a soil or rock changes significantly from the top to the base of a layer, such as silty sand becoming clayey sand. Based on the soil layer description, a module is developed to enable the interpretation of the underground condition and build up the soil profiles between different boreholes. By moving the emphasis to soil stratigraphy, the GeoTec database model reflects the attempt to emphasize the soil layer description in the geotechnical database model.

NGES-National Geotechnical Experimentation Sites Database (Satyanarayana & Benoit, 2001)

The NGES database is developed with a view to provide a central data repository for the data acquired at the NGES sites. The relational database includes 90 tables and resides in a SYBASE SQL Server operating under UNIX environment. The server is hosted at the Federal Highway Administration Turner-Fairbank Highway Research Center in McLean, Virginia.

For each NGES site, the test results are divided into either laboratory or in-situ test. Consequently, every test is assigned a database table with the database fields representing the test data. All these tables are in-turn tied to a test hole or an in-situ borehole. The simplified structure of the database is shown in Figure 3-3. The tables are briefly described below:

- **NGES sites**

This is a table containing the list of all NGES sites. More than forty sites are included in the NGES system. Every site has multiple boreholes and/or test holes.

Figure 3-3 NGES database structure (after Satyanarayana & Benoit, 2001)

- **Boreholes**

The data pertaining to the boreholes/test holes are stored in this table. These data describes the borehole attributes such as borehole name, final depth and hole diameter.

- **In-situ tests**

It represents all the available in-situ test methods (e.g., SPT test). Each in-situ test method has its own table. At each borehole, at least one in-situ test has been performed and thus data are available for that in-situ test.

- **Specimen**

This table represents a single table of soil samples, both disturbed and undisturbed, collected for laboratory tests.

- **Lab tests**

This table represents all the available laboratory test methods (e.g., Atterberg Limits, Sieve Analysis, etc.). Each laboratory test method has its own table, and for each specimen at least one laboratory test has been conducted and the data are stored in the corresponding laboratory test table.

All these tables are tied together using foreign keys and primary keys according to the relational format. By using those foreign keys, the consistency of the database is maintained. As an example, one could not enter data into any in-situ test table without the corresponding information in the boreholes table. Similarly, one could not delete borehole information from the borehole table without deleting corresponding information from the specimen, in-situ tests, or lab tests tables.

The NGES database is developed to provide research quality information. It provides a high level of details for geotechnical information. The level of details would be a major consideration in this data modelling process as more detail data are needed for research works. Another feature of the NGES database is its query interface which starts with the site location. Although in this database model the site location is represented by a text field (Texas, California et al.), it represents a natural way of accessing the data. If a spatial analysis tool like GIS were available, obviously the data retrieval process would be more flexible and effective.

Interactive Database of Soil Strength Parameters (Ansary et al. 1999)

Ansary et al. (1999) described a database where the geotechnical data were collected from different government and non-government organizations. Three tables are created in a Microsoft Access database, the name of the three tables are Places, Soil properties and Soil raw data. The data model is shown in Figure 3-4. The detailed descriptions for the fields contained in each table are listed in Table 3-1. The objective for developing this database is to train an artificial neural network for soil parameters estimation. The development of spatial analysis tool gives rise to the requirement for spatial dataset. To train the neural network, a query is

Figure 3-4 Relationship model for a geotechnical database proposed by Ansary et al. (1999)

Table 3-1 Field Description of the relationship database (After Ansary et al, 1999)

<i>Place table</i>	
Field	Description
PLACE_ID	Autonumber – Number automatically assigned to the new place-identification of the place
LOCAT_NM	Location name - Name of the place/Address
X	Easting – X-coordinate of the place
Y	Northing – Y-coordinate of the place
LCODE_NO	Code number of the place
TOTAL_BH	Total number of boreholes
BH_USED	Number of borehole used in table <i>Soil_properties</i>
FIN_YEAR	Financial year in which the test is performed
TESTED_BY	The agency by which the test is performed
CLIENT	The name of the client
REMARKS	Comments

<i>Soil properties table</i>	
Field	Description
ID	Autonumber
PLACE_ID	Identification number of the place (same entry as in <i>Place</i> table)
DEPTH	Depth of the soil property test
DEPTH_CODE	Code of the depth.
TYPE	Type of the test
BH_NO	the number of borehole of the defined place
SL_PER_BH	The serial number of the test per borehole number
CODE_NO	It is composed of [DEPTH_CODE] + [TYPE] + [BH_NO] + [SL_PER_BH]
GS	Specific gravity
W%	Water content
D_DENS	Dry density
SPT	Standard penetration test N value
LL	Liquid limit
PL	Plastic limit
SL	Shrinkage Limit
X-OPT :	Optimum or effective X-value from graph
Y-OPT	Optimum or effective Y-value from graph

Table 3-1 Field Description of the relationship database (after Ansary et al, 1999)
(Continued)

<i>Soil properties table</i>	
Field	Description
TYPE_CURVE	OLE object – A picture representing the test results
SAND%	% of sand in the soil sample
SILT%	% of silt in the soil sample
CLAY%	% of clay in the soil sample
RESULT_1	C-value for shear strength test and so on
RESULT_2	ϕ -value for shear strength test and so on
UNIT	S. I., F. P. S. or M. K. S.
REMARKS	Comments

<i>Soil raw data table</i>	
Field	Description
DATA_ID	Auto-number
PLACE_ID	Same entry as in <i>Soil_properties</i> table
CODE_NO	Same entry as in <i>Soil_properties</i> table
DEPTH	Same entry as in <i>Soil_properties</i> table
TYPE	Same entry as in <i>Soil_properties</i> table
1st_VECTOR	Raw data of soil tests taken along X-axis
2nd_VECTOR	Raw data of soil tests taken along Y-axis
REMARKS	Comments

developed to generate Standard Penetration Test (SPT) dataset. The dataset consists of the SPT values together with their spatial component: Easting, Northing and depth. Although this database model is quite simple and only applied to SPT data as reported, it represents a typical example of generating a spatial dataset from the database. One disadvantage of the database model is that the SPT data could not be further classified based on soil layers. Supposing the SPT data were obtained from two different soil layers, by separating the dataset into two subsets based on the soil layers and conducting the analysis separately, the estimation results would be more rigorous than using the dataset as a whole. Obviously, this could not be achieved using the current database model.

USGS Geologic Site Database (Ponti, 2001)

To support the Los Angeles Basin and Santa Clara Valley projects of the U.S. Geological Survey (USGS), a geological site database is developed. Over 6 GB of data from the Los Angeles Basin, Santa Clara Valley, and Monterey Bay resides on a server located in Menlo Park, CA. The relational database model consists of 63 tables. Twenty eight tables contain actual geological data with the remaining tables serving as lookups and for standardizing input and generating reports. Figure 3-5 shows the major tables in the database model.

In this database model a site is defined as a point in space (the origin - usually found at the ground surface but this is not required), plus a path that extended from the origin. The path could have any positive length, including zero. Sites could be of various types (outcrop, well, test hole, trench, geophysical profile, etc.) and could have any type of path geometry. The *Site* table is the primary parent file in the database. It contains a primary key field, a unique 20-character text field that is automatically generated when a site record is created, along with fields that track when and by whom a record is created or modified. All data records in other tables in the database (except for lookup tables) contain a foreign key field which is related to the site record as well as similar modification tracking fields. The *Site* table also contains fields for the site name, information about the site location, and the coordinates of the origin. Both geographic (with a specified datum) and projected coordinates (for both State Plane and UTM projections) are stored, with the z coordinate specified as elevation above mean sea level (feet or meters in a specified vertical datum). In addition to the coordinate fields are fields that characterize the source, reliability and accuracy of the coordinate information. Information on the path geometry is contained in the related *Path* table. Every data element associated with a site is then defined at a given distance (or interval) along the path from the origin. In this fashion, the coordinate (x, y, z) of every data element associated with the site is defined. A path record exists for every defined vertex along a path and consists of four principal fields: distance from origin (feet or meters), x, y (projected coordinates in feet or meters) and z (feet or meters above

Figure 3-5 The kernel USGS geologic site database (after Ponti, 2001)

mean sea level). A site itself does not require path records. If no path records existed, the path is assumed to be vertical (extending in a positive fashion below the origin), with the path length defined by the total depth field in the *Site* table. The *Construction* table pertains only to well and test hole sites and holds information on the operator and construction history.

The *Samples* table contains information on geological samples collected for follow-up description or laboratory analysis. Information such as the date collected, collector, collection method and sample type is stored in the samples table along with the primary lithology, grain size and bulk density, if available. These values could be input directly into the records of the *Samples* table or obtained through a path that extended from the origin. Records in the samples table also contain a primary key field and serve as parents to records stored in various analysis tables. The samples table also contain foreign key fields to associate the records with core samples or wells (if appropriate), and with the site. The descriptive tables (*Lithology*, *Accessories*, *Color*, *Remarks*, and *Graphics*) contain most of the non-parametric geologic information contained in the database. The *Lithology* table stores the lithologic information. Various kinds of observation data useful for the identification of critical surface and depositional facies are stored in the *Accessories* data table. The sediment color information is stored in the *Color* table. Comment on import features, observations and conditions that cannot be adequately represented elsewhere is stored in the *Remark* table. The *Graphics* table contains a field to hold graphic data or provides a reference to graphic files.

Direct measurements of various kinds at specific intervals along the site path are stored in the *In Situ-Measurements* table. This table contains a set of fields that store parameter information, measurement type, measurement value, measurement units, and graphics. This table stores oxidation-reduction, thermal conductivity, and unconfined compressive strength data (derived from pocket penetrometer) obtained from core samples. However, the table format is flexible enough to store most other in-situ test data that are not obtained from cores, such as standard penetration test data.

Several analysis tables are developed to store relevant results of various laboratory analyses that are performed on samples collected from a site. Tables are also developed to store soil moisture content, cation, anion, salinity and pH data from pore waters, fossil lists and interpretations, results of numerical dating analyses, and

paleomagnetic results. Other analytical tables could also be developed as the need arose.

This database is created to support the development of 3D stratigraphic and physical properties models of Quaternary basin sediment. It is realized that “geologic information was inherently descriptive, often not standardized, and might contain a significant amount of interpretation that could conceal important fundamental observation and analysis”. Naturally, the database model includes a large amount of descriptive data and interpretive data. It is identified that “there is a current need to better integrate the site database with commercial modeling and geologic analysis software”. A study is under-taken to try to directly tap into the site database, quickly create geologic cross-sections that would facilitate sequence-stratigraphic correlations, and to create horizontal records that could be exported into other types of geological modeling software.

A Geotechnical Database for Seismic Hazards Mapping in California (Real & McGuire, 2001)

For the purpose of liquefaction and landslide hazard zoning under California’s Seismic Hazards Mapping Act of 1990, a geotechnical database has been under continuous development to provide a means of managing borehole logs and laboratory soil tests for thousands of geotechnical borings used in the analysis of seismic hazard in both Northern and Southern California (Real & McGuire, 2001). The database contains over 500 tables and one million records. They reside on a UNIX server and are managed by Informix’s Standard Engine. There are seven tables used to represent the information contained in a geotechnical borehole log. Figure 3-6 shows the entities, their relationships, and the attributes of each entity.

Figure 3-6 Entity relationship diagram for distributed geotechnical data (after Real & McGuire, 2001)

The following is a brief description of the entities and their relationships.

- **Sample_location**

This table is used to store general information about the location where the subsurface investigation is made. This includes items such as spatial coordinates, reference elevation, total depth, type of investigation (borehole, test pit, trench, etc), the operator conducting the investigation, the USGS quad and any street address information. Each investigation is assigned a unique name (*official_name*) which is based on a batch number, a sequence number within the batch and the USGS code where the investigation is located in. This is the parent table to all other tables in the model.

- **Excavation_compl**

This table stores information about how the well is completed, for example, the drilling date and method, the hole diameter, casing and fill type etc. For each record in *Sample_location*, there would be zero or one record in *Excavation_compl*.

- **Fluid_pen**

This table stores information about the fluid penetrations found in a well. Since a well might have more than one fluid contact, the combination of *official_name*, *fluid_name* and *top_depth* must be unique.

- **Lithology**

This table stores lithologic descriptions and qualifiers as recorded or interpreted from the log. For any well there would be one or more lithologic layers, so the combination of *official_name*, *lith_type*, and *top_depth* must be unique.

- **Sample_data**

This table captures the basic information about the sample intervals recorded for a log such as type of sampler and recovery. Each sample interval is assigned a sample id and the top and bottom depths as interpreted from the log are recorded. The combination of *official_name*, *sample_id* and

top_depth must be unique.

- **Sample_analysis**

This table stores information that describes the results of any laboratory tests that are run on the sample intervals. This table is a child table to *Sample_data*, and for each record in *Sample_data* there might be zero or more records in *Sample_analysis*, i.e. one sample interval might be analyzed for dry density, moisture content, grain size, etc. The combination of *official_name*, *sample_id*, and *parameter_name* must be unique.

- **Pen_test_results**

This table captures the results of any penetration tests such as SPT conducted in the boring along with several parameters of the penetration test itself, such as hammer type, hammer mass, etc. It is also a child table to *Sample_data* and the combination of *official_name* and *sample_id* must be unique.

The database model is developed to supply a GIS application mapping the seismic hazard. Implementation of GIS tools generates new category criteria for the geotechnical data. The data could be categorized not only as basic data, derived data and interpreted data as usual, they could also be categorized based on spatial dimensions. In this model, the sample location, lithology & sample data could all be represented as a spatial point in a GIS. The database model fulfills its development requirement, although the spatial component of the geotechnical data has not been fully explored.

ROSRINE Database (Nigbor et al., 2001)

ROSRINE (Resolution of Site Response Issues from the Northridge Earthquake) is a government-academic-industry research collaboration aims at improving engineering models of earthquake ground motion through the collection, synthesis, and dissemination of data on subsurface conditions at key Strong Motion (SM) sites. It includes geological and geophysical characterization of around 50 SM sites in Southern California.

The ROSRINE database contains all the information from site investigation boreholes, laboratory experiments and other related data. The database comprises ten major tables. Figure 3-7 shows the relationship between these tables. The database structure is designed to be compatible with GIS. The database supports one-to-one and one-to-many relationships between the tables. Its data structure is created by trial and error. It is thought to be general enough to be applied to many other types of spatial data, provided they are collected at scattered points.

The database identifies four main entities (or objects) and they are categorized in six main tables:

- *Sites*
- *Strong Motion Stations*
- *Data Files*
- *References*
- *Address Book*
- *Data Types*

The concept of sites implies a small area centered about geographical coordinates, which yields data either from field investigations, laboratory investigations, or strong motion recordings. Table *Sites* describes a site using geographical coordinates; a short and full name for identifying location; surface geology and site type which indicates that a site might be a ROSRINE investigation site or another type of site.

Table *Sites* and *Strong Motion Stations* are the only tables with geographical coordinates, and are therefore the only connections to GIS spatial queries. Table *Strong Motion Stations* is similar in structure to Table *Sites*.

Figure 3-7 Tables and relationships between tables in ROSRINE database (after Nigbor et al., 2001)

The second major type of ROSRINE entity is *Data Files*. These files are the main items to be distributed to the public. Data files are organized in a catalog, characterized using a few attributes, including type of investigation, virtual or physical location on hard drive server, some textual description, time stamps, project phase, and depth of sampling in case of field samples extracted from boreholes. All data files are organized in directories on the server, and are assigned a virtual address or URL for HTML file retrieval. Table *Data Types* completes table *Data Files*. It contains various types of main activities conducted during the ROSRINE project such as shear wave velocity measurement and laboratory investigations. So far, there are nine types of activities which have generated data files:

- (1) Boring Log;
- (2) USGS Downhole P&S Wave Velocity;
- (3) ROSRINE Suspension P&S Wave Velocity;
- (4) ROSRINE Downhole P&S Wave Velocity;
- (5) ROSRINE Electric Log;
- (6) Laboratory Tests - Index Properties;
- (7) Laboratory Tests - Dynamic Properties - University of Texas at Austin / Stokoe;
- (8) Laboratory Tests - Dynamic Properties - UCLA / Vucetic;
- (9) Geotechnical Site Model.

The two other tables *Address Book* and *References* are common to many databases. *Address Book* defines the names and addresses of individuals who have contributed to the project. *References* table contains all the bibliographic references which are related to a data file. *References* add information to *Sites Strong Motion Stations* and *Data Files*. For instance, *References* identify the method used to generate data in a data file, or provide a list of hardcopy reports relevant to a particular site. References could also point to hardcopy reports and/or URL addresses.

The ROSRINE database handles one-to-many relationships by introducing four list tables: *Data File List*, *Data File Sublist*, *Reference List*, and *Address List*. These tables do not contain any new information except information links, or indexes. If in the future it was desirable to add additional data files such as strong motion records into the database, a new list table could easily be added as an index to table *Strong Motion Stations*. They are all constructed based on the same principle. A particular data file might have originated from several references. A one-to-many relationship could be described using a list. A particular data file could be associated with several entries in the table *Reference*. The same principle applies directly to *Data Files List* and *Address List*, which permits one site to be related to multiple data files, individuals, and agencies.

Table *Data Files Sublist* is a list table describing the relationships between data files. A site might have some data files related to several other data files. For instance, a borehole data file may be related to several laboratory investigation data files.

The ROSRINE database is also integrated with a GIS. The Strong Motion Sites and Investigation Sites are represented as point symbols in the GIS application and the attributes of the two tables could be displayed with the points. It is just a primitive integration and the database model does not support complex spatial analysis. In fact it has been found that the “GIS map-based query had limited flexibility in querying the database. No queries could penetrate data file and read their contents as those files were treated as hermetic entities that barred further queries” (Nigbor et al., 2001, pp. 47).

A Database for Island of Oahu, Hawaii (Oloufa et al., 1992; Oloufa et al., 1994)

A database had been integrated with a GIS to manage the site construction investigation on the island of Oahu, Hawaii (Oloufa et al., 1992; Oloufa et al., 1994). The database model is presented in Figure 3-8. There are four entities in this model, project *Reference*, *Borehole*, *Sample* and *Layer*, as listed below:

- **Reference**

The *Reference* entity provides a uniform referencing scheme for all the reports available and includes information about the referenced project document in order to allow its identification.

- **Borehole**

Boreholes here are defined as investigation points drilled to retrieve information about the subsurface soil. This entity provides information about the boreholes in the reference from which the subsoil information is taken.

- **Layers**

A *Layer* here represents a soil stratum. The collection of a vertical ordering of layers forms a boring log. Each layer gives information about the soil stratum it represents.

- **Samples**

A *Sample* is an extraction of soil material from a borehole. *Samples* are used in tests to determine the values of soil properties and parameters. Such values are reported in the *Sample* entity

In the *Layer* entity there are values for fields that provided description of the soil encountered in each layer. These fields included descriptor of material type, density, stiffness, inclusions, origin of material, weathering conditions and other descriptors. One specific feature of this database model is that a dictionary list for these descriptors has been predefined and the user is advised to choose from the list rather than using their own terms. The GIS is tailored to display the borehole with the descriptors.

Figure 3-8 Data model for site investigation (after Oloufa et al., 1994)

WELLHEAD for Dane County, Wisconsin (Adams et al., 1993)

Rather than representing a borehole as a point in GIS applications, the WELLHEAD is a GIS application that enabled users to view and to retrieve subsurface data interactively and graphically (Adams et al., 1993). The system is developed for storage of borehole data available in Dane County, Wisconsin. It has utilities to allow interactive input of log data and representing subsurface by surface maps and fencing diagrams. Figure 3-9 shows the WELLHEAD database model. In each boring log there are a set of soil segments and a set of samples. Depending on whether the sample is rock, peat, or an inorganic soil, each sample might be tested for some or all of its soil properties such as permeability, water content, and trace elements.

The extent of the subsurface strata is limited to what is obtained from borehole drilling and individual borehole log analysis. No full 3D subsurface site representation is available. It is concluded that to achieve a full 3D subsurface representation would require topological or solid modelling of the subsurface strata based on 3-D fence diagram and the interpolation of knowledge for site characterization.

GeSEP of Bangkok Subsoil (Suwanwiwattana et al., 2001)

A database system is developed as part of a geotechnical information system for the Bangkok subsoils in Thailand. The relational database PostgreSQL is used as the database server running on the Linux system and it is connected to the Kasetsart University network (Suwanwiwattana et al., 2001). The GeSEP (Geotechnical Subsoil Explorer Project) data model is presented in Figure 3-10. The database tables are classified using four groups:

- **General borehole information relation group**

This group includes tables for storing information about borehole identification, location, project name, user and owner of the borehole, test method and other information.

*Figure 3-9 Hierarchic representation of WELLHEAD Boring Log Database
(after Adams et al., 1993)*

- ***Wash boring information relation group***

This group contains all the selected laboratory tests, and all tables are related through the borehole identification number and the sample depth.

- ***Field Vane shear relation***

This table stores the in-situ test results of field vane and is connected to the general information group through the borehole identification number.

- ***Cone penetration relation***

This table holds the cone penetration test results and is also connected to the general information through the borehole identification number.

This database is integrated with a GIS to generate a 3D soil consistency map. To classify the soil consistency, criteria are set up according to the SPT blows counts and the type of soil (sand or clay). However, the specific criteria need further validation and modification when they were applied to other areas.

Figure 3-10 GeSEP data model (after Suwanwittatana et al., 2001)

3.2 Applicability of the Reviewed Database Models for 3D GIS

There are three components for a GIS application: map interface, spatial analysis and database (Hellowell et al., 2001). A 3D GIS application involves the 3D spatial analysis of geotechnical data and the representation of the geotechnical data in a graphical interface. It hence requires a database model that supports spatial analysis and 3D presentation. The data used in the spatial analysis and 3D GIS presentation also need to be stored and managed by the database.

For a user of a GIS application, it is desirable that the user is not only able to retrieve soil information from existing boreholes, but is also able to retrieve soil information from location with no existing boreholes. The later function is especially helpful for those users who are interested in the site conditions but lack the geotechnical skill to interpolate from the existing borehole data. To support such a GIS application, the database model should support the following functions:

- **Spatial definition of the underground stratigraphy**

To estimate soil properties at any un-sampled location, the category of this soil at the location must be derived (For instance, the soil at that specific location is categorized as a marine clay). Hence, the underground stratigraphy must be defined in the database.

- **Spatial dataset generation**

The existing borehole data are the basis for estimating the soil properties at un-sampled location. The database model should support the generation of a dataset based on the soil category which the soil at the un-sampled location (For instance, a dataset would be all the field vane test readings in the marine clay layer).

- **Spatial description**

Spatial descriptions enable user to build up an interpolation model to estimate unknown value from the spatial dataset derived from existing borehole data.

Using the previous condition as criteria, the reviewed database models are listed in Table 3.2. For most of the reviewed databases, the data are normally retrieved based on project and borehole identifiers, the spatial components of geotechnical data are either ignored or used as a supplemental data retrieval criterion. For instance, in the NGES database, the *Site* table has a text attribute to record the site location, which is used as a supplemental criterion for data retrieval. Spatial analysis raises the requirement for a spatial dataset, the database reported by Ansary et al. (1999) presented a preliminary solution to cope with such a requirement. The early application of GIS generally involved using maps to access borehole through a identifiers, the database was integrated with the GIS through a link between the database table and the GIS point attribute table, the spatial functions provided by GIS tools were not fully explored. In view of this, the reviewed database models were not suitable for 3D GIS application. A new database model is needed.

By reviewing existing database models, it was found that different data requirements existed and the database model had to evolve to cope with the database requirements. To support a 3D GIS application, the data in the database model needs to be extended for incorporating the requirement of 3D spatial analysis. The geotechnical database should not only focus on the physical characteristics and mechanical characteristics of the geotechnical data, but also their spatial characteristics. One commonly used soil parameter for spatial variation description and spatial geostatistical analysis is the variogram. The following section briefly reviews the variogram autocorrelation study on geotechnical data. The purpose of this review is to obtain the general characteristics of variogram for geotechnical data.

Table 3-2 List of reviewed database models

Database model	Spatial dataset generation	Spatial description	Spatial stratigraphy definition
<i>dBase III for Changi South Bay</i> (Lee et al., 1990)	No	No	No
<i>GeoTec Database of SIGMA</i> (Oliver & Toll, 1996)	Yes	No	No
<i>NGES-National Geotechnical Experimentation Sites Database</i> (Satyanarayana & Benoit, 2001)	No	No	No
<i>Interactive Database of Soil Strength Parameters</i> (Ansary et al. 1999)	Yes	No	No
<i>USGS Geologic Site Database</i> (Ponti, 2001)	Yes	No	No
<i>A Geotechnical Database for Seismic Hazards Mapping in California</i> (Real & McGuire, 2001)	Yes	No	No
<i>ROSRINE Database</i> (Nigbor et al., 2001)	unsure	No	No
<i>A Database for Island of Oahu, Hawaii</i> (Oloufa et al., 1992; Oloufa et al., 1994)	Yes	No	No
<i>WELLHEAD for Dane County, Wisconsin</i> (Adams et al., 1993)	Yes	No	No
<i>GeSEP of Bangkok Subsoil</i> (Suwanwiwattana et al., 2001)	No	No	No

3.3 Variogram of Geotechnical Data

3.3.1 Definition of Variogram

Geotechnical data are associated with a specific spatial location in the ground. Spatial continuity exists in most earth science data sets. For a geotechnical attribute which exhibits continuous variation, the attribute values at two points close to each other are more likely to be similar than the attribute values from two points that are far apart. This spatial continuity can be quantitatively summarised through some indices. The variogram is one such commonly used parameter (Isaaks & Srivastava, 1989). The variogram measures the average dissimilarity between data separated by a separation vector h . A variogram is defined as the average squared difference between two attribute values and is expressed as a function of the separation vector h . Equation 3-1 gives the mathematical expression for the variogram $\gamma(h)$:

$$\gamma(h) = \frac{1}{2N(h)} \sum_{\alpha=1}^{N(h)} [z(u_{\alpha}) - z(u_{\alpha} + h)]^2 \quad (3-1)$$

where $N(h)$ is the number of pairs of data, u_{α} is the location, and $z(u_{\alpha})$ is the data value at location u_{α} .

From a set of samples of a reasonable sample size, the *variograms* for a limited number of separation distances can be obtained. To obtain the *variogram* at any separation distance, a model needs to be inferred for computing a *variogram* value for any possible separation distance. The *variogram* can be used as a descriptive tool for spatial continuity. More importantly, it can be further used for geostatistical estimation (Isaaks & Srivastava, 1989; Goovaerts, 1997).

3.3.2 Variogram Studies on Geotechnical Data

The early work on spatial autocorrelation and regionalized variables goes back to the 1960s. It was applied first to geochemistry, but quickly to soil science with the development and availability of spatial analysis tools (Matheron, 1970; Cliff & Ord, 1981; Oliver & Webster, 1990). In general, the information of soil variability is important for various decision making processes commonly encountered in geotechnical engineering as suggested by Bekele (2001). The potential applications include:

- Designing suitable sampling schemes;
- Determining the scaling behaviour of soil properties;
- The appropriate design and analysis of field experiments;
- Predicting soil properties at unsampled locations; and,
- Inferring processes that might be related to the causes of variability.

Ali et al. (1989) reported a study of the collapse-susceptible soils in Tucson,

Arizona, using geostatistics of soil parameters from 414 locations in the city. These parameters included dry unit weight, moisture content, void ratio, porosity, degree of saturation, plastic limit, sample depth and collapse criteria. Variograms were estimated for each parameter by using a discrete number of values obtained from test data. The derived variogram data were then used in conjunction with ordinary kriging method to estimate the parameter values at unsampled locations. The estimation results were further used to generate a prediction plot of high probability of occurrence of soils having a high collapse potential in this area. The applicability of geostatistical approach was demonstrated through the estimation results and plotted output.

Soulie et al. (1990) conducted a study on the structure of spatial variability of undrained shear strength within a clay deposit. Field vane test results from a site in the James Bay area of Quebec, Canada were investigated. A stationary model with a constant mean was used. Spherical and exponential variogram model (see section 4.4.1) were applied and both reflected the main characteristics of the spatial variability. By assuming a normal distribution of shear strength, a probability analysis for the detection of weak zones was conducted using the variogram data derived. It was concluded that the variogram was an effective descriptor of the spatial variability of the undrained shear strength of the clay in the study area. By the use of the variogram, it was possible to make local estimations of the undrained shear strength at locations within the deposits where vane measurements were not available.

Moustafa & Yomota (1998) used a spherical model to fit the experimental variograms of soil properties of samples collected in an area of 1,533 ha located in the western portion of the Nile Delta of Egypt. The soil parameters included hydraulic conductivity, water table depth and soil salinity. The variogram model was chosen by a validity test, which tested the two conditions of the estimation error: unbiasedness condition and minimum variance condition. A kriged hydraulic conductivity map was generated using the derived variogram to help in designing a

suitable drainage system. The study demonstrated the value of spatial modelling of soil properties as a useful tool to estimate the soil property at unsampled locations.

Honjo & Kuroda (1991) found that the spatial variation of the uniaxial compressive strength of soil at north Tokyo could be modelled using an exponential variogram model vertically and horizontally. The fitted variogram function was used to develop a reliability-based design process for slope stability assessment for an embankment. It was concluded that by using the spatial variogram, not only the values of geotechnical sample data, but also the locations of sample data were taken into account in the design process.

In a study about the spatial variability of a lightly overconsolidated and highly sensitive clay deposit located near Montreal (Chiasson et al., 1995), the vertical variabilities of field vane test and piezocone test results were modelled using a trend and a spherical model. The standardize statistical distribution for both testing devices were found to have similar shapes. A linear vertical trend was found. However, it was not unique to the site. The data were de-trended using a local trend. The covariance derived had a sill and showed linear behaviour near the origin. Meanwhile, the covariance had a 2m autocorrelation distance and lacked a nugget effect. Both testing devices yielded the same spatial covariance and statistical distribution. Chiasson et al. (1995) concluded that “the assessed variability, which is the adopted representation, is not dependent on the operating mode of the testing device.”

Li & Hayashi (1999) used a probabilistic model to determine the borehole spacing and spatial variability of soil properties during site investigation. The correlation distance, variance, and other statistics of soil properties such as undrained shear strength, soil layer thickness had been established based on a geotechnical database of Saga plain, Japan. The correlation distance of layer thickness and undrained shear strength varied in different directions within the soft ground. The use of a database appeared the best way to obtain these statistical parameters. Using the

extensive derived spatial variability for the Saga plain, the model could be used to decide the optimal spacing of site exploration according to predefined estimation error.

Przewłócki (2000) described a horizontal variation study of field vane test on a uniform clay located on Bysewo, Borland. A second order surface trend was found from the obtained data. After de-trending, autocorrelations were obtained by dividing the covariance by the variance. Using a Loss function (a sum of squared deviation) as a criterion of the best fitting, several correlation functions were obtained to fit the results.

In general, despite the variability of soil parameters and characteristic associated with the specific site conditions, very often, it was found that a basic model may be sufficient to describe the soil property variation. As in the case reported by Soulie et al. (1990), most studies were conducted in two dimensions and a horizontal isotropy assumption was used. The issue related to variogram data management was seldom addressed in the review studies, although Fenton (1999, pp. 470) suggested that “most reliability based design will benefit from a database of second-order soil statistics”.

3.4 Summary

Database Management System (DBMS) has been increasingly used in major civil engineering projects to manage geotechnical data since early 1990s. From a single user pc-based DBMS (Lee et al., 1990; Yu et al., 1999; Luna et al., 2001) to large multi-tier, multi-user DBMS (Wither et al., 2000), the geotechnical community has experienced and benefited from the rapid advances of information technology. Conventional geotechnical database models abstract geotechnical data in a predetermined domain and often ignore its spatial components. On the other hand, 3D GIS applications model the subsurface domain using spatial features. Although the spatial features could be interpreted from the geotechnical data, the conventional database models currently available are unsuitable for 3D GIS application. A new model is hence needed.

The variogram data functions as a descriptive tool for spatial continuity, and is also used for geostatistical estimation. The variogram data is normally bonded to a specific soil layer. The process of deriving variogram from sample data relies on a series of user's decisions, which are backed by experimental data or ancillary information. It also needs extensive data support and high performance computer analysis tools. However, the issue of creating a database for storing variogram data is seldom addressed. No detail on storing and dissemination of spatial description of soil properties has been found although Li & Hayashi (1999, pp.55) had suggested that "a database appears the best way to obtain this statistical parameter for a ground". The new database model should incorporate the variogram data for supporting the spatial analysis of 3D GIS applications.

Chapter 4 Geotechnical Data for the Proposed Database Model

3D GIS geotechnical applications function as tools for spatial analysis and present geotechnical data in a 3D frame. This chapter discusses the geotechnical data required for a geotechnical database supporting 3D GIS applications. This chapter begins with a description of the general process of geotechnical characterization, then the geotechnical data involved in this characterization process are discussed. The geotechnical data are grouped into three broad categories: Borehole data, Stratigraphy data and Spatial variation data. XML formats for encapsulating stratigraphy data and variogram data are also proposed in this chapter. In general, this chapter presents which geotechnical data will be modelled in the proposed database model and proposes a suitable data format for stratigraphy data and variogram data.

4.1 Geotechnical Characterization Process

The main purpose of a 3D geotechnical GIS is to serve as a decision support tool for geotechnical engineers to conduct geotechnical visualization and analysis. The geotechnical analysis may focus on predicting the spatial variation of key geological variables. Houlding (1994) has summarized a sequence of steps for the general geological/geotechnical analysis and characterization process. This clearly defined process is adopted in this study. Figure 4-1 presents a schema to summarize the generic process of geotechnical characterization and the steps are described below.

Management, correlation and integration of site investigation data

The root of the geotechnical characterization process is site investigation. Site investigation is an essential source of geotechnical data. The geotechnical

Figure 4-1 Summary of the generic process of geotechnical characterization (after Houlding, 1994)

characterization process hence starts with the organization and management of the various data obtained from site investigation.

Interpretation of geotechnical characteristics

This step involves the discretization and qualification of subsurface information into irregular volumes with similar distinguishing geological/geotechnical characteristics. The most frequently used controlling characteristic can be soil and rock type, or lithology. As the observations from site investigation are limited, geotechnical experience and intuition, and general knowledge of the geological history of a region are necessary ingredients in this interpretation process.

Analysis and prediction of geotechnical variables

The information sources are reviewed and analyzed in term of their quality, sufficiency, scale and spatial variability. The spatial variations of relevant variables are predicted based on sample information, geotechnical interpretation and appropriate prediction techniques.

Spatial analysis and presentations

Using the predicted variables and interpreted characteristics, spatial analysis can be conducted to support the visualization procedures and/or for transferring the results to other relevant end processes.

Geotechnical data are used and generated through the data interpretation, prediction and analysis steps. Three broad categories of geotechnical data will be described in detail. These three categories are:

- Borehole test data from site investigation;
- Stratigraphy data for geotechnical characteristic representation; and,
- Variogram data for spatial variation.

4.2 Borehole Test Data

The term borehole test data is used in this study to denote all the data derived from borehole observation and samples, either from in-situ tests or laboratory tests. Borehole test data are the key product of all site investigations. In a GIS application the borehole data are normally associated with points and intervals along a line or a trace with a 3D configuration, or associated with points, lines, polygons referenced to a plane with a 3D orientation (Houlding, 1994). Most of the GIS applications reviewed in the previous chapter in fact are focused on borehole test data.

As presented in the literature review, the geotechnical data derived from in-situ test and laboratory tests have been extensively illustrated and modeled in the conventional geotechnical database models. Generally there are two factors that need to be considered for the in-situ test data and laboratory test data. One is the kind of tests that need to be included in a database model. As numerous test types are available, it is almost impossible to create a database to cover exhaustively all the available test types. There is therefore a need to identify the geotechnical tests which are most frequently used, and provisions made in the database model for adding test types whenever necessary.

Another consideration is the accepted atomic level of detail, based on consideration of which of the test data can be categorized as “primitive data” and “derived data” (Oliver & Toll, 1996). In general, the “primitive data” are stored in the database. By retrieving and interpolating these geotechnical “primitive data”, different geotechnical outputs, such as borehole logs, section profiles, and reports, can be generated to enhance the decision-making process. The supplementary data and the interpolated output are usually stored outside the DBMS for permanence. For data which can be interpolated from the primitive data following some standard procedures without intensive computational demand, the redundancy can be reduced by storing only the primitive data. However, geotechnical data and information usually flow across multiple departments or organizations and are

called upon to play different roles at various stages. From a data lifetime perspective, the derived data may also function as primitive data for the downstream analysis process and thus need to be stored. For example, the undrained shear strength of clay could be treated as primitive data in a foundation analysis process but it also could be treated as the interpreted data of a stress-strain curve obtained in a laboratory test. The decision should be made to achieve a balance for an efficient database model. Based on these two considerations, in this study the database model adopts the well accepted geotechnical data structure: AGS (The British Association of Geotechnical Specialists) format for in-situ and laboratory geotechnical data.

The AGS format is published by The British Association of Geotechnical Specialists (AGS) for digital geotechnical data structure and transfer. It provides a means of transferring geotechnical and geo-environmental data by using the American Standard Code for Information Interchange (ASCII). The initial specification was conceived in 1991 and published in 1992, and now the third edition is available. The latest AGS specifications can be found on the AGS web site at www.ags.org.uk. With the continuous effort of the association, the AGS format has garnered a large user group and received support from many geotechnical software vendors. Now the specification is commonly used in the United Kingdom, Ireland, and Hong Kong for the electronic transfer of subsurface investigation data (Nicholls et al., 1996; Hutchison & Chandler, 1999). AGS categorizes geotechnical data into different groups that can be easily mapped into a database. Many geotechnical databases have been implemented based on the AGS format. Although the characteristics of AGS format may not always be favoured in all situations (McPhail, 2001), one advantage of AGS is that with two decades of development and evolution, the required geotechnical data and level of detail for transferring and sharing are well identified and clearly defined in the AGS format.

4.3 Stratigraphy Data

4.3.1 Surface Data and Soil Layer

A primary purpose of 3D GIS applications in the geotechnical field is to create a realistic representation for the spatial variation of the underground information. A reasonable representation of the geological/geotechnical features is required to enable the user to have a global view of the structure to be modeled. For such a representation to be feasible, simplification and idealization are inevitable. The underground domain can be subdivided into various soil layers by analyzing borehole data in conjunction with other relevant information. The soil layer can be represented by the solid volume or the bounding surface in GIS applications. In this study the focus is set on the bounding surface representation of soil layers. The modeling process can be thought of as a partition of the 3D space into regions, bounded by surfaces between different soil layers (Winkler et al., 1999).

Surface Data

A surface is the outermost boundary of an object. Mathematically a surface can be defined as a two-dimensional locus of points located in a three-dimensional space. It is a continuous field of values that may vary over an infinite number of points. Because a surface contains an infinite number of points, it is impossible to measure and store the z-value at every point. A mathematical surface model is employed to approximate a surface by taking a sample dataset of the values at different points on the surface and an estimator can be used to interpolate the values between these points. By vertically registering a series of surfaces, the information about a subsurface domain can be characterized and presented.

Raster and TIN (Triangulated Irregular Network) are two commonly used surface models in GIS. TIN represents a surface by a set of points linked to form a network of triangles and the elevation values are stored in the nodes of the tin. The

methodologies of using TIN to represent soil subsurfaces have been well described (Jones & Wright, 1993; Camp & Brown, 1993). The TIN is constructed directly from scattered data points representing soil layer encountered in boreholes. It stores the elevation values at the nodes to honour the data points. The TIN can also be easily reshaped in an interactive fashion by adding and removing data points. For elevation values that are not represented in the TIN node, an estimator can be created using interpolation algorithms such as the breakline bivariate quintic interpolation algorithm. These algorithms can be coded in procedures to interactively create any number of two-dimensional subsurface profiles as desired. However, the soil profile generated by these algorithms may be mathematically smooth but less rigorous in geotechnical meaning when the data nodes are sparse. A TIN is suitable for an exhaustive dataset, a scenario that is often not available in geotechnical practice.

A raster represents a surface in a two-dimensional, rectangular pixel grid. The grid refers to a set of x-y intersects. The elevation value z is stored as the content of the cells defined by two pairs of x-y grids (Zeiler, 2000). A raster surface may be created by interpolation from a limited number of sample data points. Different interpolation methods such as nearest neighbour, inverse distance weighted, spline and kriging are available. A raster surface can be generated by choosing a suitable interpolation method. In geotechnical study, due to the scarcity of available geotechnical data, a decision may be heavily based on the interpolation and extrapolation from a limited set of data. The major focus is hence given to the rigour of the interpolation methods. A raster format gives the flexibility to use the different interpolation methods or utilize their results. Within the raster data model, spatial calculation such as add, subtract, can be easily conducted between rasters with same resolution. This is desirable for geotechnical engineers who need to conduct a significant number of spatial calculations such as the evaluation of the volume for excavation, or the creation of a surface by combining several surfaces (merging rasters into a new raster). For a project conducted in several stages, the subsurface model can be developed in tempo with the development of the project,

and may keep on changing during the life-time of the project. By changing the resolution of the raster, a large study area can also be represented using low-resolution data. When required, it can also be subdivided into several sub-zones which can be represented using high resolution data. The major disadvantage of raster is the high storage requirement and high memory demand during processing. However, this disadvantage is partially offset by the rapidly declining unit cost of storage media. Another constraint of raster model is its inherent 2.5 dimension attribute; nevertheless it applies well in most situations where a unique elevation data exists. It also requires for more data than the TIN for equivalent representational fidelity of an elevation model.

Soil Layer

In GIS applications the geotechnical data are usually presented through thematic layers (Oloufa et al., 1992; Adams et al., 1993; Camp & Brown, 1993; Oloufa et al., 1994; Giles, 1994; Nathanail & Rosenbaum, 1998; Kimmance et al., 1999; Suwanwiwattana et al., 2001; Hellowell et al., 2001). Construction of such thematic layers needs queries based on a soil layer catalogue, like “select some PARAMETERS from some LAYERS, in the SCOPE of” to the database. Meanwhile, such kind of layer-based query is also needed when section profiles and reports are created.

One principle responsibility for a geotechnical engineer is to interpolate the information obtained in the drilled holes or soundings, and to categorize the subsoils into soil layers consisting of materials with approximately the same engineering properties. It is also necessary to estimate for each zone the average values of the pertinent soil parameters, all based on the fragmentary information obtained through site investigation (Terzaghi et al., 1996). When a borehole from a geotechnical site investigation is analyzed, stratum data are produced by classifying the soils based on visual inspection of retrieved specimens or from the interpolation of the sounding data. Each stratum identified in a borehole can be correlated to strata in other boreholes through the subsurface soil layer. The soil layer

classification combines findings from field observation, experience derived from previous projects, and engineering judgment. It provides an experience based, qualitative assessment of the categories for borehole stratum data. There could be different soil layer interpolation for a particular domain, and the diversity of classification for the subsoil domain is caused not only by the variability of the soil materials but also by the users through whom the classification is created. In that sense, the soil layer is a “concept”, and hence is subjective and personal. However, this issue is often ignored by most reported geotechnical databases. One major reason is that most of the time, the geotechnical database is project-oriented. The scope of a geotechnical project is relatively small and simple, and normally the soil layer classification schema is unique. This also explains why this particular issue first emerged in the geological field, which experiences a much more complex situation and covers bigger areas than the geotechnical field. However, problems can emerge in a database where data are sourced from a variety of projects and the classification criteria adopted may not necessarily be the same. Meanwhile, even in a single project, the soil layer classification schema can also evolve to give rise to multiple versions of interpolations; either because the data are collected through multiple-stages or they are interpreted by different engineers.

A soil layer can be represented as a domain partitioned by two surfaces: a top surface and a bottom surface. A spatial position in a 3D domain separated by these surfaces can be treated as a 3D point. If the point is not beyond the layer in x or y, the spatial relationship between the spatial position with a surface may have three options: this spatial point can be located right on the surface, otherwise it could be either above the surface or below the surface. Such a spatial relationship can be obtained using the basic GIS spatial analysis functions. Hence, the spatial relationship of a spatial point with respect to a soil layer can be logically inferred as follow:

- If this spatial point is above top surface or below the bottom surface, this spatial point is outside the soil layer,
- Otherwise this spatial point is inside the soil layer.

However, a spatial point inside a soil layer may not lead to the conclusion that this spatial point belongs to that soil layer category. This is because a soil layer may contain interbedded soil layers. For a soil layer containing interbedded soil layers, a spatial point belonging to this soil layer needs to be inside this soil layer but outside all the interbedded soil layers. So this soil category inferring process would be a process that recursively checks each soil layer and its interbedded soil layers till the soil category is identified. To facilitate this process, the relationships between different soil layers need to be defined.

4.3.2 Stratigraphy Data

The topological relationships between soil layers are defined in a stratigraphy model. Evaluation works for geotechnical or geo-environmental projects are normally based on the subsurface stratigraphy information that attempts to represent the reality of the domain under study as closely as necessary. Normally, the subsurface reality is represented and populated as two dimensional soil profiles not only because they are simple and easy to create but also due to the constraints of the traditional transmitting media such as hardcopy records. Most projects require preparation of a considerable number of cross-sections in order to develop the subsurface geotechnical description of a site and to resolve the associated design and construction issues. This is because the “grain” or principal direction of critical site conditions and project features is not always readily apparent or necessarily encountered in the first few cross-sections (Hamel, 1998). Moreover, sometimes these generated profiles may not meet the requirements of the geotechnical engineer. Very often the engineer needs to rebuild a three-dimensional subsurface model mentally based on these available profiles, relying heavily on his personal experience or expertise in the process. The process could be time consuming and may also import “twisted” or “prejudiced” information into the subsurface model.

In general, subsurface stratigraphy model building is an ambiguous and unstructured process. The level of uncertainty is often high because information is limited. Often, the available information is vague and open to multiple

interpretations. Detailed knowledge of the issue and understanding of its context is a prerequisite for any successful construction of a subsurface model. In a situation in which subjective judgment plays a major role, the interpretation and stratigraphy work often fall on the shoulder of a geotechnical specialist, by whom the information is summarized or selectively highlighted, taking into account the specific information needs of a project by absorbing uncertainties and widening its boundaries. Building a subsurface stratigraphy model inside a system can eliminate the unnecessary process to mentally recreate the subsurface model repeatedly. So once the subsurface model is created, no doubt with the input of an experienced geotechnical specialist, it can be reused. With the help of GIS, the production of soil profile sections will be more efficiently conducted and become more flexible. Meanwhile, the more powerful 3D representations can also be readily constructed. To build a subsurface stratigraphy model that defines the relationship between soil layers in an information system, the difficulties associated with representing human knowledge in the computer must be resolved. The former is usually vague and abstract and the latter needs to be rigid and structured. This calls for the development of a rigorous model to represent stratigraphy data.

4.3.3 XML, GeotechML, GeosciML, AGSML and DIGGSML

XML

The intrinsic logical structure of stratigraphy data requires an efficient format to store and present the information. XML emerges as a natural choice for such purpose. XML is a language for representing data as a string of text that includes interspersed “markup” for describing properties of data. It is a subset of the early Standard Generalized Markup Language (SGML). Its original goal was to enable generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML. XML has been designed for ease of implementation and for interoperability with both SGML and HTML. The first version XML 1.0 became a W3C (World Wide Web Consortium) Recommendation in 1998.

XML describes a class of data objects called XML documents and partially

describes the behavior of computer programs that process them. An XML document is made up of storage units called entities, which contain either parsed or unparsed data, and some of which form markup. Markup encodes a description of the document's storage layout and logical structure. Using markup allows text to be interspersed with information related to its content or form. XML provides a mechanism to impose constraints on the storage layout and the logical structure.

As a vendor-neutral, platform-neutral, and language-neutral technology for data exchange, XML provides a rigorous set of rules enabling the structure inherent in data to be easily encoded and unambiguously interpreted using human-readable text documents. One advantage of XML is its ability to integrate data and documents. Using XML and an appropriate set of tags, data of any kind can be encoded, from highly structured database query results to unstructured documents. XML not only can represent data and free form text, it can do both in the same document. The XML is more expressive than the relation used in a relational database. It can contain other elements, vary the order and number of attributes, and contain multiple elements within the same element type. This makes it easier to represent more complex data (Graves, 2002).

The geotechnical community can benefit significantly from implementing the XML technology for geotechnical data management. First of all, as a geotechnical data transmission format, the electronic data exchange can be significantly improved. The characteristic of being platform-neutral renders the geotechnical data in an XML format easily exchangeable between different applications on different platforms. The exhaustive industry support means that tools are easily available for storing and extracting data from an XML format. Different users or departments can retrieve only the data that is relevant to their requirement from a single XML format. The possibility of error due to duplicability of multiple formats can be significantly reduced. The XML format is self-explanatory and readable by humans and computer programs. Geotechnical data in an XML format can be checked without the help of a computer. Considering the wide and differing requirements for

geotechnical data from different disciplines and different departments, it is very difficult to build a unique dictionary to satisfy all the interested parties. However, with the powerful XML transformation technology, the geotechnical data can be easily transformed back and forth following the specified vocabularies. The geotechnical data and information hence can be shared. Meanwhile, geotechnical data in an XML format can have various storage options in a database system. This provides further flexibility for managing geotechnical data.

To facilitate the sharing of information among diverse users, a formal way to describe the format is required. The XML recommendation specifies a standardized way of specifying validation rules for a document: DTD (Document Type Definition). A DTD is a text document that formally defines the lexicon of legal name for the tags in a particular XML vocabulary, as well as the meaningful ways that tags are allowed to be nested. It provides a layer of additional power on top of the document structures built into XML. A DTD includes the following essential elements:

- Definitions of tags that are used to mark up an application's documents
- A description of the allowed structural relationships between tags
- Lists of the properties allowed or required to be applied to the tags
- Everything else there is to say about the markup language's grammar

One of the strengths of XML is the sharing of such vocabulary. DTDs can be shared across network to define how a whole set of XML documents should be structured for a particular domain, all of which use the same basic syntax, parsers, and other tools. Shared XML vocabularies enable more easily searchable documents, and provide a way to exchange information between many different organizations and computer applications. The DTD defines the lexicon of tags using a syntax described in the DTD specification, which is an integral part of the XML 1.0

specification. An XML document can then be associated with a particular DTD to enable supporting programs to validate the document's contents against that document type definition; in other words, to check that the document's syntax conforms to the syntax allowed by the associated DTD (Steve, 2000). The disadvantage of DTDs is that their syntaxes are difficult to write and understand, and the lack of structure makes it obscure and hard to use. The promising next generation for DTD is the XML schema, which uses XML syntax and adds supports for data type, a content model and the XML Document Object Model. However, considering the wide availability of DTD-aware parsers and transformation sheets, DTD is chosen in this study. Some typical syntaxes for DTD used in this study are illustrated in Table 4-1. The XML attribute types and defaults used in this study are listed in Table 4-2 and Table 4-3.

GeotechML

In the geotechnical field, the development of a GeotechML was undertaken by W3G (The World Wide Web of Geotechnical Engineers). By the definition given by W3G (<http://www.ejge.com/GML/index.htm#status>), GeotechML is the geotechnical engineering application of the new Web language, XML, but it can be thought of as the new generation of HTML especially designed for the transmission of data over the Web. Basically, GeotechML introduces geotechnics-specific tags so that geotechnical data can be easily represented using simple files. The GeotechML project was initiated at Oklahoma State University and was aimed at establishing standards for the storage and interchange of geotechnical and geological data as well as geo-environmental data using XML (McPhail, 2001).

The specific vocabulary is defined by the proposed DTD. Under the root node of the document, the proposed DTD includes core tags for the first three levels of data tree representing the geotechnical data structure. The first level of geotechnical data structure contains the office, laboratory and field categories. The second and third levels involve the tasks that are normally performed at each site. The data structure

Table 4-1 Description of XML syntaxes

<i>Name</i>	<i>Syntax</i>	<i>Note</i>
XML declaration	<?xml version='1.0'?>	It tells the processor that it is dealing with an XML documents and it may contain version declaration, encoding declaration.
Document type declaration	<!DOCTYPE rootElements [...declarations...] >	It can contain declarations, references to external files containing declarations, or both.
Comments	<!--comments -- >	It allows user to insert information intended only for human to read.
Element type declaration	<!ELEMENT elementName contentModel>	It identifies an element and its potential content, whether the content is text, other elements, both, or neither.
ANY content model	<!ELEMENT elementName ANY>	An element assigned the ANY content is ready to hold any kind of information.
Attribute list declaration	<!ATTLIST elementName attributeName attributeType defaultDeclaration [...more attributes..] >	It can contain multiple attributes for the same element. The attribute type identifies the kind of content for the attribute, and the attribute defaults define the possible default situations.

Table 4-2 Typical XML attribute types

<i>Default</i>	<i>Meaning</i>
#REQUIRED	Element instances must explicitly provide a value for this attribute each time it is used.
#IMPLIED	Element instances can provide a value for this attribute, if the document author desired.

Table 4-3 Typical XML attribute defaults

Identifier	Meaning
CDATA	The attribute value can be any string of legal XML characters.
ID	The value must begin with a letter and can otherwise be composed of letters, digits, hyphens, underscores, and full stop characters. The value must also be unique within the document among all attributes of type ID.
IDREF	The attribute value must match the value of an ID attribute for an element in the same XML document.

can be extended by adding and customizing new tags based on needs. Extending and customizing the tags can be conducted at the fourth level by using a Tag Definition File (TDF). The latest Proposed Recommendation for Core Tag is available at following web address: <http://www.ejge.com/W3G/RFCGs.htm>. For geotechnical ground investigation data, the detailed GeotechML DTD can be accessed through the web address: <http://www.dur.ac.uk/-des0www4/geotechml/>. This DTD is also appended in Appendix A for reference.

GeosciML

The GeoSciML was initiated in year 2003 with “the short-term goal of representing geoscience information associated with geologic maps and observations, as well as being extensible in the long-term to other geoscience data” (Simons et al, 2006). The following are some of the major classes of GeoSciML:

- Abstract Feature: Class representing real-world objects
- Abstract Geometry: Class describe the geometry of features
- Metadata: Data about data
- Definition: Classes representing the reference systems, controlled vocabularies and dictionaries
- GeologicPortrayal: Model elements representing the selection and symbolization of mapped feature and AbstractGeometry instances, along with base map depiction and extents to draw a geologic map visualisation.

- **GeologicObject** :geological concepts that can be described in their own right and may be properties of other geological concepts, but are not mappable features.

The GeosSciML mainly focus on map data exchange at current stage. As Simons et al. (2006) pointed out that the current gap of GeoSciML is the lack of an agreed set of vocabularies for the data content, limiting the ability to deliver standard data content.

AGSML

In response of demands of the XML development for geotechnical data, AGS also set up a committee in 2003 for reviewing the potential of adopting XML for electronic transfer of AGS data. The following conclusions are obtained by the committee (AGS, 2005):

- By adopting an XML file format the AGS would be able to embrace, and be compatible with, other XML based formats, such as GML. It could also be defined as a sub set of a larger Geotechnical data format that may include sub structure information.
- By using existing AGS field and group names and the existing data hierarchy the format would be fully backwardly compatible.
- By embracing XML the AGS would be ideally positioned to work on international data format project that could benefit AGS members either directly or indirectly.

The first version of proposed AGSML has been released in August, 2005. Based on the review, the committee has been working with external parties in the preparation of an international format.

DIGGSML

The Data Interchange for Geotechnical and Geoenvironmental Specialist (DIGGS) is an international effort for geotechnical data exchange format. The development targets on combining existing Consortium of Organization for Strong Motion Observation System (COSMOS), and University of Florida data formats to develop an international standard for geotechnical data sharing. The initial data dictionary has been based on the collection of dictionaries from these three groups consisting of Laboratory testing, Insitu-testing, Piling and basic geophysical data.

XML has been extensively recognized as a tool for transfer geotechnical data. With the extensive support from the database industry, it also can be used as a storage unit for archive geotechnical data. In the following context, two new XML schemas are proposed for the proposed database model.

4.3.4 XML for Stratigraphy Data

The subsurface stratigraphy model defines the relationships between soil layers, by which the spatial inference can be conducted. The relationships between soil layers can be represented by a hierarchical family view. An example of some typical scenarios defined in this study is illustrated in Figure 4-2. The soil layer relationships are defined through this example as follow:

- **Sibling relationship**

As shown in Figure 4-2(a), the soil layers (layer A & B) are exclusive but share the same generation. The soil layers are either directly contained by the same soil layer, or directly under the stratigraphy root. The relationship between the two layers is similar as a sibling and hence named as sibling relationship.

- **Parent-child relationship**

As shown in Figure 4-2(b), where one soil layer (layer A) directly contains

the sub soil layers (layer B). In this relationship, the top surface of the parent layer is the topmost surface and the bottom surface of the parent layer is the lowest surface comparing to the surfaces of the child soil layer. Layer B is a lens layer contained by layer A.

Similar to a hierarchical family, the stratigraphy data may have several generations of soil layers and the number of generations is project dependent. The relationships between soil surfaces are defined in the stratigraphy data and these relationships should be evaluated as an integral unit of stratigraphy data. There is no logical meaning to a further break down of the information. With the XML technology available, it is now possible to obtain a more efficient way to store and transfer the stratigraphy information.

The relationships between soil layers are embedded in the XML document, and the DTD for the stratigraphy model is presented in Figure 4-3. The root element of the proposed DTD is stratigraphy. It has soil layer elements as child elements. The content model of the stratigraphy element is set to constrain the soil layer element to appear one or more times. The soil layer element is the other element in this DTD. The content model for the soil layer element is assigned as “ANY”. It can then contain any combination of elements (layer elements specifically in this study) or text. This makes it very flexible to cover a wide variety of stratigraphy situations. The soil layer element has the following four attributes:

- Name
- Topsurface
- Bottomsurface
- Description

Figure 4-2 The relationships between soil layers

The Name attribute is for uniquely identifying the soil layer in the project, its attribute type is set to “ID”, and attribute default is assigned to “REQUIRED”. The attributes of Topsurface and Bottomsurface give the *URI* (Uniform Resource Identifier) of surface file in the system, by which the GIS can locate the surface file and conduct the spatial query. The Description attribute is used for soil layer description. The attribute defaults for the last three attributes are set to “IMPLIED”. Figure 4-4 gives an example of XML document for stratigraphy information. In this example, the stratigraphy includes six soil layers: *Fill*, *OrganicClay*, *MarineClay*, *F1*, *F2* and *OASand*. The six soil layers are in a sibling relationship as described earlier.

The Description attribute presents the soil description of the soil layer. For instance, the soil description of the “*Fill*” layer is “*Well graded, well compacted, low compressibility material*”. Through the Topsurface and Bottomsurface attributes, the surface file for the top surface and bottom surface can be obtained. For soil layer of “*OASand*”, there is only a Topsurface attribute that refers to an “*oasand_t*” file available, this happens only for the bottom soil layer whenever the bottom surface can be neglected.

```
<?xml version="1.0"?>
<!-- DTD for stratigraphy-->
<!ELEMENT STRATIGRAPHY (LAYER+)>
<!ELEMENT LAYER ANY>
<!ATTLIST LAYER
  name ID #REQUIRED
  topsurface CDATA #IMPLIED
  bottomsurface CDATA #IMPLIED
  description CDATA #IMPLIED
>
```

Figure 4-3 DTD for subsurface stratigraphy


```

<?xml version="1.0"?>
<!DOCTYPE STRATIGRAPHY [
<!-- DTD for stratigraphy-->
<!ELEMENT STRATIGRAPHY (LAYER+)>
<!ELEMENT LAYER ANY>
<!ATTLIST LAYER
  name ID #REQUIRED
  topsurface CDATA #IMPLIED
  bottomsurface CDATA #IMPLIED
  description CDATA #IMPLIED
>
]>
<STRATIGRAPHY>
  <LAYER name="Fill" topsurface="fill_t" bottomsurface="fill_b"
  description="Well graded, well compacted, low compressibility material"/>
  <LAYER name="OrganicClay" topsurface="organ_t" bottomsurface="organ_b"
  description="Low density, large compressibility and low plasticity soil"/>
  <LAYER name="MarineClay" topsurface="mc_t" bottomsurface="mc_b"
  description="Low density, large compressibility and high plasticity soil"/>
  <LAYER name="F2" topsurface="f2_t" bottomsurface="f2_b" description="Sandy
Clay"/>
  <LAYER name="F1" topsurface="f1_t" bottomsurface="f1_b" description="Clayey
Sand"/>
  <LAYER name="OASand" topsurface="oasand_t" description="Weathered zone, residual
soil"/>
</STRATIGRAPHY>

```

Figure 4-4 An example of stratigraphy XML data

4.4 Characterizing Spatial Variation of Soil Parameters

4.4.1 Variogram

Basic Variogram Model

As defined in section 3.3, the variogram measures the average dissimilarity between data separated by a vector h . To obtain the variogram at any separation lag distance, a model needs to be inferred for computing a variogram value for any possible lag distance. Figure 4-5 illustrates a plot of the variogram model. Variograms are often described through three characteristics as illustrated in Figure 4-5:

- **Sill:** Sill is the plateau that variogram reaches. The sill of a variogram is approximately equal to the variance of the data. When the sill is reached, it means that the data are not spatial correlated.
- **Range:** Range is the distance at which the variogram reaches the sill.
- **Nugget effect:** The nugget effect refers to the vertical height of discontinuity at the origin. Theoretically, the average square differences are zero at the separation distance of zero. However, the variogram often does not converge to zero as the separation distance gets smaller due to short-scale variation and sample error.

Figure 4-5 Characteristics for variogram description

Some basic variogram models exist and the five most frequently used basic models are described below (Isaaks & Srivastava 1989):

- 1) Nugget effect model

$$\gamma(h) = \begin{cases} 0 & h = 0 \\ 1 & \text{otherwise} \end{cases} \quad (4-1)$$

2) Spherical model with range a

$$\gamma(h) = Sph\left(\frac{h}{a}\right) = \begin{cases} 1.5 * \left(\frac{h}{a}\right) - 0.5 * \left(\frac{h}{a}\right)^3 & \text{if } h \leq a \\ 1 & \text{otherwise} \end{cases} \quad (4-2)$$

3) Exponential model with range a

$$\gamma(h) = 1 - \exp\left(\frac{-3h}{a}\right) \quad (4-3)$$

4) Gaussian model with range a

$$\gamma(h) = 1 - \exp\left(\frac{-3h^2}{a^2}\right) \quad (4-4)$$

5) Power model

$$\gamma(h) = h^\alpha \quad 0 < \alpha < 2 \quad (4-5)$$

Where a is the range, h is the separation vector, and $\gamma(h)$ is the variogram for separation vector h . The first four model types are also called transition (bounded) models. The sill they reach is standardized to 1 (Isaaks & Srivastava 1989).

Anisotropy

Directional variability is defined as anisotropy. When the range changes with direction, while the sill remains constant, the anisotropy is said to be geometric anisotropy. When the range remains constant but the sill changes with direction, the anisotropy is called zonal isotropy. One solution to model anisotropy is to use a transformation model. Isaaks & Srivastava (1989) illustrated the two steps of using a transformation model for anisotropy. The first step is to identify the anisotropy axes, and the second step is to define a transformation that reduces all directional variograms to a common model with a standardized range of 1. Then the transformed vector h' can be expressed as:

$$h' = \bar{T} \cdot \bar{R} \cdot h \quad (4-6)$$

where h is the vector in the original x-y-z coordinate system; $\bar{R}$ is the transformation matrix from an x-y-z coordinate system to the new x'-y'-z' anisotropy axes system defined as:

$$\bar{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4-7)$$

where α is the rotation angle about the x' axis, forming the new axes x', y', it is typically called *azimuth*; β is the rotation angle about the y' axis and called *dip*; and θ is the rotation angle about z' to the final x'-y'-z' coordinate system. And the matrix $\bar{T}$ is given by:

$$\bar{T} = \begin{bmatrix} \frac{1}{a_x} & 0 & 0 \\ 0 & \frac{1}{a_y} & 0 \\ 0 & 0 & \frac{1}{a_z} \end{bmatrix} \quad (4-8)$$

where a_x , a_y , a_z are the ranges of the directional models along the anisotropy axes (Isaaks & Srivastava 1989).

The variogram modelling can be a complex process. As Goovaerts (1997) pointed out, the “goodness” character of a variogram is elusive and cannot be measured by rigorous tests; there is no best variogram model. The user should decide on a model according to the user’s experience, information available and objective of the study. A subjective decision that is clearly documented is preferable to a blind decision based on elusive tests. The variogram modeling result can be expressed as the

combination of the basic models.

4.4.2 XML for Variogram Data

The internal structure of variogram requires a flexible solution for storage. XML turns out to be a suitable option due to following considerations:

- **Share data among applications**

Using geostatistical estimation to interpret soil properties offers another option to the geotechnical engineer's toolbox. There are various geostatistical analysis packages available for geotechnical soil properties estimations. Even for the same variogram, the specifications of variogram input format can be significantly different for each application. So the documented variogram needs to be easily transformed to fulfil the specifications required by each particular application package. XML is a non-proprietary format which is human readable and machine understandable. The use of XML for storing variogram data separates the content of variogram from formatting. With the help of well-supported transforming tools, the XML variogram data can be formatted following the input requirements of the applications. These make XML a good option for variogram documentation.

- **Efficient storage and management**

Depending on the analysed dataset, the variogram description can vary from very simple basic models to highly complex nested models. For instance, a nested variogram may consist of several basic models, each adopting different anisotropy axes. It is very difficult to store and manage such a scenario using traditional relational databases, although the complexity can be partially solved by using the object-relational or object-oriented database. To build the covariance matrix for kriging applications, the applications would need to obtain the variogram information such as the number of the nested variograms, the type of each variogram and the corresponding parameters. For a variogram stored in a database, no matter whether it is a

relational database or object-oriented database, the query for such information needs a set of internal manipulations which are database dependent. With XML, the solution would be more straightforward and efficient. Instead of storing a variogram in linked tables or creating a new object type, the XML document for a variogram can be stored intactly in a database table. The database does not need to provide special provision for the internal structure of the variogram and it can treat the XML document in the same way as in other data types. The XML document for variogram then can be retrieved using standard SQL, searched using Xpath and transformed using XSLT tools. All these tools are available and can be easily tailored for specific application regardless of the database at the backend. Meanwhile, more and more XML processing tools have emerged to allow internal processing inside the database by the vendors.

A practical variogram for spatial description can be treated as a composition of basic variograms. Based on the considerations mentioned above, an XML DTD for variogram documentation is proposed in this study and presented in Figure 4-6.

```
<? Xml version="1.0"?>
<!-- DTD for variogram -->
<!ELEMENT Nugget ANY>
<!ELEMENT Variograms (Variogram*)>
<!ATTLIST Number CDATA #REQUIRED>
<!ELEMENT Variogram (
    Type
    | Coefficient
    | (Angle1, Angle2, Angle3)
    | (Max_horizontal_range, Min_horizontal_range, Vertical_range)
)>
<!ELEMENT Type ANY>
<!ELEMENT Coefficient ANY>
<!ELEMENT Angle1 ANY>
<!ELEMENT Angle2 ANY>
<!ELEMENT Angle3 ANY>
<!ELEMENT Max_horizonyal_range ANY>
<!ELEMENT Min_horizonyal_range ANY>
<!ELEMENT Vertical_range ANY>
```

Figure 4-6 DTD for variogram documentation

The root element of the DTD is “variograms”, which contains at least one variogram element. If there is a nugget effect then the nugget element should be used. The following elements are defined in the DTD to describe a basic variogram:

- Type: it defines the type of the basic variogram.
- Coefficient: it is the weight of current variogram in the whole variogram.
- Angle1, Angle2, Angle3: these angles describe the geometric anisotropy angles.
- Max_horizontal_range, Min_horizontal_range, Vertical_range: these elements define the ranges of the current variogram.

4.5 Summary

The geotechnical data required for 3D GIS applications are identified in this Chapter: The data are organised into three categories as borehole test data, stratigraphy data and variogram data.

The borehole test data are well described in previous studies. Considering the variety and level of details of geotechnical data, the AGS format was adopted as a template for this study, to take advantage of the well established development of this format. However, the adoption of the AGS format does not exclude the potential of specific modification whenever the need arises.

The process of extracting the stratigraphy information from soil investigation data is a vital process in geotechnical characterization since stratigraphy data describes the internal logic for subsurface soil layers. The relationships between soil layers require an efficient and flexible means of storing and presenting stratigraphy data.

Spatial variability is an inherent feature of all geotechnical parameters. The variogram is a spatial description measuring the average dissimilarity between data. There exist some predefined basic variogram models and a variogram can be

presented as a combination of basic variogram models.

With a set of rules enabling the structure inherent in the data to be easily encoded, XML provides an effective solution for the storage of geotechnical data, especially stratigraphy data and variogram data. The DTDs for stratigraphy data and variogram data have been proposed in this chapter.

Chapter 5 Hybrid Database Model

The geotechnical database model conceived in this study requires the development of a data model for capturing the borehole test data, stratigraphy data and variogram data. This chapter discusses the logical relationship between these geotechnical data. This chapter is organized in three parts. It begins with an introduction of the NADM model (North American Geologic-map Data Model) as the worldview of the NADM model is adopted by the proposed database model. Then a detailed description of the proposed database model is given. Three components of the proposed model are illustrated: Fact, Concept, and Meta-data. At the final part of this chapter, a prototype is presented to illustrate more clearly the applicability of this hybrid database model.

5.1 Worldview of Geosciences and NADM Model

Database model in this study refers to the logical database model rather than the physical database model. A database model is a formal abstraction of reality. Hence the conceptualisation of space is a fundamental problem which affects and accompanies any development aimed at modelling the geoscience world. The worldview, as a set of assumptions and commitments embedded in a database model, must be developed. Raper (2000) illustrated the following fundamental issues that must be covered in a worldview:

- The theoretical grounding and conceptualisation of the “world” employed (metaphysics);
- The methodology by which the contents of the conceptualised world are defined, ordered and signified (ontology);
- The procedures by which knowledge of the conceptualised world is established and evaluated (epistemology);

- The nature of human knowledge of the conceptualised world (philosophy of mind);
- The nature of language and its role in communication and the construction of meaning (linguistics);
- Processes of cognition, the nature of intelligence and the functioning of the mind (cognitive science); and
- The nature of computation employing symbolic and informational representations of human knowledge (informatics).

Li & Hayashi (1999) described two conflicting hypotheses about the world. The first one views the world as “the spatial-temporal clusters of known attributes are things”. The second views the world as “there exist things in time and space which have (known and unknown) attributes”. Raper (2000) proposed that geoscience can be set consistently within a worldview that the world is represented from diverse human perspectives. As Brodaric & Gahegan (2000) pointed out, scientific features are naturally more dynamic, numerous and subjective. Scientific communities, such as geoscience disciplines which are reliant on observational data, are particularly more likely to share methodology than ontology, as observation methods are shared, but facts are partially subjective and personal. And the geoscientist’s task is to build the models of the possible world as what it might be. The NADM model (Johnson et al., 1999) and its variants are among the representatives of the data models based on these viewpoints.

The purpose of the NADM model is to provide structure for the organization, storage, and use of geologic map data in a computer (Johnson et al., 1999). It is designed to be a catalogue of separate, digitised geologic maps. The simplified relational NADM model is presented in Figure 5-1 (Brodaric & Gahegan, 2000).

Figure 5-1 North America Data Model simplified (after Brodaric & Gahegan, 2000)

Rather than treating features as clearly defined by assuming the unquestioned world, the features are delineated into two classes: facts and concepts. The facts are the observations and description attributes. The concepts are the interpretations and classifications used to organize the spatial objects and data into a coherent model of the “possible world”. Based on a set of existing facts a concept is generated, so a concept can be treated as a compound object consisting of facts. Decoupling features into facts and concepts reflects the evolution of the exploration life cycle. By emphasizing on the interpretative aspects of defining a feature, it “enables the linking, from one or more sources of information, of concepts, data and associated spatial objects through a legend or classification” (CGDMWG, 2000). Hastings & Brodaric (2001) developed the model into an object-oriented meta-model in the National Geologic Map Database project. The data model comprises four meta-objects as shown in a diamond diagram as shown in Figure 5-2:

Figure 5-2 Diamond diagram of National Geologic Map Database Project (after Hastings & Brodaric, 2001)

- Symbol: the cartographic notation used to depict objects.
- Concept: the geologic formalism selected for representation.
- Occurrence: the geologic reality recognized in the field and/or laboratory.
- Descriptor: the digital value chosen to characterize map Symbol, Concept and Occurrence.

In this model, elaboration of concepts leads into geologic ontology (what we know) and elaboration of occurrences leads into geologic epistemology (how we know). Although the NADM and its variants are emphasized on digital geologic maps, cartography and analysis, its philosophy applies equally well to the geotechnical field.

5.2 Database Model Descriptions

Database model provides a link between user needs and the software solutions that meet them. On the one hand, human perception of reality is dynamic, tentative and

vague. For instance, geologists agree that chert, granite, limestone, sandstone, and tuff are all distinct types of rocks, but even the first level classification of these types is arguable sometimes. Classification options have changed and continue to change over time. On the other hand, computers are machines that are precise and rigid. Hence a worthwhile database model must strike a balance between the user needs and software solutions (Hasting & Brodaric 2000).

In this study a hybrid geotechnical database model is proposed for consolidating and integrating geotechnical data into a custodian database, where geotechnical data are consolidated independent of any application. It will function as a data warehouse domain as described by Stirling (2002). Figure 5-3 illustrates the data warehouse domain together with another basic data management domain: data application domain. In the application domain the database and data management functions are an integral part of the application, while in the warehouse domain the database and data management functions are independent of any application. This hybrid model is developed based on the NADM model proposed by Johnson et al. (1999).

To create a stratigraphy-centred database model, the subjective factor involved in the database model must be considered, a high level abstraction must be built on top of the existing geotechnical database model. A new database model is proposed in this chapter. Inherited from the NADM model, Figure 5-4 shows the proposed data model. This model is composed of three archives:

- Fact
- Concept
- Meta-data.

The laboratory and in-situ tests are abstracted as different classes and are stored in the *Fact* archive. They are the atomic classes in this data model. The *Concept* archive consists of classes used for classification of those *fact* classes. The *Meta-*

data archive includes information about project, borehole, tests, owner, and user of borehole data. Only selected classes are presented in Figure 5-4 and the following archive illustration. These classes are chosen to be representative based on their usage frequency in routine geotechnical practice. However other tests can be easily added following the same procedure. One easy way to define class attributes of the laboratory tests and in-situ tests is to map these attributes from the established data structure. As illustrated in Chapter 4, the AGS format provides a rigid and well-recognized geotechnical data dictionary through over two decades of development and evolution, hence the attributes of the in-situ tests and laboratory tests are defined in this study by mapping these attributes from the AGS format.

Figure 5-3 The data application domain versus the data warehouse domain (after Stirling, 2002)

Figure 5-4 Hybrid data model for geotechnical database

5.2.1 Fact

As shown in Figure 5-4, the *Fact* archive contains the borehole test data. Figure 5-4 only shows three in-situ tests and three laboratory tests for illustration purpose. Figure 5-5 gives the typical laboratory and in-situ test classes defined in the *Fact* archive. The laboratory tests presented in Figure 5-5(a) are soil classification test, particle size distribution test, consolidation test, shearbox test, triaxial test, compaction test, laboratory permeability test and relative density test. The in-situ tests presented in Figure 5-5(b) include SPT (Standard Penetration Test), CPT (Cone Penetration Test), field vane, in-situ permeability, resistivity and density test. All laboratory tests are inherited from the *sample* class, which is also included in Figure 5-5(a). The detailed explanation of each attribute for the typical *Fact* classes is given in Table 5-1. The contents of most existing database models can be categorized in this archive.

5.2.2 Concept

The *Concept* component is what distinguishes this proposed model from other geotechnical database models. It is an archive for geotechnical interpolation and judgment. The information stored in this archive could be interactively created by the user. Figure 5-6 shows the six classes used for soil layer definition in the concept archive, which is the core part for the proposed database model in this study. Their attributes are presented in Table 5-2. The proposed classes include:

- ***Stratum* class**

The *stratum* class is the atomic unit in a borehole. It has attributes to record the top and bottom elevations and descriptions of each stratum. Stratum data is generated either during the borehole drilling process or from interpolation of sounding tests such as SPT or CPT. A strategy of some applications for stratum sorting is to define a stratum attribute (a sequence number or code) for reference use. This generates a situation where stratum data are tightly

Figure 5-5(a) Class attributes of typical Fact classes for laboratory tests

Fact																				
<table><tr><th>SPT</th></tr><tr><td>Depth: number Seatingdriveblow: int Maindrive: int Totalpenetration: number N: number Casingdep: number Waterdep: number Testtype: varchar Remarks: varchar Seatingblow1: int Seatingblow2: int Testblow1: int Testblow2: int Testblow3: int Testblow4: int Seatingpen1: number Seatingpen2: number Testpen1: number Testpen2: number Testpen3: number Testpen4: number</td></tr><tr><td></td></tr></table>	SPT	Depth: number Seatingdriveblow: int Maindrive: int Totalpenetration: number N: number Casingdep: number Waterdep: number Testtype: varchar Remarks: varchar Seatingblow1: int Seatingblow2: int Testblow1: int Testblow2: int Testblow3: int Testblow4: int Seatingpen1: number Seatingpen2: number Testpen1: number Testpen2: number Testpen3: number Testpen4: number		<table><tr><th>CPT</th></tr><tr><td>Depth: number Testtype: varchar Reference: varchar Resistance: number Frictionresistance: number Porepressure1: number Porepressure2: number Porepressure3: number Conductivity: number Temperature: number</td></tr><tr><td></td></tr></table> <table><tr><th>PIEZOMETER READING</th></tr><tr><td>Depth: number Date: date Time: time Waterdept: number Waterhead: number Remarks: varchar</td></tr><tr><td></td></tr></table> <table><tr><th>FIELDVANE</th></tr><tr><td>Depth: number Testnum: int Remarks: varchar Testresult: number Remouldedresult: number</td></tr><tr><td></td></tr></table>	CPT	Depth: number Testtype: varchar Reference: varchar Resistance: number Frictionresistance: number Porepressure1: number Porepressure2: number Porepressure3: number Conductivity: number Temperature: number		PIEZOMETER READING	Depth: number Date: date Time: time Waterdept: number Waterhead: number Remarks: varchar		FIELDVANE	Depth: number Testnum: int Remarks: varchar Testresult: number Remouldedresult: number		<table><tr><th>PERMEABILITY INSITU</th></tr><tr><td>Top: number Base: number Stagenumber: int Testtype: varchar Waterdepbefore: number Waterdepstart: number Testzonediameter: number Pipediameter: number Permeability: number Remarks: varchar Flow: number AWL: number HEAD: number</td></tr><tr><td></td></tr></table> <table><tr><th>HOLEINFO</th></tr><tr><td>Holeid: varchar Exploretype: varchar Eastofstart: number Northofstart: number Groundlevel: number Finaldepth: number Startdate: date Loggingperson: varchar Remarks: varchar Eastofend: number Northofend: number GLend: number Reference: varchar Xlocal: number Ylocal: number Zlocal: number Enddate: date Backfilldate: date Driller varchar Orientation: number Locationmethod: varchar</td></tr><tr><td></td></tr></table>	PERMEABILITY INSITU	Top: number Base: number Stagenumber: int Testtype: varchar Waterdepbefore: number Waterdepstart: number Testzonediameter: number Pipediameter: number Permeability: number Remarks: varchar Flow: number AWL: number HEAD: number		HOLEINFO	Holeid: varchar Exploretype: varchar Eastofstart: number Northofstart: number Groundlevel: number Finaldepth: number Startdate: date Loggingperson: varchar Remarks: varchar Eastofend: number Northofend: number GLend: number Reference: varchar Xlocal: number Ylocal: number Zlocal: number Enddate: date Backfilldate: date Driller varchar Orientation: number Locationmethod: varchar	
SPT																				
Depth: number Seatingdriveblow: int Maindrive: int Totalpenetration: number N: number Casingdep: number Waterdep: number Testtype: varchar Remarks: varchar Seatingblow1: int Seatingblow2: int Testblow1: int Testblow2: int Testblow3: int Testblow4: int Seatingpen1: number Seatingpen2: number Testpen1: number Testpen2: number Testpen3: number Testpen4: number																				
CPT																				
Depth: number Testtype: varchar Reference: varchar Resistance: number Frictionresistance: number Porepressure1: number Porepressure2: number Porepressure3: number Conductivity: number Temperature: number																				
PIEZOMETER READING																				
Depth: number Date: date Time: time Waterdept: number Waterhead: number Remarks: varchar																				
FIELDVANE																				
Depth: number Testnum: int Remarks: varchar Testresult: number Remouldedresult: number																				
PERMEABILITY INSITU																				
Top: number Base: number Stagenumber: int Testtype: varchar Waterdepbefore: number Waterdepstart: number Testzonediameter: number Pipediameter: number Permeability: number Remarks: varchar Flow: number AWL: number HEAD: number																				
HOLEINFO																				
Holeid: varchar Exploretype: varchar Eastofstart: number Northofstart: number Groundlevel: number Finaldepth: number Startdate: date Loggingperson: varchar Remarks: varchar Eastofend: number Northofend: number GLend: number Reference: varchar Xlocal: number Ylocal: number Zlocal: number Enddate: date Backfilldate: date Driller varchar Orientation: number Locationmethod: varchar																				

Figure 5-5(b) Class attributes of typical Fact classes for in-situ tests

Table 5-1 Class attributes of *Fact* classes (Extracted from AGS format)

Class name: CONSOLIDATION			
Attribute name	Unit	Description	Example
Consolidationinfo		Consolidation test information	
Stressincrementnum		Oedometer stress increment number	3
Voidratioinitial		Initial void ratio	0.80
Endstress	kN/m ²	Stress at end of stress increment	400
Voidratiofinal		Final void ratio	0.62
INMV	m ² /MN	Coefficient of volume compressibility over stress increment	0.32
consolidationcoeff	m ² /yr	Coefficient of consolidation	4.12
INSC		Coefficient of secondary compression over stress increment	0.12
Class name: COMPACTION			
Attribute name	Unit	Description	Example
Compactioninfo		General compaction information	
Pointnum		Compaction point number	1
Watercontent		Moisture content	13.5
Drydensity	Mg/m ³	Dry density	1.85
Class name: SAMPLE			
Attribute name	Unit	Description	Example
Depthtotop	m	Depth to top of sample	25.3
Samplerefernum		Sample reference number	2
Sampletype		Sample type	U
Diameter	mm	Sample diameter	100
Description		Sample description	Stiff brown clay
Sampleblownum		Blows to driven sampler	29
Remarks		Notes	
Class name: PERMEABILITY LABORATORY			
Attribute name	Unit	Description	Example
Sample		General sample information	
Testnumber		Test number	2
Testmethod		Test method	Constant head
Condition		Sample condition	undisturbed
Cutsize	mm	Size cut off of material too coarse for testing	5
UNS	%	Proportion of material too coarse for testing	36
Diameter	mm	Sample diameter	102
Length	mm	Sample length	200
MC	%	Moisture content	20.3
BDEN	Mg/m ³	Initial bulk density	2.24
DDEN	Mg/m ³	Dry density	1.84
Voidratio		Void ratio	0.37
Coefficient	m/s	Coefficient of permeability	4E-6
Meanstress	kN/m ²	Mean effective stress	112
SaturationI	%	Initial degree of saturation	72
SaturationF	%	Final degree of saturation	98
Particledensity		Particle density	2.65

Table 5-1 Class attributes of *Fact* classes (Extracted from AGS format)
(Continued)

Class name: TRIAXIAL			
Attribute name	Unit	Description	Example
Triaxialinfo		General triaxial test information	
Testnumber		Traxial test number	2
Diameter	mm	Specimen diameter	38
MCinitial	%	Initial moisture content	15
Cellpressure	kN/m ²	Total cell pressure	100
Deviatorstress	kN/m ²	Deviator stress at failure	360
Length	mm	Sample length	76
Bulkdensity	Mg/m ³	Initial bulk density	2.12
Drydensity	Mg/m ³	Initial dry density	1.84
PWPinitial	kN/m ²	Pore water pressure at start of shear stage	50
PWPfinal	kN/m ²	Pore water pressure at failure	60
Failurestrain	%	Failure strain	9
Failuremode		Failure mode	Brittle, plastic
Class name: RELATIVEDENSITY			
Attribute name	Unit	Description	Example
Sample		General sample information	
Testmethod		Test method	
Maxdrtdensity	Mg/m ³	Maximum dry density	2.15
Sieve375	%	Weight percentage retained by 37.5 mm sieve	8.0
Sieve63	%	Weight percentage retained by 63 mm sieve	13
Sieve2	%	Weight percentage retained by 2 mm sieve	5.8
Mindrydensity	Mg/m ³	Minimum dry density	1.66
Class name: SHEARBOX			
Attribute name	Unit	Description	Example
Shearboxinfo		General shearbox test information	
Stagenumber		Shear box stage number	1
Bulkdensity	Mg/m ³	Bulk density	1.63
Drydensity	Mg/m ³	Dry density	1.21
Normalstress	kN/m ²	Shear box normal stress	63.5
Displacerate	mm/s	Displacement rate	
Peakstress	kN/m ²	Peak shear stress	65.5
Residuestress	kN/m ²	Residue shear stress	47.2
Peakdis	mm	Displacement at peak shear stress	2.35
Residuedis	mm	Displacement at residue shear stress	12.3
Particledensity		Particle density	2.65
Voidratio		Initial void ratio	0.5
MCinitial	%	Initial water content	20
MCfinal	%	Final water content	18.3
Class name: PARTICLEDISTRIBUTION			
Attribute name	Unit	Description	Example
Sample		General sample information	
Sievesize	mm	Sieve or particle size	3.35
percentage	%	Percentage passing	13.5
testtype		Test type	WS

Table 5-1 Class attributes of *Fact* classes (Extracted from AGS format)
(Continued)

Class name: CLASSIFICATION			
Attribute name	Unit	Description	Example
Depthtotop	m	Depth to top of sample	6.5
Samplerefnnumber		Sample reference number	12
Sampletype		Sample type	U
Specimenrefer		Specimen reference number	2
Specimendep	m	Specimen depth	6.6
Moisturecontent	%	Moisture content	58
Liquidlimit	%	Liquid limit	62
Plasticlimit	%	Plastic limit	38
Drydensity	Mg/m ³	Dry density	1.06
Sieve425	%	Percentage passing 425 sieve	12
Bulkdensity	Mg/m ³	Bulk density	1.66
Particaldensity			2.65
Class name: DENSITYTEST			
Attribute name	Unit	Description	Example
Depth	m	Depth to top of test	13.5
Remarks		Notes	
Bulkdensity	Mg/m ³	Bulk density	1.86
MC	%	Moisture content	18
Class name: PIEZOMETERREADING			
Attribute name	Unit	Description	Example
Depth	m	Depth to top of test	13.5
Date	dd/mm/yy	Date of reading	26/3/2001
Time	hhmmss	Time of reading	164000
Waterdept	m	Depth to water below ground surface	6.40
Waterhead	m	Head of water above piezometer tip	0.85
Remarks		Notes	
Class name: RESISTIVITYINSITU			
Attribute name	Unit	Description	Example
Top	m	Depth to top	13.5
Base	m	Depth to base	23.5
Type		Test type	
Result	ohmcm	Result	2000
Remarks		Notes	
Class name: CPT			
Attribute name	Unit	Description	Example
Depth	m	Depth to top of test	13.5
Testtype		Cone test type	PC
Reference		Identification reference	PQ47
Resistance	MN/m ²	Cone resistance	20
Frictionresistance	kN/m ²	Local unit side friction resistance	1000
Porepressure1	kN/m ²	Porewater pressure	15.0
Porepressure2	kN/m ²	2 nd porewater pressure	15.0
Porepressure3	kN/m ²	3 rd porewater pressure	15.0
Conductivity	uS/cm	Conductivity	0.01
Temperature	DegreeC	Temperature	10

Table 5-1 Class attributes of *Fact* classes (Extracted from AGS)
(Continued)

Class name: SPT			
<i>Attribute name</i>	<i>Unit</i>	<i>Description</i>	<i>Example</i>
Depth	m	Depth to top of test	13.5
Seatingdriveblow		Blow number for seating drive	14
Maindrive		Blow number for main test drive	35
Totalpenetration	mm	Total penetration	450
N		SPT "N" value	35
Casingdep	m	Casing depth at time of test	12.00
Waterdep	m	Depth to water	2.50
Testtype		Type of SPT test	S
Remarks		Notes	
Seatingblow1		Blow number for 1 st increment (seating)	6
Seatingblow2		Blow number for 2 nd increment (seating)	8
Testblow1		Blow number for 1 st increment (test)	8
Testblow2		Blow number for 2 nd increment (test)	9
Testblow3		Blow number for 3 rd increment (test)	9
Testblow4		Blow number for 4 th increment (test)	9
Seatingpen1	mm	Penetration for 1 st increment (seating)	75
Seatingpen2	mm	Penetration for 2 nd increment (seating)	75
Testpen1	mm	Penetration for 1 st increment (test)	75
Testpen2	mm	Penetration for 2 nd increment (test)	75
Testpen3	mm	Penetration for 3 rd increment (test)	75
Testpen4	mm	Penetration for 4 th increment (test)	75
Class name: FIELDVANE			
<i>Attribute name</i>	<i>Unit</i>	<i>Description</i>	<i>Example</i>
Depth	m	Depth to top of test	13.5
Testnum		Vane test number	1
Remarks		Notes	
Testresult	kN/m ²	Vane test result	60
Remouldresult	kN/m ²	Vane test remoulded result	45
Class name: PERMEABILITY INSITU			
<i>Attribute name</i>	<i>Unit</i>	<i>Description</i>	<i>Example</i>
Top	m	Depth to top	12.2
Base	m	Depth to base	12.95
Stagenumber		Stage number	1
Testtype		Test type	Constant head
Waterdepbefore	m	Depth to water in borehole immediately prior to test	10.60
Waterdepstart	m	Depth to water at start of test	5.40
Testzonediamater	m	Diameter of test zone	0.15
Pipediameter	m	Diameter of stand pipe or casing	0.019
Permeability	m/s	Permeability	5E-9
Remarks		Notes	
Flow	l/s	Average flow during packer test stage	2.3
AWL	m	Depth to assumed standing water level	10.0
HEAD	m	Applied total head of water during test stage at center of packer test zone	20.5

Table 5-1 Class attributes of *Fact* classes (Extracted from AGS)
(Continued)

Class name: HOLEINFO			
Attribute name	Unit	Description	Example
Holeid		Exploratory hole name/number	Hole125-4
Exploretype		Type of exploratory hole	CP
Eastofstart	m	Easting of hole or start of traverse	23541
Northofstart	m	Northing of hole or start of traverse	45672
Groundlevel	m	Ground level relative to Datum of hole	16.3
Finaldepth	m	Final depth of hole	33.60
Startdate	dd/mm/yy	Start date of excavation	23/5/2003
Loggingperson		The responsible person	DPG
Remarks		Notes	
Eastofend	m	Easting of end of traverse	25648
Northofend	m	Northing of end of traverse	65141
GLend	m	Ground level relative to end of traverse	9.67
Reference		Ordnance survey letter grid reference	TQ 123456
Xlocal	m	Local x coordinate	450
Ylocal	m	Local y coordinate	425
Zlocal	m	Local z coordinate	106.5
Enddate	dd/mm/yy	Hole end date	28/5/2003
Backfilldate	dd/mm/yy	Hoel backfill date	4/6/2003
Driller		Name of driller	
Orientation	degree	Orientation of the hole	010
Locationmethod		Method of location	GPS

Figure 5-6 Class attributes of Concept classes for soil layer category

Figure 5-7 Relationship types of stratum in a connection

Table 5-2 Class attributes of *Concept* classes

Class name: LAYER			
Attribute name	Unit	Description	Example
Source		Reference to source of definition	
Name		Layer name	Layer 1
Topsurface		Top surface	
Bottomsurface		Bottom surface	
Connections			
Status			
Description		Layer descriptions	
Class name: SURFACE			
Attribute name	Unit	Description	Example
Source		Source of definition	
Reference		Surface reference name	
Type		Connection type	Top
Connections		Collection of connection components	
Class name: CONNECTION			
Attribute name	Unit	Description	Example
Source		Source of definition	
Strata		Collection of stratum components	
Type		Relationship type	Map
Class name: STRATUM			
Attribute name	Unit	Description	Example
DepthtoTop	m	Depth to top	12.5
DepthtoBase	m	Depth to base	14.5
Description		Stratum description	Grey sandy clay
Geology		Stratum geology	Jurong formation
Holeinfo		Borehole information	
Class name: STRATIGRAPHY			
Attribute name	Unit	Description	Example
Project		Project name	
Engineer		Engineer define the stratigraphy	
Stratigraphy		XML for stratigraphy	
Class name: VARIOGRAM			
Layer		Reference to soil layer	
Datatype		Type of dataset	SPT, CPT
Variogram		XML for variogram	

bounded with a stratigraphy schema. It works in a project-specific environment but it is still hard to manage. When a sequence number is applied, whenever the stratum information is revised or updated, it will require the resorting of the strata (Oloufa et al., 1994). The alternative of using reference codes requires the keeping of a global sequence dictionary and hence causes the extensibility problems. In this model, the stratum class is made an atomic class independent of the stratigraphy schema. In general the stratum class is relatively stable because it is seldom changed once it is defined.

- **Connection class**

A stratum identified in a borehole can be linked to the strata identified from other boreholes. The connection class is a group of strata, meanwhile all the strata defined in a connection should be correlated with same relationships. Figure 5-7 shows three types of exclusive relationships of strata defined in this study for the proposed database model: *Map*, *Top* and *Bottom*. When the top elevations of a collection of strata belong to the same soil surface while the bottom elevations of this collection of strata also share another soil surface, the relationship for this collection of strata is defined as “*Map*”. If only the top elevations of a collection of strata are on the same soil surface then the relationship for this collection of strata is defined as “*Top*”. Meanwhile, if only the bottom elevations of a strata collection can share the same soil surface then the relationship for this stratum collection is defined as “*Bottom*”. The *Connection* class is composed of a collection of *stratum* classes which shares the same relationship.

- **Surface class**

The *surface* class is the elevation source for the soil surface model. It is composed of a collection of *connection* classes. A continuous soil surface can be defined using the retrieved elevation values from all the strata in this collection of connection classes. A *surface* type attribute is designed to infer the elevation values from a collection of strata sharing a “*Map*” relationship.

The optional attribute values for this attribute can either be “*Top*” or “*Bottom*”, which clearly identifies the source of elevation values when a “map” relationship is presented.

- **Layer class**

A soil layer is hereby composed of three components: a *surface* class for the top surface, a *surface* class for the bottom surface and a collection of *connection* classes which defines all the strata inside that soil layer.

- **Stratigraphy class**

The relationship between each layer is stored as *stratigraphy* XML data. The stratigraphy is linked to the project by project name and the source of the stratigraphy definition is also recorded.

- **Variogram class**

The variogram data are also categorized in the *concept* archive. The XML variogram data is bounded with soil layer and the type attribute is used to identify the test used to obtain the dataset from which the variogram is derived.

Geotechnical knowledge and judgement is needed in the definition of a *stratum* object, a *surface* object and a *layer* object. As the geotechnical judgement can be subjective and personal, a *source* class is appended to these *concept* classes for tracing the source of the judgment.

5.2.3 Meta-data

Meta-data is data about data. It is a description of the information resources. Figure 5-8 shows the typically defined *Meta-data* classes in this study. Their attributes are detailed in Table 5-3. The major classes include:

- **Project class**

Information such as the project title, client name, and contractor name are stored in the *project* class. The *project* class is associated by *holeinfo* class

using a one-to-many relationship in this proposed model.

- **Source class**

The information for the origination of the geotechnical judgment is defined in the *source* class. It is logical in some situations that the concept objects are identified through tracing the original source.

- **Compaction Info class**

This class contains the general information about laboratory compaction test. As the test is a laboratory test, it also contains sample reference for tracing the sample information.

- **Shearbox Info Class**

The general information about shearbox test is defined in this class. Similarly, it also contains a sample reference.

- **Consolidation Info class**

This class contains the general information about the consolidation test.

- **Triaxial Info class**

The general information about triaxial test is defined in this class.

- **Piezometer Installation class**

Information in this class includes the installation depth of the piezometer, installation date, piezometer type etc.

Information about in-situ tests and laboratory tests in the source archive are general descriptions about the tests conducted and can be shared by repeated observations and readings. Decoupling this information from those repeated observations and readings, where those repeated observations and readings are recorded in the fact archive and the general information are stored in the meta-data archive, will avoid excessive information storage and redundancy and manage the geotechnical data more efficiently.

Figure 5-8 Class attributes of Meta-data classes for geotechnical tests

Table 5-3 Class attributes of *Meta-data* classes (Extracted from AGS)

Class name: CONSOLIDATIONINFO			
Attribute name	Unit	Description	Example
Sample		Object reference to sample	
Depthtotop	m	Depth to top of sample	6.5
Specimenrefer		Specimen reference number	3
Specimendep	m	Specimen depth	6.9
Testtype		Oedometer or Rowe, primary or secondary consolidation	Oed, Rowe
Samplecondition		Sample condition	undisturbed
Remark		Test details	
INCM	m ² /MN	Coefficient of volume compressibility	0.36
Stressrange	kN/m ²	Defined stress range	P'_0 to $P'_0 + 100$
Diameter	mm	Test specimen diameter	75
MCinitial	%	Initial moisture content	19
MCfinal	%	Final moisture content	21
BDinitial	Mg/m ³	Initial bulk density	2.12
DDinitial	Mg/m ³	Initial dry density	1.75
ParticleDensity	Mg/m ³	Particle density	2.65
Saturationdeg	%	Initial degree of saturation	98
Swellingpressure	kN/m ²	Swelling pressure	100
Heightchange	%	Height change of specimen	1.1
Class name: COMPACTIONINFO			
Attribute name	Unit	Description	Example
Sample		Object reference to sample	
Depthtotop	m	Depth to top of sample	6.5
Specimenrefer		Specimen reference number	3
Specimendep	m	Specimen depth	6.9
Testtype		Compaction test type	vibro
Mouldtype		Mould type	Standard or CBR
Sieve375	%	Weight percent on 37.5mm sieve	7
Sieve20	%	Weight percent on 20mm sieve	15
Particledensity	Mg/m ³	Particle density	2.65
Maxdrydensity	Mg/m ³	Maximum dry density	2.06
Watercontent	%	Moisture content at maximum dry density	14
Remarks		Notes	
Class name: SHEARBOXINFO			
Attribute name	Unit	Description	Example
Sample		Object reference to sample	
Depthtotop	m	Depth to top of sample	6.5
Specimenrefer		Specimen reference number	3
Specimendep	m	Specimen depth	6.9
Testtype		Test type	Small shear box
Remark		Notes	
Peakintercept	kN/m ²	Peak cohesion intercept	5
Peakangle	deg	Peak angle of friction	26.5
Residualintercept	kN/m ²	Residual cohesion intercept	1
Residualangle	deg	Residual angle of friction	13.0

Table 5-3 Class attributes of *Meta-data* classes (Extracted from AGS)
(Continued)

Class name: TRIAXIALINFO			
Attribute name	Unit	Description	Example
Sample		Object reference to sample	
Depthtotop	m	Depth to top of sample	6.5
Specimenrefer		Specimen reference number	3
Specimendep	m	Specimen depth	6.9
Testtype		Test type	UU
Samplecondition		Sample condition	Undisturbed
Remarks		Notes	
CU	kN/m ²	Undrained shear strength	50
Cohesion	kN/m ²	Cohesion intercept	3
Frictionangle	deg	Friction angle	32
CLASS NAME: PROJECT			
Attribute name	Unit	Description	Example
Title		Project title	PG 125A
Location		Location of project	London Road
Client		Name of client	ADE group
Contractor		Name of contractor	BHF construction
Engineer		Name of engineer	BHF consulting
Remarks		Notes	
Date	dd/mm/yy	Date of production of data	15/6/2003
Class name: PIEZOMETERINSTALLATION			
Attribute name	unit	Description	Example
Date	dd/mm/yy	Date of reading	22/3/2003
Type		Piezometer type	
Top	m	Depth to top of response zone	6.50
Base	m	Depth to base of response zone	7.50
Remarks		Notes	
Class name: SOURCE			
Attribute name	unit	Description	Example
Engineer		Name of engineer	Tom
Project		Project name	PDG-123
Definetime	dd/mm/yy	Time of definition	16/6/2003
Remarks		Notes	

5.3 Prototype Illustration

To illustrate the proposed hybrid database model, a prototype is presented in this section. In this prototype evaluation, the data model is transformed into an object-relational database schema and implemented in an Oracle 8i database platform. The

database design model is presented in Figure 5-9. Only the laboratory classification test results are included in the database schema for illustration purposes; no in-situ test results are considered in this prototype. The implemented attributes for this prototype are illustrative rather than exhaustive. Figure 5-9 also shows three stored procedures developed in the prototype and stored inside the database package: *LayerManager*. The called procedures navigate through layer table to sample_class table (Figure 5-9) based on the identifier of the soil layer and return the expected results: mean value of water contents, number of samples, and number of classification tests.

A simplified scenario is presented in Figure 5-10. A total of three boreholes are used in this scenario. Four samples are obtained, two are from Hole 1 and one each from Hole 2 and Hole 3, respectively. The test data are “facts” that exist independently of the stratigraphy scheme. The stratigraphy judgments are made based on the borehole information. Figure 5-10 shows four possible stratigraphy scenarios. The multiple views of the same dataset for a unique reality reflect our interpretation of the available data, although at the final stage normally only one optimal schema can be reached by evaluating all the available information. All the soil property data and soil layer catalogues are populated into the database using Oracle specific PL/SQL. Some intrinsic logic can also be stored inside the database through stored procedures. The stored procedure can be accessed by other client applications because it is callable by standard SQL. Meanwhile the net communication between the client application and the database can be reduced significantly by embedding the data intensive tasks inside the database.

An example of a SQL statement calling the stored procedures is presented in Figure 5-11. The statement queries the database for the number of samples, the total classification test numbers and the mean value of water content of each interpreted soil layer based on the project name. Figure 5-11 also lists the results for the four possible scenarios presented in Figure 5-10. The first column of returned result

Figure 5-9 Database diagram for the prototype

Figure 5-10 Interpreted stratigraphy resulting from different judgments based on available borehole information

Figure 5-11 A SQL statement using the stored procedure

shows the name of the soil layer for the four scenarios of Figure-10. The second column shows the number of sample for each layer. The third column is the number of classification tests for each soil layer. The final column presents the mean water content of each soil layer. The SQL statement queries the database based on the soil layer references. Two points can be demonstrated through this prototype:

- The database implementing the proposed database model can handle the multiple interpretations that may exist in a project. In this case, four stratigraphy scenarios based on three boreholes are developed and stored in the database.
- Through this proposed geotechnical database model, the sophisticated data navigating process can be buried inside the database. What the user needs to handle is a simple search bounded on the soil layer which can be conducted through a standard SQL statement eg. Figure 5-11.

5.4 Database Model Evaluation

With the development of the hybrid database model, the geotechnical database model is further evaluated against the following criteria:

- **Completeness**

The model inherits the data structure from the well established AGS format, and hence can handle the data requirements of most geotechnical practices. In addition to the traditional geotechnical borehole data, the geotechnical database model developed provides support for stratigraphy data and variogram data via XML documents treated as objects.

- **Nonredundancy**

One major characteristic of the proposed geotechnical database model is the decoupling of the geotechnical test data from the stratigraphy data. Such an approach allows the geotechnical test data to be shared between different stratigraphy schemas or projects without reproducing the data in the database.

- **Enforcement of business rules**

The variogram data is bounded with the soil layer and the spatial relationships between geotechnical test data can be inferred through the stratigraphy data. The internal relationships for geotechnical data, especially the spatial relationships, are clearly defined in the geotechnical database model through stratigraphy data.

- **Data reusability**

One objective for developing the geotechnical database model is to enable the sharing of the data between different parties and projects. The database model decouples the geotechnical borehole data from the stratigraphy definition and releases the subjective constraints of geotechnical test data. Meanwhile, with the stratigraphy data in the database model, the spatial

domain can be easily reloaded.

- **Stability and flexibility**

Both the stability and flexibility of AGS format are inherited as the borehole data are defined directly following a well established format which has been widely accepted. Meanwhile, the stratigraphy data provides a high level abstraction on borehole test data. Using the stratigraphy data, the borehole test data can be seamlessly linked to and managed by the GIS thematic layer. For instance, a dataset can be generated based on the soil layer category.

- **Elegance**

Using XML for variogram data and stratigraphy data presents an integrated, efficient, consistent and simple way to store data. Without the XML technology, both variogram and stratigraphy data have to be separated into discrete pieces in the database without any physical meanings.

- **Communication**

With the UML diagram and documentation, the concepts and terms used in this database model have been clearly defined and explained.

5.5 Summary

A hybrid data model based on the NADM model is presented. The model decouples the geotechnical data and the soil layer stratigraphy category and manages them separately. It provides a context where geotechnical data can be shared and reused based on their category and classifications.

The data model implementation can contain multiple-perspective views of the same object. This is affirmed through the illustration example. Site exploration may be conducted in multiple stages, whenever new information is available, it should be combined with existing information to achieve a better understanding. Sometimes a new frame of reference needs to be constructed and the information needs to be reorganized and re-categorized. These requirements can be fulfilled by this

proposed database model.

The search through soil layer categories is the most natural and most frequently used search criterion for the geotechnical data collecting process. The prototype shows that this task can be accomplished easily using this database model. By defining the layer, surface, and stratum relationships inside the database, a high level abstraction of geotechnical data can be obtained. This flexibility is difficult to be achieved by using other existing models. The proposed model, together with a subsurface stratigraphy model, provides a logical solution for accessing geotechnical data through spatial locations.

The spatial characteristics of geotechnical data are not explored in this chapter. However, the extensibility of the model makes it easy to connect a geometry archive through a GIS gateway. A GIS prototype will be developed in the following chapter and this model provides a key component to fill the gap between GIS and DBMS and allow them to be integrated tightly.

Chapter 6 *CrystalGround: A Demonstration Application for the Proposed Database Model*

In this chapter, a demonstration application, *CrystalGround*, is presented. The purpose of developing *CrystalGround* is to demonstrate the potential of the database model in a manner more familiar to geotechnical engineers. *CrystalGround* will showcase an interface example that is independent of the physical interpretation of the database model. The demonstration will illustrate how the proposed database model can be implemented and integrated into the geotechnical data analysis process. A geostatistical kriging tool is integrated into the application. *CrystalGround* provides functions to create an interpolated borehole log at any location within the domain specified by the user. By extracting the stratigraphy information contained in the database, *CrystalGround* can also create soil section profiles on demand. As its name implies, *CrystalGround* is a tool transparently delivering the underground information defined in the geotechnical database.

Although the proposed models are composed of general rules that can be adopted in any development, it is difficult to avoid the use of some vendor specific components at the implementation level. *CrystalGround* is constructed using ESRI MapObject2.0 and Oracle 8i database on the Microsoft Win2000 platform.

6.1 System Analysis

The interactions between external users and the system under consideration are presented by a *use case* diagram shown in Figure 6-1. The *use case* diagram is a standard practice for capturing functional requirements in software design. It defines a goal-oriented set of interactions between external users and the application under development. The use case diagram in Figure 6-1 shows three external users and four use cases for *CrystalGround*. The three external users in Figure 6-1 are defined as

follows:

- ***Data Administrator***

Data administrator is the person responsible for database management. The function of the data administrator includes the input and the retrieval of geotechnical data from the database, user management, and data backup.

- ***Geotechnical User***

Geotechnical user is the person who uses this application to obtain geotechnical information. The geotechnical information could be in the form of a soil borehole log report or a soil profile.

- ***Data Analyst***

Data analyst would conduct data analysis on the geotechnical data to obtain the data description and estimate the geotechnical data values. In this study, the data analyst specifically refers to an analyst who conducts geostatistical analysis on geotechnical data. The data analyst is a specialist on data analysis and the analysis results will be used by geotechnical user.

The four use cases defined for *CrystalGround* in Figure 6-1 are described below:

- ***Create Borehole Log***

When a geotechnical user requires a geotechnical borehole log report, the user will interact with the application using this *use case*. The result of this *use case* will be a geotechnical borehole log report, where the precondition is that a geotechnical report template must exist. The process for this *use case* should include:

- a) Load the existing borehole locations in the study area from the geotechnical database.
- b) Define the position of the borehole where information are required.
- c) Retrieve, assemble and organize the data estimated or interpolated for the subsoil condition at the specified location.

Figure 6-1 Use case diagram of the demonstration application

- d) Present the information using the predefined borehole log report template.

- **Create Profile**

When a geotechnical user requires a soil section profile from the system, this *use case* will be used. The output of this *use case* will be a set of three-dimensional polylines (a polyline is a series of straight line segments). These polylines are retrieved from the corresponding surface files. The precondition of this *use case* is that these surface files must exist. The process for this *use case* would include:

Chapter 6 *CrystalGround: A Demonstration Application for the Proposed Database Model*

- a) Define the 'route' along which a vertical section of the soil profile is required; the route can be curved on plan.
- b) Retrieve the stratigraphy information from the database. The stratigraphy data will provide access to the surface files.
- c) Query the surface files to generate the lines of intersection between the vertical projection of the 'route' and the soil layer surfaces.

- ***Estimation***

When a data analyst tries to obtain a soil property at a particular spatial location where no borehole data are available, this *use case* will be used. The result of this *use case* will be the estimated soil property. The precondition for the *use case* is that the spatial description of the specified soil property is available. The process of this *use case* includes the following tasks:

- a) Identify the soil layer name based on the spatial coordinates.
- b) Create a dataset based on the soil layer name.
- c) Query the spatial soil property description and estimate the soil property values based on the available information stored in the database. In this study, the estimation of a particular soil parameter is achieved using geostatistical kriging and the spatial description is defined by means of the spatial variogram for that particular soil parameters.

- ***Retrieve data***

The data administrator interacts with the system through this *use case*. The data will be used by the geotechnical user or the data analyst. The data administrator triggers this *use case* when geotechnical data need to be retrieved from the geotechnical database. Not all the geotechnical data stored in a database are relevant and hence required for a specific situation. Normally only a set of data sharing some common characteristics are needed. The result of this *use case* will be the retrieved dataset. The precondition for this *use case* is that the data tables for the geotechnical data must exist. The

process of this *use case* will include obtaining data search criteria and assembling data following the requested format.

6.2 System Implementation

CrystalGround is developed as a client-server system in a Win2000 network. Figure 6-2 gives the deployment diagram of the demonstration application. The components of *CrystalGround* are divided into two parts: client side application and geotechnical database server. The geotechnical database server implements the proposed database model in an Oracle8i database. The client side application includes a map interface developed using ESRI MapObject2.0 together with a report module, an estimator module and a data manager module.

6.2.1 Geotechnical Database Server Implementation

Based on these use cases, Figure 6-3 shows the object classes that need to be stored using a static class model diagram. For *CrystalGround*, only three geotechnical test classes are presented in Figure 6-3. *SPT* class stands for SPT in-situ test, *Classification* class stands for soil classification test, and *TRIG* class stands for triaxial test. However, other geotechnical test classes can be added easily using a similar approach when required. All the classes are implemented from the proposed hybrid database model excepting two new classes: *Raster* class and *Point* class.

Raster class and *Point* class are two standard GIS classes used to represent soil subsurface and geotechnical boreholes in a GIS platform for this application. The *Raster* class in Figure 6-3 refers to raster model of soil surfaces at the soil layer interfaces. The *Point* class defines basic point geometry together with a set of attributes. The geotechnical boreholes locations are represented by points (a point on a 2D plane is defined by (x, y)) on a GIS layer in this application. The point class builds up a “one to one” relationship with the *Hole* class through the definition of its attributes. Meanwhile, the *Raster* class is used to derive soil category together with the *Stratigraphy* class. Through the relationship established between the points and

Figure 6-2 Deployment diagram of CrystalGround

Figure 6-3 Static class model for CrystalGround

Chapter 6 *CrystalGround: A Demonstration Application for the Proposed Database Model*

geotechnical boreholes and the relationship built between rasters with soil stratigraphy, a procedure to access geotechnical data through a GIS platform is thus established.

The component on the server side is a geotechnical database. The database is built up using an Oracle8i database with XML support. The geotechnical data classes presented in Figure 6-3 are implemented as Oracle objects and stored in the object tables. The implementation is conducted using Oracle PL/SQL procedures. The lists of the procedures for creating this database table are presented in Appendix B.

The *Point* class and *Raster* class are two standard GIS classes. The *Point* class in this application is created as ESRI point shape file using an ESRI specific managing tool, ArcCatalog. Figure 6-4 shows the attribute fields of the points shape file. The first three fields are ESRI specific fields for the point geometry. In this example, the fields “Project”, “HoleName” and “Contractor” are attributes used to identify the corresponding geotechnical borehole in the application. Whenever these attributes are not sufficient to build up a unique “one-to-one” relationship between a point and a geotechnical borehole location, more attribute fields can be added by editing the shape file.

Of the two available surface models, vector model and raster model, the raster model is used in the development of *CrystalGround*. The *Raster* class is implemented using ASCII formatted files. Figure 6-5 shows an example of a raster class represented using ASCII format. The first six rows of the file are relevant parameters about the raster. These parameters include the number of rows and columns, the coordinate of the lower left cell, the cell size and the value used to represent a null data cell in the raster. The other rows are the cell values for the raster following a row-column sequence. In the example of Figure 6-5, the raster has 120 columns and 50 rows, the coordinate of the lower left cell is given as (2990.70, 2988.17). The cell size is 1.0 and a cell without data is given a value -9999.

Figure 6-4 Attributes of point objects for representing geotechnical boreholes

Figure 6-5 Raster file in ASCII format

6.2.2 Client Side Implementation

Figure 6-2 shows four client modules have been developed for *CrystalGround: GIS Interface, Data Manager, Estimator* and *Report Generator*. These modules are developed using Visual Basic 6.0. The VB codes are appended at Appendix C.

GIS Interface

The GIS interface is developed using ESRI MapObject ActiveX control. It provides a means for the geotechnical user to interact with the system. Figure 6-6 shows a snapshot of this GIS interface. When the user is presented with a project plan through this interface, the user can specify the location where a soil profile or a borehole log report is required. Based on the location of the ‘route’ as defined by the user in a horizontal plan, a vertical plan can be generated. This vertical plan will intersect with the soil layer surfaces which are the interfaces between different soil layers. As soil layer surfaces are represented using raster model, using the basic spatial query functions provided by the ESRI MapObject control, a set of

Chapter 6 *CrystalGround: A Demonstration Application for the Proposed Database Model*

(a) Select locations for creating borehole log reports

(b) Define a route for creating soil profile

Figure 6-6 Snapshots of the GIS interface of CrystalGround

intersecting lines can be created. The set of intersection lines between vertical surfaces projected from the user defined polylines and the soil interfaces are stored as ESRI shape files. These shape files also can be displayed in this GIS interface. Further requirements such as labeling and marking these lines can be incorporated, but these were not done in this study.

Data Manager

This module handles the data communications with the database server. The *Data Manager* module is the kernel module for the client application. The *Estimator* module will interact with the *Data Manager* for the soil property variograms and geotechnical dataset. The *GIS Interface* also obtains spatial data through the *Data Manager* module. The *Data Manager* module connects to the database server through two interfaces: Oracle OO4O and ESRI SDE. Oracle OO4O (Oracle Objects for OLE) is an Oracle middleware that allows native access to Oracle from client applications using the Microsoft Object Linking and Embedding (OLE) standard. The ESRI SDE (Spatial Data Engine) is an ESRI specific interface for accessing spatial data. In this application, SDE is used to access the point objects stored in the database representing boreholes in a GIS platform.

Report Generator

This module is used to create a borehole log report based on the geotechnical data extracted from the database. It is developed using ActiveReports 1.1 ActiveX control. A predefined report template was used. Figure 6-7 shows an example of a template for geotechnical log report. The module takes the extracted geotechnical data as input; retrieves the geotechnical data and presents data according the format specified in the report template.

Figure 6-7 A template for borehole log report

Estimator

This module performs the crucial function of geostatistical estimation for geotechnical data. It includes a dynamic linked library called Kt3D.dll. This dynamic linked library is built up using existing Fortran code of the geostatistical library GSLIB (Deutsch & Journel, 1998). The *Estimator* module takes the variogram XML document and a soil parameter dataset as input and conducts geostatistical kriging to give an estimated value of the given soil parameter at the specified spatial position. The dynamic library is declared as a visual basic function in this module as shown in Appendix C.

6.3 Summary

The purpose for the development of *CrystalGround* is to demonstrate how the proposed database model can be implemented and to showcase its advantages. The outline of *CrystalGround* is described in this chapter and the code is appended in Appendix C. Only a few key features are included in the prototype. These features are chosen based on the consideration of what a geotechnical engineer would require from a set of geotechnical boreholes. The features chosen particularly include three critical aspects: the creation of geotechnical cross sections; creation of the likely soil profiles at a given location; and an estimation of soil properties.

The features developed in this demonstration application are highly dependent on the implementation of the proposed hybrid database model. The *Data Manger* module retrieve the geotechnical data based on the stratigraphy data. The *Estimator* module relies on the variogram data to conduct geostatistical estimation. Without these data, the functionalities provided in *CrystalGround* would be very difficult to achieve. *CrystalGround* also demonstrates a way the GIS components can be linked to the database model. By representing the boreholes with the standard GIS *Point* and linking the GIS *Raster* with stratigraphy data, the proposed database model can be fully integrated with a GIS platform. In the next two chapters, two case studies are presented using *CrystalGround* to further illustrate the advantages of the

Chapter 6 CrystalGround: A Demonstration Application for the Proposed Database Model

proposed database model.

Chapter 7 Case Study: Stratigraphy Modelling and Geostatistical Kriging

Using *CrystalGround*, the demonstration application described in the previous chapter, two case studies were conducted. As the framework of *CrystalGround* was constructed based on the proposed database model, the objective of these two case studies was to demonstrate the rationale, applicability and advantages of the proposed database model through the implementation of the *CrystalGround*.

This chapter describes the first case study, in which a stratigraphy modelling and geostatistical kriging of a soil property was conducted. In order to eliminate the uncertainties associated with a geotechnical model derived from real and limited borehole data, and also to avoid being distracted from the study objective, the case study was conducted using a synthetic dataset. The advantage of using a synthetic dataset is that the study domain is “fully defined” for the analysis (Benson & Rashad, 1996). A similar approach has also been used to conduct a study on the sampling plan configuration (DeGroot & Baecher, 1993). Taking advantage of a fully defined domain, a process for creation, storage and usage of the stratigraphy data and variogram data is described.

7.1 The Synthetic Dataset

An important consideration for the synthetic dataset representing the subsurface domain should be realistic and geotechnically consistent. The following factors were considered in the creation of this synthetic dataset:

- 1) The size of the study domain should be representative of a typical geotechnical project. And it should not be too large so as to avoid placing

excessive demand on storage requirement and computing power. The resolution of the study domain should also provide sufficient detail.

- 2) For the spatial variogram demonstration, only a single variable was considered in the dataset to simplify the process. The chosen variable should be a continuous soil variable which can be treated as any continuously varying parameter. It can be any soil parameter commonly encountered in geotechnical field, such as water content, soil shear strength, soil permeability, or soil stiffness. One critical requirement is that the variable should exhibit a realistic spatial variation. This means that the variable should be created through a spatial model mirroring a real soil parameter.

In view of the above considerations, the dimension of the domain chosen was 500 m x 500 m in plan and 50 m deep. The size was quite representative of the normal scope of a small to medium size construction project. The domain was further divided into 200,000 rectangular solid. Each element had a dimension of 5 m x 5 m x 0.25 m (length x width x height) and 8 nodes at the corners of the rectangular solid element. A Cartesian coordinate system was used. The X axis was associated to the east direction, the Y axis was associated to the north direction, and the Z axis was associated to the elevation with a positive direction going upward. Every node of an element could be characterized by a stratigraphy identifier and was assigned a value for the soil parameter. There were altogether 214,221 nodes and the whole domain was treated as fully defined.

7.1.1 Stratigraphy Data

Stratigraphy data defines the internal relationships of interfaces between soil layers. To model the soil interfaces, the raster model was adopted. Each soil interface/boundary was represented by a raster with 101 x 101 cells. The size of the cell was 5 meters. For this particular study, the domain was divided into four soil layers as shown in Figure 7-1: *Fill*, *MC* (Marine Clay), *F2* and *OA* (Old Alluvium). The stratigraphy was based approximately on the configuration of the soft marine

Figure 7-1 Configuration of the synthetic study domain

clay deposits reported by Clark & Prebahren (1987). There were eight soil surfaces on the interfaces between the four soil layers. The top surface of the *Fill* layer and bottom surface of the *OA* layer were configured as constants at elevations 50 m and 0 m respectively. They represented the top and bottom boundary of the study domain. This configuration presented a basic scenario where a soil layer (*MC*) was interspersed by a thin discontinuous soil layer (*F2*). Since the bottom surface of the *Fill* layer and the top of the *MC* layer referred to same interface and so did the bottom surface of *MC* layer and the top surface of the *OA* layer, there were four soil surfaces that needed to be generated: top, bottom surfaces of *MC* layer and top, bottom surface of *F2* layer.

For the top and bottom surfaces of the *MC* layer, the soil surfaces were created through the following steps.

Step 1: A data generator program was used to create a sample dataset for each surface. The dataset comprised 100 entry values representing elevations of the soil surface. The dataset was assumed to follow a normal distribution and was randomly generated. The mean value and standard deviation for creating the sample dataset were predefined. For the top surface of the *MC* layer, the mean value was set as 42 m and the standard deviation was chosen as 1.0 m. For the bottom surface of the *MC* layer, the chosen mean value was 10 m and the standard deviation was 1.0 m. These predefined parameters and other parameters used in creating the soil surfaces were obtained through a trial and error process. The parameters were chosen to create a configuration that exhibited similar characteristic as the profile of the marine clay reported by Clark & Prebahren (1987). Similar results could also be achieved through other parameter combinations.

Step 2: As the surface was modelled using a raster model that had 10201 (101 x 101) cells, 100 cells were randomly selected from the 10201 cells and assigned with the values of the synthetic dataset obtained in the previous step. For the other cells (10101 cells) that were not assigned with a value

from the synthetic dataset, each cell was given a value interpolated from the 100 cells which had been assigned with values derived from the synthetic dataset. The interpolation was conducted using the spline interpretation method through ESRI *Spatial Analyst* program.

Step 3: A special process was adopted for creating the F2 layer. The surfaces of the F2 layer were first created using the procedure described above. Then linear drifts were added to the two surfaces to cause the two surfaces to clip each other. By changing the parameters of the linear drifts through a trial and error process, the top surface and the bottom surface of the F2 layer would clip each other and generated the scenario shown in Figure 7-1. The ESRI *Spatial Analyst* program was used to remove the clipped parts of the two surfaces. The values for cells in the removed area were set as “NoData”. The “NoData” denoted that data was not available for that cell in a raster model.

With the created soil surface data, each node was assigned to a soil layer based on the spatial relationship of the node and soil surfaces. The sample dataset for generating the surfaces is given at Appendix D. It should be noted that the parameters and interpolation methods used in the process of stratigraphy generation were only one of the many possible options available. The main objective of this exercise was to create a stratigraphy configuration with the desired characteristics. There were no preferred parameters or interpolation methods in the stratigraphy generation process.

7.1.2 Simulation of Soil Parameter Values

Once the stratigraphy had been defined in the study domain, an additional task was to choose a soil variable for spatial analysis. The chosen soil variable was the undrained shear strength. Due to the lack of reported information on the spatial variation of the undrained shear strength of the soft Marine Clay, it was decided that the variogram parameters obtained from another field study would be adopted. The required parameters were obtained from a study on the undrained shear strength of a

clay deposit in the James Bay area of Quebec, Canada (Soulie et al., 1990). The undrained shear strengths were obtained by means of Nilcon vane tests carried out on the site in 1979, 1980 and 1981. The reported data showed a normal distribution with a global mean value of 31.5 kPa and a standard deviation of 7.09 kPa. The variogram of the data was found to fit a spherical model without nugget effect. The vertical range of the variogram was 3 meters. The horizontal range of the variogram was 30 meters.

In order to honor the distribution and spatial variation of the soil variable, the *non-conditional Sequential Gaussian Simulation (SGS)* program in *GSLIB* was used to create the synthetic dataset for the undrained shear strength. The Sequential Gaussian Simulation method was conducted by first creating a random node sequence, and then the value of the soil parameter at each node was simulated as a Gaussian random variable following the random sequence. The mean value and the standard deviation of the random variable were obtained from the estimated value and variance by simple kriging the *nearby* node values. The *nearby* nodes refer to the nodes within a certain distance around the node where the value of the soil variable was being sought. This distance was defined by the user based on the spatial variation of the soil variable. Whenever the *nearby* node values were not available, the global mean and standard deviation were used. The Sequential Gaussian Simulation drew a realization that honored the data and reproduced aspects of the pattern of the spatial dependence. (Goovaert, 1997; Deutsch & Journel, 1998). In general, multiple realizations would be created. However, for this study only one realization was generated. The result was used as the soil parameter values in the synthetic dataset.

In this study, due to the large number of nodes in the study domain, the study domain was divided into 10 x 10 x 10 sub-domains and simulated separately. The size of each sub-domain was a cube measuring 50 m x 50 m in plane and 5 m in height. The choice of this sub-zoning was decided with the following considerations:

- The maximum computing node constraint of the program used to conduct the Sequence Gaussian Simulation.
- To mediate the ergodicity fluctuation, where the mean value and standard deviation of the simulation result were different from the mean and standard deviation of the sample dataset.
- To reduce the artefact effect at the interfaces of these sub-domains.

The simulated data were stored in 100 files using ASCII format. Altogether the realization created 139 megabytes of data for the undrained shear strength in the synthetic ground domain. A query program was developed to retrieve these data based on the plane coordinate value (x, y) of a point. The program would first determine which data file of the 100 files to open based on the (x, y) value. It would then scan the data inside the relevant file and retrieved the value at the nodes with the closest (x, y) coordinate reference. Thus, all the retrieved values were from nodes with the same (x, y) but different depth. However, only the data at the nodes associated with the MC layer had the desired physical meaning and were thus relevant to the purpose of this case study. By combining the stratigraphy information and the spatial variation information from these two references, a synthetic dataset with the desired characteristics was thus created.

7.2 Application of *CrystalGround*

7.2.1 Stratigraphy Study

This section presents the process of generating the stratigraphy information using the data extracted from the idealized synthetic ground. The study was conducted on an area measuring 240 m x 100 m in plan, a size representative of a typical small construction project.

In the absence of existing geotechnical information, the first step in the soil investigation would involve the planning of the number of boreholes to be drilled

and the layout of these boreholes. Many sample planning strategies were available. DeGroot & Baecher (1993) illustrated four commonly used sampling layouts: Systematic (Regular); Nested; Stratified Random and Random. The geotechnical engineer would have to consider the spatial location of the boreholes so as to obtain as much information as possible with the constraint of a limited budget for soil investigation.

For the purpose of the current study, four trial sampling plans were chosen in this study. All the sampling plans were systematic (regular) plans. Figure 7-2 shows the layout of the four sampling plans. Sampling plans 1 and 2 each had 8 boreholes but the locations of the boreholes were arranged differently. Plan 3 and plan 4 had 11 and 21 boreholes respectively. Aggregating all the layouts into consideration, a total of 23 unique boreholes were required. The distance between boreholes varied from 40 meters to 80 meters. Using the query program developed in this study, the stratum data for these boreholes were retrieved from the synthetic dataset. Figure 7-3 shows the stratum information for these 23 boreholes. Based on the borehole stratum information, the soil layer information was summarized in Table 7-1.

Stratum Data

The process of defining and inputting the stratum data was straightforward by following the borehole logs shown in Figure 7-3. For instance, there were 5 strata defined in H1 and 3 strata defined in H3. The stratum data was an independent unit in the proposed database model. It was not constrained by the soil layer and stratigraphy schema. Although there were four sampling plans, the stratum data only need to be input into database once.

Raster Surface Data

Raster surfaces were generated using the ESRI *Spatial Analyst* program following the same procedure of generating surfaces in the synthetic dataset. Instead of using the sample dataset shown in Appendix D, the datasets used for the surfaces generation were based on the elevation values obtained from the boreholes. Taking

Figure 7-2 Borehole plans for stratigraphy interpolation

Figure 7-3 (a) Borehole logs (H1-H8)

Figure 7-3 (b) Borehole logs (H9-H16)

Figure 7-3 (c) Borehole logs (H17-H23)

Table 7-1 Soil layer information for sample boreholes

Hole No.	Location		Soil Layer							
	X (m)	Y (m)	Sand Fill		Marine Clay		F2		Old Alluvium	
			Top (m)	Bottom (m)	Top (m)	Bottom (m)	Top (m)	Bottom (m)	Top (m)	Bottom (m)
H1	90	300	50.0	43.26	43.26	9.68	31.80	29.69	9.68	0.0
H2	170	300	50.0	42.42	42.42	8.53	32.57	30.97	8.53	0.0
H3	250	300	50.0	42.20	42.20	8.87	-	-	8.87	0.0
H4	330	300	50.0	40.80	40.80	9.93	30.46	28.41	9.93	0.0
H5	90	400	50.0	41.61	41.61	8.99	33	28.88	8.99	0.0
H6	170	400	50.0	39.85	39.85	9.64	30.06	29.98	9.64	0.0
H7	250	400	50.0	41.92	41.92	8.31	-	-	8.31	0.0
H8	330	400	50.0	42.00	42.00	10.19	30.59	29.47	10.19	0.0
H9	130	350	50.0	40.60	40.60	8.99	32.58	29.57	8.99	0.0
H10	210	350	50.0	41.17	41.17	8.43	-	-	8.43	0.0
H11	290	350	50.0	42.36	42.36	11.09	29.82	29.30	11.09	0.0
H12	90	350	50.0	41.25	41.25	10.37	32.13	28.89	10.37	0.0
H13	130	300	50.0	43.05	43.05	9.07	32.45	30.62	9.07	0.0
H14	130	400	50.0	39.76	39.76	8.52	32.24	29.16	8.52	0.0
H15	170	350	50.0	40.22	40.22	8.62	31.52	30.42	8.62	0.0
H16	210	300	50.0	42.60	42.60	9.88	31.59	31.56	9.88	0.0
H17	210	400	50.0	42.35	42.35	8.83	-	-	8.83	0.0
H18	250	350	50.0	41.25	41.25	8.54	30.21	30.07	8.54	0.0
H19	290	300	50.0	41.25	41.25	8.74	-	-	8.74	0.0
H20	290	400	50.0	41.76	41.76	9.38	30.18	29.63	9.38	0.0
H21	330	350	50.0	41.46	41.46	12.52	30.47	28.53	12.52	0.0
H22	150	350	50.0	40.06	40.06	8.61	32.12	29.96	8.61	0.0
H23	270	350	50.0	42.25	42.25	9.66	30.01	29.72	9.66	0.0

the bottom surface of *Sand Fill* layer for Plan 1 as an example, there were 8 boreholes (H1, H4, H5, H8, H16, H17, H22, and H23) and all the boreholes showed the presence of the *Sand Fill* layer. The dataset for the bottom surface of the *Sand Fill* layer then included 8 data points: elevation values (z values) together with corresponding location coordinates (x, y values). Applying the *Spatial Analyst* program to the dataset allowed the raster surfaces to be interpreted. The interpreted surfaces then need to be checked and might be modified for consistency. For instance, the *Spatial Analyst* program was used to remove the clipped part of the top and bottom surfaces for the *F2* layer. After the consistency check, the resulting raster surfaces were stored in ASCII format and the names of the surface files could be used as locators for the surface data.

Soil Layer Data and Stratigraphy Data

The soil layer data defined two types of information: the boundary and the strata in the soil layer. The boundary of the soil layer was defined by the top and bottom surface data. The strata belonged to the soil layer were defined by an array of strata data. The relationships between different soil layers were built up using the XML stratigraphy data developed in this study. List 7-1 shows a SQL statement for inserting one row of stratigraphy data into a database table named *stratigraphy_table*. The stratigraphy XML data was stored in this table together with the information about the project name and engineer name. The soil layers *SandFill*, *MC* and *OA* had a “sibling” relationship as defined in Chapter 4. They were listed from top to bottom sequentially. The *F2* soil layer was a lense layer inside the *MC* soil layer, the “parent-son” relationship between the *F2* layer and the *MC* layer was represented in the XML data by embedding the *F2* layer element into the *MC* layer element.

```

insert into stratigraphy_table values
('SyntheticData', 'GeoStudy',
'<?xml version="1.0"?>
<!DOCTYPE STRATIGRAPHY [
<!-- DTD for stratigraphy-->
<!ELEMENT STRATIGRAPHY (LAYER+)>
<!ELEMENT LAYER ANY>
<!ATTLIST LAYER
    name ID #REQUIRED
    topsurface CDATA #IMPLIED
    bottomsurface CDATA #IMPLIED
    description CDATA #IMPLIED
>
]>
<STRATIGRAPHY>
  <LAYER name="Sandfill" topsurface="Synthetic_top" bottomsurface="mc_top"
description="F2 lense"/>
  <LAYER name="MC" topsurface="mc_top" bottomsurface="mc_bottom"
description="Well graded, well compacted, low compressibility material">
    <LAYER name="F2 " topsurface="f2_top" bottomsurface="f2_bottom" description="f2 lense"/>
  </LAYER>
  <LAYER name="OA" topsurface="mc_bottom" bottomsurface="synthetic_bottom"
description="Sandish rock"/>
</STRATIGRAPHY>')

```

List 7-1 PL/SQL procedure for storing the stratigraphy XML datagram

Soil Profile Generation Using CrystalGround

The subsurface data prototyped in the proposed database model were defined and entered into the database. Using the profile generation function of *CrystalGround*, the soil cross-section profiles could be efficiently created. As a demonstration application, the output profiles of *CrystalGround* were stored in ESRI *shape* format. The soil profiles obtained through *CrystalGround* were further processed to improve the output quality. In this case, ArcGIS 9.1 was used for post-processing the results from the application by adding labels, notes and legends.

Figure 7-2 (a) also shows four sections cutting across the site (Section A-A, B-B, C-C, and D-D) used for demonstration in this case study. The sectional soil profiles generated from limited borehole data from the various sampling plans are presented in Figure 7-4. For comparison, the ‘actual’ soil profile derived from the exhaustive synthetic dataset is also shown in these figures.

As there were only minor variation in the elevations of the top and bottom surfaces of the Marine Clay (*mc_top* and *mc_bottom*), the derived soil profiles were evaluated based on the coverage of the *F2* layer. In terms of coverage of the *F2* layer, the interpolated stratigraphy information from the 21 boreholes (Plan 4) gave a better simulation of the “reality”. Three of the four sections (Section A-A, Section B-B, Section C-C) for Plan 4 showed good agreement with the original profiles. The existence of the *F2* layer was underestimated in section D-D for all the sampling plans. Plan 1 and Plan 2 each had 8 boreholes. The coverage of *F2* layer was overestimated in Section B-B and Section C-C for Plan 1. This could be partially attributed to the differences in the geotechnical information revealed in boreholes H17 and H23. H23 which was used in the sampling Plan 1 showed the presence of the *F2* layer. In contrast, the *F2* layer was absent in borehole H17 used in sampling Plan 2.

There were three more boreholes (H9, H10, and H11) in Plan 3 as compared to Plan 2, the interpolated stratigraphy results appeared to give no significant

improvements. A review of the boreholes in the Plan 2 and Plan 3 showed that H10 had no contribution to the interpolation of the *F2* Layer since the layer was absent from the borehole record. The contribution of H9 was also small in this case as it was located in the middle of four boreholes (H1, H2, H5, and H6) and all the four boreholes showed the existence of the *F2* layer.

Extracting soil layer stratigraphy information from borehole records is a complex process involving data interpolation, rules of thumb, engineering experience and judgment. Figure 7-4 shows the effect of sampling plan and density of boreholes on the interpolated results. The limited borehole data, the chosen interpolation method and the subjective characteristic of the interpolation process all contributed to the final outcomes. The results presented in Figure 7-4 clearly demonstrated that the stratigraphy information could be built up using the proposed database model. In this database, the borehole data were shared by the four different scenarios. The only differences between the four different scenarios were the stratigraphy data, surface data and project information which contain the borehole references for each project. The stratigraphy information was stored in the database through the proposed XML schema. The internal relationships between various soil layers hence could be managed efficiently.

Figure 7-4(a) Section A-A for different sampling plans

Figure 7-4(b) Section B-B for different sampling plans

Figure 7-4(c) Section C-C for different sampling plans

Figure 7-4(d) Section D-D for different sampling plans

7.2.2 Soil Parameter Estimation

Careful consideration of the sampling plan is also required for soil variable analysis. The variogram estimation is significantly influenced by sample spacing. Superior estimates are provided by plans with a number of borings located at separation distance less than the autocovariance distance (DeGroot & Baecher, 1993). However, by taking advantage of the known variogram for the soil parameter, the sampling locations could be chosen through trial and error to ensure that the variogram derived from the sample data is close to the “real” variogram. Figure 7-5 shows the chosen test location plan with 28 holes. This configuration of sample locations is also similar to the sampling plan reported by Soulie et al. (1990). It should be noted that the only pre-condition for choosing the number and location of the boreholes was that the variogram derived from the samples should not deviate significantly from the “real” variogram. There was no other preference except the pre-condition. The configuration shown in Figure 7-5 was just one example of configurations fulfilling this pre-condition.

Variogram

In this case study for variable estimation, the objective soil parameter was the undrained shear strength. The values and distribution of the undrained shear strength had been built into the synthetic dataset of the artificial ground described in previous sections of this chapter. Using a search algorithm developed in this study, the undrained shear strength data within the Marine Clay layer at location (x, y) were retrieved at 1-meter intervals vertically. Using the undrained shear strength as input data, the variogram was obtained through the *gamv* program, a subroutine of *GSLIB* for extracting variogram from irregular spaced data. Figure 7-6 gives the analysis results of the variogram modeling for the horizontal direction and the vertical direction respectively. For the horizontal variogram, the variogram was calculated for four directions: North, North East, East and South East.

Figure 7-5 The location plan for the sampled data and estimated data

(a) Horizontal variogram

(b) Vertical Variogram

Figure 7-6 The variogram model of C_u for the Marine Clay

The derived variogram model showed a horizontal range of 30 m and variogram of 50 kPa². The vertical variogram is presented in Figure 7-6(b), where the derived variogram showed a range of 3 m and a sill of 50 kPa² variogram. The results were almost identical to the actual variograms used to generate the synthetic dataset. The results were compiled into a XML document using the schema developed in this study. List 7-2 shows the PL/SQL procedure used to store the variogram data into the database. In this list, the variogram data was inserted into the *variogram_table* together with a soil layer reference pointed to a soil layer named *MC* from the soil layer table *layer_table*. And the data type was defined as *FV*, representing the Field Vane test.

```

declare
    lyr ref layer_type;
begin
    select ref(a) into lyr from layer_table a where name='MC';
    insert into variogram_table values(
        lyr, 'FV',
        '<?xml version="1.0"?>
        <VARIOGRAMS>
        <NUGGET>0.0</NUGGET>
        <VARIOGRAM>
        <TYPE>Spherical</TYPE>
        <COEFFICIENT> 7.9</COEFFICIENT>
        <MAX_HORIZONTAL_RANGE>30
        </MAX_HORIZONTAL_RANGE>
        <MIN_HORIZONTAL_RANGE>30
        </MIN_HORIZONTAL_RANGE>
        <VERTICAL_RANGE>3.0</VERTICAL_RANGE>
        </VARIOGRAM>
        </VARIOGRAMS>
    );
end;
```

List 7-2 PL/SQL procedure for inserting variogram table

Geostatistical Kriging Using CrystalGround

The geostatistical estimation module of *CrystalGround* was evaluated by using it to obtain an estimation of the soil undrained shear strength of the Marine Clay layer at specified locations. Figure 7-5 also shows the locations of the three boreholes (BH-A, BH-B, and BH-C). The undrained shear strengths of the Marine Clay layer at the three boreholes were estimated at 0.5 m interval vertically. The confidence interval combines the estimate and error variance and an assumption of normality was applied as suggested by Isaaks & Srivastava (1989). Figure 7-7 shows the Gaussian-type confidence interval centered at the estimated value. It shows the estimates obtained from the kriging progress together with the lines marking one standard deviation (68.3% confidence interval) from the mean. From the “defined” exhaustive dataset, the “actual” undrained shear strength values at the three boreholes can also be obtained. These data were plotted in Figure 7-7 for comparison. As the variances of the dataset were almost constant in this study domain, the change of the krigged variance was relatively small. The kriging estimates demonstrated a smoother variation with less abrupt scatter with depth. This was inherent in the geostatistical model where the soil variable was modeled as a continuous random variable.

The estimated data for BH-B fitted the original defined data reasonably well as only 4 values were outside the one standard deviation interval, comparing 8 values for BH-A and 11 values for BH-C. The results for BH-A and BH-C could potentially be improved by subdividing the Marine Clay into two layers delimited by the *F2* layer and estimating the data separately. Nevertheless, this demonstration has shown conclusively that the proposed database model was capable of storing the variogram information and could support the estimation function of soil parameters.

Figure 7-7(a) The comparison between the undrained shear strength estimation obtained by geostatistical kriging with the original defined data for BH-A

Figure 7-7(b) The comparison between the undrained shear strength estimation obtained by geostatistical kriging with the original defined data for BH-B

Figure 7-7(c) The comparison between the undrained shear strength estimation obtained by geostatistical kriging with the original defined data for BH-C

7.3 Summary

The case study was performed using a synthetic dataset. Special care was given to ensure that the synthetic ground encompassed as much realism as possible. Apart from this, the analysis process for borehole data sampling, the stratigraphy creation, the estimation of soil parameters at location between boreholes can also be applied to actual data.

The data were organized and managed based on the proposed database model. Except the surface data, all the data involved in the case study were stored in a database developed using the proposed database model. The raster data were stored in the ASCII format files outside the database and linked to the database through file locators (URL). The kernel of the database was the stratigraphy data, which contained the internal relationships between various soil layers and was stored in the XML schema developed in this study. Through the processes of stratigraphy analysis and soil parameter analysis, the following conclusion could be made:

- During the analysis process, borehole data, stratum data, surface data, stratigraphy data and variogram data were generated, stored and retrieved from the database. It demonstrated that the proposed database model could be implemented in a database to manage the geotechnical data
- The stratigraphy information was stored in an XML schema developed in the proposed database model. The *CrystalGround* program generated the soil profiles by inferring the right surface files from the stratigraphy data. The geostatistical estimation was conducted based on a dataset extracted from the database based on the soil layer category, and the soil layer category in turn was obtained through the stratigraphy inference. The output results have affirmed the validity of the stratigraphy centric database model.

- By implementing the proposed database model, the borehole data could be decoupled from the stratigraphy schema. The stratigraphy schema could be seen as a higher level abstraction above the borehole data. The borehole data could be categorized by the soil layer stratigraphy but were not restricted with the stratigraphy. As shown in the case studies, although four stratigraphy scenarios were conducted in the analysis, all the borehole data and strata data were input only once. The same borehole information could be shared between different scenarios or projects.
- Through the creation of soil surface, soil layer and stratigraphy data and by importing the data into the database, the study domain had been accurately defined in a 3D frame. Although only four sections were generated in the demonstration, *CrystalGround* could be used to generate soil profiles on demand as will be shown in the next case study.
- A variogram schema had been developed for the proposed database model. Through the geostatistical kriging process, it was shown that the variogram data could be stored, retrieved and used in geostatistical kriging as part of the functionalities of the proposed hybrid database model.

Chapter 8 Case Study: 3-D Model of the Subsurface Stratigraphy at a Failed Excavation Project

The previous chapter provides a demonstration using data derived from a synthetic site. In this chapter, *CrystalGround* is applied to borehole information of an actual high rise condominium development project in Singapore. During excavation for basement construction, the sheetpile retaining wall system failed, causing extensive damage to the adjacent road and a nearby substation. The failure was mainly attributed to the error in the deduced soil profile based on the 4 boreholes drilled in the original soil investigation programme. Subsequent to the failure, extra boreholes were drilled to gain additional information on the subsurface condition. The borehole data, both the original set and the post-collapse investigation data, were used in this case study. The purpose of the case study was to further test and evaluate the proposed database model through applying *CrystalGround* to manage geotechnical data derived from an actual project.

8.1 Borehole Information

The borehole data used in this study were derived from a condominium housing development project comprising 2 blocks of 12-story residential buildings. The geotechnical data comprised 11 boreholes derived from two stages of site investigations. The detailed borehole logs for the two site investigation are appended in Appendix E. The initial site investigation comprised 4 boreholes (BH-1, BH-2, BH-3, and BH-4). The design of the retaining structure and foundation, and the construction program was based on the geotechnical information deduced from the 4 boreholes obtained in the original site investigation. The post-failure site investigation comprised 7 boreholes (K-1, K-2, K-3, K-4, K-5, K-6, and K-7). These additional boreholes were specifically drilled after the collapse event as part of the post-failure investigation and it was intended to gain further insight into the

subsoil condition. Figure 8-1 shows the boreholes layout from the site investigations. Out of the 4 boreholes drilled in the original site investigation programme, only BH-1 showed the presence of a weak marine clay layer with a thickness of about 1.6 m. The existence and extent of the soft marine clay was under-appreciated. The 1.6 m thick marine clay stratum together with several nearby strata were combined and treated as a single firm clay layer. After the failure, an additional 7 boreholes were drilled. The data from the second stage of the site investigation revealed that the weak marine clay layer was more extensive than what was originally deducted from the initial four boreholes. Five of the 7 boreholes from the post-failure investigation showed the presence of this weak soil layer.

8.2 Application of *CrystalGround*

A process of managing the 11 borehole data and interpreted stratigraphy data using *CrystalGround* will be demonstrated through the following activities:

- The 11 borehole data were entered into the geotechnical database of *CrystalGround*, which implemented the proposed hybrid database model.
- The interpreted surface data, soil layer data and stratigraphy data were entered into the database of *CrystalGround*.
- Graphical presentations of vertical sections through the project site were generated.

8.2.1 Data Input

The borehole data were imported into the Oracle geotechnical database of *CrystalGround* through a set of PL/SQL procedures. The relationships between these tables and the attributes of each table are described in Chapter 6. The scripts to create these tables are appended at Appendix B. Examples of PL/SQL procedures are described below:

Figure 8-1 Layout of boreholes at the project site

Project information

The general information about a project was stored in the *project_table*. List 8-1 gives the PL/SQL procedure to insert three rows into the *project_table* for three projects. The three projects were defined subsequently for the original site investigation (BH-1~4), post-failure site investigation (K-1~7) and post-failure analysis based on the 11 boreholes. The contractor name was used to identify each project in this case study. Rather than using the real names of the contractors, the *contractors* for the three projects were given fictitious names as “*SI Ltd*”, “*ASIA Drilling Ltd*”, and “*GEO EXPLORATION*” respectively.

```

begin
  insert into project_table values (
    project_type('Proposed Erection of A condominium', 'BLK SomeWhere', 'DD
    Development Ltd', 'SI Ltd', 'David', '4 holes BH1-BH4', '23-sep-03'));
  insert into project_table values (
    project_type('Proposed Erection of A condominium', 'BLK SomeWhere', 'DD
    Development Ltd', 'ASIA Drilling Ltd', null, '7 boreholes k1-k7', '28-sep-03'));
  insert into project_table values (
    project_type('Proposed Erection of A condominium', 'BLK SomeWhere', 'DD
    Development Ltd', 'GEO EXPLORATION', null, 'for retaining wall design review',
    '28-oct-03'));
end;

```

List 8-1 PL/SQL procedure for inserting project table

Borehole information

The borehole information of the 11 boreholes was entered into *holeinfo_table* using the PL/SQL procedure shown in List 8-2. Each borehole contained a reference to the project it originally belonged to. In this list the first 4 boreholes (BH-1~4) were inserted first and referred to the project identified by the *contractor* column which was “*SI Ltd*”. The other 7 boreholes (K-1~7) were inserted subsequently where the project reference pointed to the project identified by contractor “*ASIA Drilling Ltd*”. The project reference points, and only points, to the original borehole, as one


```

declare
  p ref project_type; /*declare a variable to hold project_type information*/
begin
  select ref(a) into p from project_table a where contractor='SI Ltd';
  insert into holeinfo_table values (
    holeinfo_type('BH-1','Rotary',3000,3000,104.18,33.28,'29-mar-01','Lee',
    null,null,null,null,null,null, '2-apr-01',null, 'A.B.Driller',null,null,
    null,null,'Stable', null,null,null,p) );
  insert into holeinfo_table values (
    holeinfo_type('BH-2','Rotary',3045.2,3029,104.04,33.1,'23-mar-01','Lee',
    null,null,null,null,null,null,null, '27-mar-01',null,
    'A.B.Driller',null,null,null,null,'Stable', null,null,null,p) );
  insert into holeinfo_table values (
    holeinfo_type('BH-3','Rotary',3066.3,2998.2,104.20,36.15,'28-mar-
    01','Lee',null,null,null,null,null,null,null, '29-mar-01',null,
    'A.B.Driller',null,null,null,null,'Stable', null,null,null,p) );
  insert into holeinfo_table values (
    holeinfo_type('BH-4','Rotary',3104.7,3025.7,104.87,30.3,'23-mar-01','Lee',
    null,null,null,null,null,null,null, '26-mar-01',null,
    'A.B.Driller',null,null,null,null,'Stable', null,null,null,p) );
  select ref(a) into p from project_table a where contractor='ASIA Drilling Ltd';
  insert into holeinfo_table values (
    holeinfo_type('K-1','Rotary',3025.3,3000,103.535,33.95,'7-jan-03',null,
    null,null,null,null,null,null,null, '11-jan-03',null,'SAF',null,null,null,'Stable',
    null,null,null,p) );
  insert into holeinfo_table values (
    holeinfo_type('K-2','Rotary',3024.5,3022,102.981,27.84,'7-jan-03',null,
    null,null,null,null,null,null,null, '9-jan-03',null, 'SAF',null,null,
    null,null,'Stable', null,null,null,p) );
  insert into holeinfo_table values (
    holeinfo_type('K-3','Rotary',3016.0,3025.0,103.951,15.95,'8-jan-03',null,
    null,null,null,null,null,null,null, '9-jan-03',null, 'SAF',null,null, null,null,'Stable',
    null,null,null,p) );
  insert into holeinfo_table values (
    holeinfo_type('K-4','Rotary',3030.8,3032.3,102.57,27.43,'10-jan-03',null,
    null,null,null,null,null,null,null, '11-jan-03',null,'SAF',null,null,null,'Stable',
    null,null,null,p) );
  insert into holeinfo_table values (
    holeinfo_type('K-5','Rotary',3076.8,3031.9,104.854,31.08,'8-jan-03',null,
    null,null,null,null,null,null,null, '9-jan-03',null, 'SAF',null,null, null,null,'Stable',
    null,null,null,p) );
  insert into holeinfo_table values (
    holeinfo_type('K-6','Rotary',3083.6,2998.8,104.249,28.78,'8-jan-03',null,
    null,null,null,null,null,null,null, '9-jan-03',null,'SAF',null,null,null,'Stable',
    null,null,null,p) );
  insert into holeinfo_table values (
    holeinfo_type('K-7','Rotary',3035.0,2994.5,103.165,28.2,'8-jan-03',null,
    null,null,null,null,null,null,null, '9-jan-03',null, 'SAF',null,null, null,null,'Stable',
    null,null,null,p));
end;

```

List 8-2 PL/SQL procedure for inserting borehole information table

borehole may be reused by various projects.

Project borehole relation

A borehole drilled for one project could be used subsequently for another project. The relationship between boreholes and project were built up in the *project_hole_table*. List 8-3 shows the list used to define the relationship of a project identified by contractor “GEO EXPLORATION” with 11 boreholes in the *project_hole_table*. Four of the 11 boreholes (BH-1~4) were originally drilled from a project conducted by “SI Ltd” and the other 7 boreholes were from a project conducted by “ASIA Drilling Ltd”, these boreholes were reused in the project conducted by “GEO EXPLORATION”.

```

declare
  p ref project_type;
  h1 ref holeinfo_type; /* declare 11 holeinfo_type variables */
  h2 ref holeinfo_type;
  h3 ref holeinfo_type;
  h4 ref holeinfo_type;
  h5 ref holeinfo_type;
  h6 ref holeinfo_type;
  h7 ref holeinfo_type;
  h8 ref holeinfo_type;
  h9 ref holeinfo_type;
  h10 ref holeinfo_type;
  h11 ref holeinfo_type;
begin
  select ref(a) into p from project_table a where contractor='GEO EXPLORATION';
  select ref(b) into h1 from holeinfo_table b where holeid='BH-1';
  select ref(b) into h2 from holeinfo_table b where holeid='BH-2';
  select ref(b) into h3 from holeinfo_table b where holeid='BH-3';
  select ref(b) into h4 from holeinfo_table b where holeid='BH-4';
  select ref(b) into h5 from holeinfo_table b where holeid='K-1';
  select ref(b) into h6 from holeinfo_table b where holeid='K-2';
  select ref(b) into h7 from holeinfo_table b where holeid='K-3';
  select ref(b) into h8 from holeinfo_table b where holeid='K-4';
  select ref(b) into h9 from holeinfo_table b where holeid='K-5';
  select ref(b) into h10 from holeinfo_table b where holeid='K-6';
  select ref(b) into h11 from holeinfo_table b where holeid='K-7';
  insert into project_hole_table values(
    p,holeinfo_ntb_type(h1,h2,h3,h4,h5,h6,h7,h8,h9,h10,h11) );
end;
```

List 8-3 PL/SQL procedure for inserting project borehole relation table

SPT data

SPT data were entered into *ispt_table*. List 8-4 gives an example of inputting the SPT data of BH-1 into *ispt_table*. Altogether 12 SPT values extracted from the log of BH-1 were inserted into *ispt_table* with a reference linked to BH-1.

```

declare
    h ref holeinfo_type;
begin
    select ref(a) into h from holeinfo_table a where holeid='BH-1';
    insert into ispt_table values (
        ispt_type(2.5,null,null,450,4,'4/30',null,null,null,null,null,null,null,
        null,null,null,null,null,null,null,null,h));
    insert into ispt_table values (
        ispt_type(3.5,null,null,450,3,'3/30',null,null,null,null,null,null,null,
        null,null,null,null,null,null,null,null,h));
    insert into ispt_table values (
        ispt_type(5.0,null,null,450,3,'3/30',null,null,null,null,null,null,null,
        null,null,null,null,null,null,null,null,h));
    insert into ispt_table values (
        ispt_type(9.5,null,null,450,10,'10/30',null,null,null,null,null,null,null,
        null,null,null,null,null,null,null,null,h));
    insert into ispt_table values (
        ispt_type(12.5,null,null,450,12,'12/30',null,null,null,null,null,null,null,
        null,null,null,null,null,null,null,null,h));
    insert into ispt_table values (
        ispt_type(15,null,null,450,15,'15/30',null,null,null,null,null,null,null,
        null,null,null,null,null,null,h));
    insert into ispt_table values (
        ispt_type(18,null,null,450,31,'31/30',null,null,null,null,null,null,null,
        null,null,null,null,null,null,h));
    insert into ispt_table values (
        ispt_type(21.0,null,null,450,48,'48/30',null,null,null,null,null,null,null,
        null,null,null,null,null,null,null,null,h));
    insert into ispt_table values (
        ispt_type(24.0,null,null,450,53,'53/30',null,null,null,null,null,null,null,
        null,null,null,null,null,null,null,null,h));
    insert into ispt_table values (
        ispt_type(27.0,null,null,450,100,'100/30',null,null,null,null,null,null,null,
        null,null,null,null,null,null,h));
    insert into ispt_table values (
        ispt_type(30.0,null,null,300,100,'100/30',null,null,null,null,null,null,null,
        null,null,null,null,null,null,h));
    insert into ispt_table values (
        ispt_type(33.0,null,null,280,100,'100/28',null,null,null,null,null,null,null,
        null,null,null,null,null,null,h));
    commit;
end;

```

List 8-4 PL/SQL procedure for inserting SPT table

Source data

The relevant source of data was stored in the *source_table*. List 8-5 is the list of procedure to insert data source into *source_table*. In the example the information about engineer name, project name and date were “GeoStudy”, “DemoAnalysis” and “3-oct-03” respectively. If any information were bounded with this source, the information should be created by engineer “GeoStudy” for the “DemoAnalysis” project in 3, October 2003.

```
insert into source_table values
(source_type('GeoStudy', 'DemoAnalysis', '3-oct-03'));
```

List 8-5 PL/SQL procedure of inserting source table

Sample data

List 8-6 shows an example for importing the test sample information into *sample_table*. The sample information was stored together with the reference to the borehole from which the sample was collected. In the example, six samples from borehole K-6 were inserted into the *sample_table*.

```
declare
  h ref holeinfo_type;
begin
  select ref(a) into h from holeinfo_table a where holeid='K-6';
  insert into sample_table values
    (sample_type(3.0,1,'UDS',100,3.80,null,null,null,null,null,null,null,null,h)
    );
  insert into sample_table values
    (sample_type(4.0,2,'UDS',100,4.80,null,null,null,null,null,null,null,null,h)
    );
  insert into sample_table values
    (sample_type(5.0,3,'UDS',100,5.85,null,null,null,null,null,null,null,null,h)
    );
  insert into sample_table values
    (sample_type(6.0,1,'UDP',100,6.85,null,null,null,null,null,null,null,null,h)
    );
  insert into sample_table values
    (sample_type(7.0,2,'UDP',100,7.85,null,null,null,null,null,null,null,null,h)
    );
  insert into sample_table values
    (sample_type(9.5,4,'UDS',100,10.0,null,null,null,null,null,null,null,null,h)
    );
commit;
end;
```

List 8-6 PL/SQL procedure for inserting sample table

Classification test

The soil classification test results were inserted into the *clss_table*. List 8-7 shows an example of inserting 7 classification test results of BH-1 into the *clss_table*. All the classification test results were related to a test sample. In the example, the test sample was identified using the relevant borehole reference and the depth to the top of sample.

```

declare
    h ref holeinfo_type;
    s ref sample_type;
begin
    select ref(a) into h from holeinfo_table a where holeid='BH-1';
    select ref(b) into s from sample_table b where holeinfo=h and depthtotop=2.0;
    insert into clss_table values (
        clss_type(null,null,'UD',null,null,81.8,null,null,1.43,null,null,null,null,
        null,null,null,null,null,null,s)
    );
    select ref(b) into s from sample_table b where holeinfo=h and depthtotop=3.0;
    insert into clss_table values (
        clss_type(null,null,'UD',null,null,153.4,135,107,1.22,null,null,null,null,
        null,null,null,null,null,null,s)
    );
    select ref(b) into s from sample_table b where holeinfo=h and depthtotop=4.5;
    insert into clss_table values (
        clss_type(null,null,'UD',null,null,130.6,null,null,1.22,null,null,null,null,
        null,null,null,null,null,null,s)
    );
    select ref(b) into s from sample_table b where holeinfo=h and depthtotop=6.0;
    insert into clss_table values (
        clss_type(null,null,'UD',1,null,100.7,null,null,1.47,null,null,null,null,null,null,
        null,null,null,s)
    );
    select ref(b) into s from sample_table b where holeinfo=h and depthtotop=6.0;
    insert into clss_table values (
        clss_type(null,null,'UD',2,null,82.2,null,null,1.52,null,null,null,null,null,null,
        null,null,null,s)
    );
    select ref(b) into s from sample_table b where holeinfo=h and depthtotop=9.00;
    insert into clss_table values (
        clss_type(null,null,'UD',null,null,27.0,null,null,1.96,null,null,null,null,null,null,
        null,null,null,s)
    );
    select ref(b) into s from sample_table b where holeinfo=h and depthtotop=12.0;
    insert into clss_table values (
        clss_type(null,null,'UD',null,null,24.0,null,null,1.94,null,null,null,null,
        null,null,null,null,null,null,s)
    );
end;
```

List 8-7 PL/SQL procedure for inserting classification table

Stratum data

List 8-8 gives an example for inputting stratum data into stratum_table. In this example, 12 strata of BH-1 were inserted into the stratum table. These strata were exactly identical to the strata presented in the BH-1 borehole log shown in Appendix E. Each stratum was bounded with appropriate borehole reference.

```

declare
  h ref holeinfo_type;
begin
  select ref(a) into h from holeinfo_table a where holeid='BH-1';
  insert into stratum_table values (
    stratum_type(0.0,1.3,'FILL-Soft Sandy Clay, Dark Brown','Fill', h)
  );
  insert into stratum_table values (
    stratum_type(1.3,2.0,'FILL-Firm Silty Clay, Reddish Pink/Light grey','Fill', h)
  );
  insert into stratum_table values (
    stratum_type(2.0,6.3,'Soft Organic Peaty Clay with Wood, black','Peaty Clay', h)
  );
  insert into stratum_table values (
    stratum_type(6.3,7.9,'Soft Marine Clay, Greenish grey','Marine clay', h)
  );
  insert into stratum_table values (
    stratum_type(7.9,10.4,'Stiff Silty Clay, Reddish Brown/Light grey','Silty clay', h)
  );
  insert into stratum_table values (
    stratum_type(10.4,14.0,'Stiff Sandy Silt, Light grey','Silt', h)
  );
  insert into stratum_table values (
    stratum_type(14.0,16.7,'Stiff Sandy Clay, Yellowish Brown','Sandy Clay', h)
  );
  insert into stratum_table values (
    stratum_type(16.7, 19.4,'Very Stiff Clayey Silt with Sand, Greyish Yellow','Silt', h)
  );
  insert into stratum_table values (
    stratum_type(19.4,22.5,'Hard clayey Silt, Greyish Yellow','Silt', h)
  );
  insert into stratum_table values (
    stratum_type(22.5,25.7,'Hard Clayey Silt, Greyish Yellow','Silt', h)
  );
  insert into stratum_table values (
    stratum_type(25.7,32.3,'Hard Silty Clay, Yellowish Brown','Silty Clay', h));
  insert into stratum_table values (
    stratum_type(32.3,33.28,'Hard Clayey Silt, Greyish Yellow','Silt', h));
end;

```

List 8-8 PL/SQL procedure for inserting stratum table

It should be noted that the stratum data were treated as an independent unit and were not constrained by the stratigraphy schema in the proposed database model. Instead of using the stratum defined in the BH-1 borehole log, it was also possible to combine several strata as one single stratum for simplification. This would be very difficult to achieve with the traditional geotechnical database model currently in existence in which the soil properties and strata were constrained by soil stratigraphy schema.

Connection of strata

The connection data were a collection of strata from different boreholes having a common relationship. They belonged to a new data type defined in the proposed database model and would be used to define the soil surface data and soil layer data. The connection data were stored in the *connect_table*. List 8-9 shows an example which defines connection data comprising a collection of strata from different boreholes. In this example, 22 strata from the 11 boreholes were selected and grouped as a connection named as “Fill”. For BH-1, BH-2, BH-3, K-4, K-5 and K-6, each borehole had 2 strata selected; BH-4, K-2 and K-3 each had 1 stratum selected; K-1 had 3 strata and K-7 had 4 strata selected. The relationships between the strata in this connection were defined as *map*. Following the strata relationship defined in the proposed database model as described in Chapter 5, it meant that all the highest stratum elevations from each borehole would share the soil surface and all the lowest stratum elevations for each borehole would share another soil surface. A reference to the source of the connection data was also stored in this table to enable tracing back to the origin of this connection data. In this example, the source of the connection data could be traced following the project name *DemoAnalysis* from the *source_table*.

Soil surface data

The surface was defined using connections together with a type attribute. An example of inputting surface data is given in List 8-10. In this example two surfaces were inputted into the *surface_table*. The two surfaces were named as *Fill_t* and *Fill_b* respectively. The connection data used to define the two surfaces was “Fill”.

```

declare
    s ref source_type;
    h ref holeinfo_type;
    l1 ref stratum_type; l2 ref stratum_type; l3 ref stratum_type; l4 ref stratum_type;
    l5 ref stratum_type; l6 ref stratum_type; l7 ref stratum_type; l8 ref stratum_type;
    l9 ref stratum_type; l10 ref stratum_type; l11 ref stratum_type; l12 ref stratum_type;
    l13 ref stratum_type; l14 ref stratum_type; l15 ref stratum_type; l16 ref stratum_type;
    l17 ref stratum_type; l18 ref stratum_type; l19 ref stratum_type; l20 ref stratum_type;
    l21 ref stratum_type; l22 ref stratum_type;
begin
    select ref(a) into s from source_table a where project='DemoAnalysis';
    select ref(a) into h from holeinfo_table a where holeid='BH-1';
    select ref(a) into l1 from stratum_table a where top=0.0 and holeinfo=h;
    select ref(a) into l2 from stratum_table a where top=1.3 and holeinfo=h;
    select ref(a) into h from holeinfo_table a where holeid='BH-2';
    select ref(a) into l3 from stratum_table a where top=0.0 and holeinfo=h;
    select ref(a) into l4 from stratum_table a where top=0.3 and holeinfo=h;
    select ref(a) into h from holeinfo_table a where holeid='BH-3';
    select ref(a) into l5 from stratum_table a where top=0.0 and holeinfo=h;
    select ref(a) into l6 from stratum_table a where top=1.5 and holeinfo=h;
    select ref(a) into h from holeinfo_table a where holeid='BH-4';
    select ref(a) into l7 from stratum_table a where top=0.0 and holeinfo=h;
    select ref(a) into h from holeinfo_table a where holeid='K-1';
    select ref(a) into l8 from stratum_table a where top=0.0 and holeinfo=h;
    select ref(a) into l9 from stratum_table a where top=1.5 and holeinfo=h;
    select ref(a) into l10 from stratum_table a where top=4.5 and holeinfo=h;
    select ref(a) into h from holeinfo_table a where holeid='K-2';
    select ref(a) into l11 from stratum_table a where top=0.0 and holeinfo=h;
    select ref(a) into h from holeinfo_table a where holeid='K-3';
    select ref(a) into l12 from stratum_table a where top=0.0 and holeinfo=h;
    select ref(a) into h from holeinfo_table a where holeid='K-4';
    select ref(a) into l13 from stratum_table a where top=0.0 and holeinfo=h;
    select ref(a) into l14 from stratum_table a where top=0.8 and holeinfo=h;
    select ref(a) into h from holeinfo_table a where holeid='K-5';
    select ref(a) into l15 from stratum_table a where top=0.0 and holeinfo=h;
    select ref(a) into l16 from stratum_table a where top=0.5 and holeinfo=h;
    select ref(a) into h from holeinfo_table a where holeid='K-6';
    select ref(a) into l17 from stratum_table a where top=0.0 and holeinfo=h;
    select ref(a) into l18 from stratum_table a where top=0.5 and holeinfo=h;
    select ref(a) into h from holeinfo_table a where holeid='K-7';
    select ref(a) into l19 from stratum_table a where top=0.0 and holeinfo=h;
    select ref(a) into l20 from stratum_table a where top=0.5 and holeinfo=h;
    select ref(a) into l21 from stratum_table a where top=3.0 and holeinfo=h;
    select ref(a) into l22 from stratum_table a where top=4.0 and holeinfo=h;
    insert into connect_table values (
        s,'Fill','map', stratum_ntb_type(l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12,l13,
        l14,l15,l16,l17,l18,l19,l20,l21,l22)
    );
    commit;
end;

```

Figure 8-9 PL/SQL procedure for inserting connection table

A source reference was also entered together with the surface data to enable tracing the data source. The type attribute defined whether the top elevation or the bottom elevation of the strata would be used to define the surface. For this case, the surface Fill_t should be defined by the stratum top elevation as the “TOP” attribute was used. On the contrary, the surface Fill_b should be defined by the stratum top elevation as “BOTTOM” attribute was used.

Using the surface data table for each defined surface, a set of elevation data from boreholes could be obtained. Based on the elevation dataset, a grid surface file could be created using the ESRI *Spatial Analyst* Program. The corresponding surface data file was stored outside the database with the same name of surface data and was used for later analysis.

```

declare
    c1 ref connect_type;
    s ref source_type;
begin
    select ref(a) into s from source_table a where project='DemoAnalysis';
    select ref(a) into c1 from connect_table a where name='Fill';
    insert into surface_table values (
        s,'Fill_t','TOP',connect_ntb_type(c1),null
    );
    insert into surface_table values (
        s,'Fill_b','BOTTOM',connect_ntb_type(c1),null
    );
commit;
end;

```

List 8-10 PL/SQL procedure for inserting surface table

Soil layer data

The soil layer data was defined by two soil surfaces and an array of connection data. All the strata in these connection data should belong to the defined soil layer. The soil layer data were stored in the *layer_table*. List-11 is an example of the PL/SQL procedure inserting the soil layer information into the database table. This example defined a soil layer named “Fill”. The top surface of this soil layer was referenced to a soil surface named “Fill_t” in the *surface_table*. The bottom surface of this soil layer was referenced to a soil surface named “Fill_b” in the *surface_table*. The connection data named “Fill” in the *connect_table* was bounded to the soil layer. This meant that all the strata in that connection belonged to the defined soil layer.

```

declare
    c1 ref connect_type;
    s ref source_type;
    q1 ref surface_type;
    q2 ref surface_type;
begin
    select ref(a) into s from source_table a where project='DemoAnalysis';
    select ref(a) into q1 from surface_table a where name='Fill_t';
    select ref(a) into q2 from surface_table a where name='Fill_b';
    select ref(a) into c1 from connect_table a where name='Fill';
    insert into layer_table values (
        s,'Fill',q1,q2,connect_ntb_type(c1), 'Well graded and compacted, Low compressibility
        material', null
    );
end;
```

List 8-11 PL/SQL procedure for inserting soil layer table

Stratigraphy data

The relationships between different soil layers were defined in an XML stratigraphy document following the schema proposed in this study. The stratigraphy data were stored in the *stratigraphy_table*. List 8-12 gives an example of inserting stratigraphy XML data for this case study. In the example, 6 soil layers were used to categorize the subsoil condition of this site. All the soil layers were in a sibling relationship as defined in Chapter 4. The names of soil layers were “Fill”, “OrganicClay”, “MarineClay”, “F2”, “F1”, and “OASand” sequentially from top to bottom.

```

insert into stratigraphy_table values (
  'DemoAnalysis', 'GeoStudy',
  '<?xml version="1.0"?>
  <!DOCTYPE STRATIGRAPHY [
  <!-- DTD for stratigraphy-->
  <!ELEMENT STRATIGRAPHY (LAYER+)>
  <!ELEMENT LAYER ANY>
  <!ATTLIST LAYER
    name ID #REQUIRED
    topsurface CDATA #IMPLIED
    bottomsurface CDATA #IMPLIED
    description CDATA #IMPLIED
  >
  ]>
  <STRATIGRAPH>
    <LAYER name="Fill" topsurface="fill_t" bottomsurface="fill_b"
    description="Well graded, well compacted, low compressibility material"/>
    <LAYER name="OrganicClay" topsurface="organ_t" bottomsurface="organ_b"
    description="Low density, large compressibility and low plasticity soil"/>
    <LAYER name="MarineClay" topsurface="mc_t" bottomsurface="mc_b"
    description="Low density, Large compressibility and high plasticity soil"/>
    <LAYER name="F2" topsurface="f2_t" bottomsurface="f2_b" description="Sandy
    Clay"/>
    <LAYER name="F1" topsurface="f1_t" bottomsurface="f1_b" description="Clayey
    Sand"/>
    <LAYER name="OASand" topsurface="oasand_t" description="Weathered zone,
    residual soil"/>
  </STRATIGRAPH>'
);

```

List 8-12 PL/SQL procedure for inserting stratigraphy table

Through the PL/SQL procedures, the borehole test data and stratigraphy data were inputted into the database of *CrystalGround*. Through the process of defining soil surface data, soil layer data and stratigraphy data, the subsurface domain was categorized geometrically and these data were completely stored inside the database by implementing the proposed hybrid database model.

8.2.2 Graphical Presentation of Soil Profiles

Once the relevant stratigraphy information had been defined and imported into the database, they could be retrieved and organized to cope with the requirements of different applications. In this study, *CrystalGround* was used to create various vertical sectional soil profiles across the project site.

No post-process tool was developed in the demonstration application for graphical output, editing and improvement. The extracted soil profile data were stored as ESRI shape files and hence could be easily accessed for further processing using existing software packages. In this study, the ArcGIS 9.1 was used to present the cross-section profiles.

A plan layout of three longitudinal sections, Section 1, Section 2 and Section 3, are shown in Figure 8-2. In this figure, Section 1 crossed 5 boreholes (BH-1, K-1, K-7, BH-3, and K-6) and Section 3 crossed 6 boreholes (K-3, K-2, K-4, BH-2, K-5 and BH-4). Section 2 passed through the centre of the site. The longitudinal variation of the soil layers for the site could be described through these sections. Figure 8-3 shows three vertical sections along the 3 routes shown in Figure 8-2. All the cross-sectional soil profiles indicated the existence of soft marine clays towards the western end of the site. The strata indicated in the 11 boreholes were overlaid onto Section 1 and Section 3 in Figure 8-3 to check the consistency of the interpreted soil layers.

Using the section profile data produced from the demonstration application, various 3D views and fence diagrams would be created. With the help of these 3D graphical representations, the existence of specific soil layers could be quickly visualized. One example of the 3-D representations created for this project is shown in Figure 8-4.

Figure 8-4 shows a 3D diagram together with the interpreted soft marine clay layer. The 11 boreholes are superimposed in this diagram. The existence of the soft

Figure 8-2 Plan of boreholes and sections

Figure 8-3 Soil longitudinal section 1-3

Figure 8-4 A 3D diagram showing the existence of soft Marine Clay

marine clay layer can be identified in this 3D diagram labeled “mc”. This layer appeared as a lense in the study domain. The thickest part of the soft clay layer was located at the western end of the site and its thickness gradually reduced towards the East. The actual failure was initiated at the location where the thickest soft marine clay was found. The most critical part of the existing retaining wall that needed to be carefully evaluated was the section of the wall fringing the western end. Through such a 3D diagram, the spatial relationships of the 11 boreholes with the soft marine clay were clearly identified. It could be used to explain why the 4 boreholes in the initial site investigation did not reveal the extent of the marine clay layer. Meanwhile, Figure 8-4 also shows that the interpolated 3D diagram was consistent with the borehole strata.

Once the 3D stratigraphy data had been created and stored in the database, soil sections along any route, including a curved route, could be generated readily. The route could represent the alignment of a pipeline or a tunnel. Although the scale of the project site was rather small for a tunnelling project, the principle of the potential application remains. Figure 8-5 shows the plan view of the alignment of a facility across the site. Eight sections (S1-S8) perpendicular to the alignment are indicated in the plan. The vertical cross sectional soil profiles along S1 to S8 are presented in Figure 8-6. In certain high stake project such as tunneling project, the stratigraphy information is critical in deciding the trunnel alignment, the construction methods and monitoring plan.

Graphical outputs depicting subsurface conditions could provide a more effective means of communication between different parties involved in a project. These parties could include the client, contractor, consultants and regulatory authorities. With the help of these three-dimensional graphical representations, engineers can gain a more vivid visualization of the soil conditions below the ground. Special features and high risk regions can be readily identified. Even parties without the requisite technical background could be better briefed on the prevailing geotechnical conditions. Decision and response can be made more effectively. More

Figure 8-5 Plan of assumed facility and sections

Figure 8-6(a) The cross sections S1-S3

Figure 8-6(b) The cross sections S4-S6

Figure 8-6(c) The cross sections S7-S8

critically to the engineer, the derived information can be stored, updated and reused through the proposed database models.

8.3 Summary

This chapter presents a case study based on borehole data derived from an actual project. The process of data input has been illustrated in detail and all the borehole data involved are included in the appendix E to enable the readers to repeat the process. Through the successful implementation of the process, it can be concluded that the proposed database model is able to handle the geotechnical data involved in a real project. The workability of the proposed database model is thus established.

The XML stratigraphy schema proposed in the database model is capable of handling the internal relationships of soil layers and enabling storage of the interpreted information in a database. By bounding the surfaces of soil layer to the GIS raster data model, the underground domain is categorized in a 3D frame. As demonstrated in the case study, cross sections could be generated not only along a straight line but also along a polyline. Two advantages have been achieved as shown in this case. The first advantage is that the stratigraphy data can be stored in the database along with the subjective characteristics since judgment is involved in the creation of the stratigraphy model. The second advantage is that the 3D spatial referencing can be conducted using the basic GIS functions and stratigraphy data by the application without any human intervention.

Chapter 9 Conclusions and Recommendations

9.1 Conclusions

Geotechnical data and information are routinely collected, processed and transferred across organizations and departments. The information are also transmitted in different formats for analysis and presentation. Geotechnical data are important not only to the original project for which the data was originally acquired, they are also important as a valuable source of historical data for future projects. Database management technology and GIS technology are two key technologies which play major roles in the storage, transfer and visualization of geotechnical data; particularly in revealing the complex 3D characteristic of the subsurface stratigraphy.

In this study, a hybrid geotechnical database model has been developed to manage geotechnical data with a view to providing integration with 3D GIS applications. The proof of concept has been established by two case studies using a demonstration application developed following the principles proposed in this study. The conclusions derived from this study are summarized as follow:

9.1.1 Geotechnical Database Model

The proposed geotechnical database model differs from the conventional database models in some major aspects. These features of the proposed database model confer significant advantages in the management, storage and transfer of geotechnical information.

1) Change of World View

As distinct from the conventional geotechnical database models, the proposed database model inherits the worldview of the NADM model with

emphasis on how definitions are related to real-world entities. The proposed database model views the world as a possible worlds rather than a known world. The fundamental change of world view is a distinct feature of the proposed geotechnical database model, it focuses on the existence of different interpolation results based on the same entities. Compared to conventional geotechnical database models, the proposed geotechnical database model provides a high level abstraction. The database model is especially suitable for the development of an archival database where data are obtained from various sources.

2) Stratigraphy Centralization

The proposed hybrid geotechnical database model decouples the geotechnical test data from stratigraphy schema. Hence, it enables the geotechnical test data to be fitted into various stratigraphy schemas to suit the specific needs of each project and the learned and subjective judgement of the engineer.

The proposed database model uses an XML schema for subsurface stratigraphy data description. Using the XML support from the database industry, the complex relationships interpreted for different subsurface condition can be easily stored in an XML database and retrieved for reuse. Instead of breaking the stratigraphy information into different segments for the purpose of storage without any associated geotechnical meanings, an XML database can manage stratigraphy data as an integrated unit. Taking advantage of XML document processing tools, the information in an XML document can be searched, sorted, and modified efficiently.

The subsurface stratigraphy model proposed in this study enables spatial inference to be conducted using the relationships embedded in the stratigraphy document. With the help of the XML tools, the layer information can be traced back and forth along a tree structure, the

relationships between soil layers can be readily reproduced, and detailed soil layer references can be quickly established. The spatial inference that is normally done conceptually by a human agent can now be processed automatically by a computer system. The automation of this step greatly simplifies the decision-making process especially for the later users. The end-users are relieved from having to duplicate the complex subsurface modelling process and can thus devote more attention to more critical issues.

3) Variogram Data Management

Estimation or interpolation of soil parameters is a routine and critical aspect of geotechnical engineering. Hence, the ability to store the geostatistical information and estimation methodologies will greatly enhance the functionality of the database model. One common methodology of geostatistical estimation is kriging using the variogram. It is realized that the variogram modelling process is a complex process and may deter the general users from tapping into the potential power of geostatistical analysis. This situation can be partially relieved by embedding the variogram description in a database model, and free the end-users from the tedious modelling work. The benefit of building a variography repository had been recognized by earlier researchers. However, due to the constraints of technology available at that time, such ideas were difficult to implement. Those limitations inhibited the wide-spread use of geostatistical tools.

With the advance in technologies, this requirement can now be realized. In this study, an XML schema for a variogram model is proposed. By taking advantage of an XML database, the variogram for soil properties can be efficiently stored and reused. Similar to the information for stratigraphy, the search and modification of variogram parameters can be quickly conducted due to the extensive availability of XML tools. The new geotechnical database model represents one of the very few attempts to manage variography data in a database, if not the first.

4) Hybrid Database

The proposed geotechnical database model is developed as a hybrid database model inheriting the advantage of existing developments. It combines the rational worldview of NADM model, the rigorous data structure of the AGS format, and the flexibility of the XML schema. The development of the proposed database model incorporates various aspects of GIS concepts, XML standard and geotechnical methodology.

9.1.2 Proof of Concepts

A demonstration application incorporating the concepts described in this report (termed *CrystalGround*) has been developed as proof of concept. Two case studies have been conducted, one based on a synthetic dataset and another based on real borehole data obtained from an actual project. Through these case studies, the following conclusions are drawn:

1) Applicability of New Geotechnical Database Model

CrystalGround was implemented using the proposed concept in an Oracle 8i database. It was used to manage the geotechnical borehole data, laboratory test and in-situ test data, stratum data, surface data, soil layer data and stratigraphy data and other data involved in the two case studies. The results from the two case studies affirmed the applicability of the new database model.

2) 3D GIS Support

With the geotechnical test data decoupled from the stratigraphy schema, the spatial relationship between a data point and the stratigraphy schema is provided by spatial references. The spatial reference for a data point and the soil surfaces of the stratigraphy schema is implemented through basic 3D GIS functions. As demonstrated in the two case studies, the proposed geotechnical database model is developed for 3D GIS support by linking the

soil subsurface data to GIS raster data,

With the new features of the proposed database model, the gaps that currently exist in the data analysis and distribution process could be narrowed or overcome through the proposed geotechnical database model, thus fulfilling one of the key objectives of this study. The solution provided in this study gives geotechnical engineers a handy tool not only for sharing geotechnical data but also, and most importantly, sharing their knowledge and experiences with others.

9.2 Recommendations for Further Study

A hybrid geotechnical database model has been proposed in this study. However, it is believed the database model can be further enhanced and improved through further work in the following areas:

1) Artificial Intelligence (AI) for stratigraphy

In this case study, a set of PL/SQL procedures were used to manually input the stratigraphy into the geotechnical database. This study agrees with the conclusion drawn by Toll (1998) that such stratigraphy process can be greatly helped with an AI system. As demonstrated in the GeoSEP database (Suwanwiwattana et al., 2001), it is believed that with the help of the AI system, the stratigraphy can be derived from borehole data automatically.

2) Variogram study on geotechnical data

With the proposed database model, it is possible to build up a variography database for geotechnical data. Study on collecting variogram information and elaborating the variogram database is highly recommended.

3) Interpolation Methods for Soil Surfaces

It is not sufficient that the soil surface models derived from a set of elevation

data are mathematically correct, they must also be geotechnically meaningful and consistent. In this study the surface interpolation was conducted using the spline function of *ESRI Raster Analyst*. It is recognized that the interpolation results may not be optimal, although the effect is not central to the objective of this study. It is recommended that a detailed study on interpolation methods for soil surfaces should be carried to provide rational interpolation guidelines based on good geotechnical principles.

4) Integration with GeotechML

This geotechnical database model used the XML schema for stratigraphy data and variography data. Further study should be carried out to integrate the proposed XML schema with the existing GeotechML as the later is a platform dedicated to geotechnical applications.

5) Pre-processing and Post-processing Tool for Geotechnical Data Input

Due to the characteristic of this research study and the limited time, no pre-processing tool such as user interface has been developed in this study. The geotechnical data for the case studies were imported into the geotechnical database through a set of PL/SQL procedures. The data input process would be greatly enhanced if some pre-processing tools were developed. Thus, tools with graphical user interface to enable the user to interactively define the stratigraphy schema are highly desirable.

References:

- Adams, T. M. (1993) "Knowledge representation and processing in relational data base", Journal of Computing in Civil Engineering, Vol. 7, No. 2, pp.238-255.
- Adams, T. M., Tang, Y. S. and Wiegand, N. (1993) "Spatial data models for managing subsurface data", Journal of Computing in Civil Engineering, Vol. 7, No. 3, pp. 260-277.
- Ali, M. M., Nowatzki, E. A. and Myers, D. E. (1989) "Geostatistical methods to predict collapsing soils", Proceedings of XII ICSMFE/CIMSTF, Rio de Janeiro, pp. 567-570.
- Ansary, M. A., Siddiquee, M. S. A. and Mannan, R. (1999) "Artificial neural networks for the development of interactive database of soil strength parameters", Civil and Environmental Engineering Conference, New Frontiers and Challenges, Bangkok, Thailand.
- Association of Geotechnical and Geoenvironmental Specialists (1999) "Electronic transfer of geotechnical and geoenvironmental data", Third Edition.
- Association of Geotechnical and Geoenvironmental Specialists (2005) "Electronic transfer of geotechnical and geoenvironmental data using XML data format", Third Edition.
- Benson, C. H. and Rashad, S. M. (1996) "Enhanced subsurface characterization for prediction of contaminant transport using co-kriging", Geostatistics for Enviromental and Geotechnical Applications, ASTM STP 1283, American Society for Testing and Materials.
- Bekele, A. (2001) "Spatial variability and isotopic studies of the Prairie-Forest transition soil in Louisiana" *Phd thesis*, Louisiana State University, US.
- Bisland, R. B. (1989) Database management: Developing Application System Using Oracle, Prentice-Hall International.
- Bonham-Carter, G. F. (1996) Geographic information systems for geoscientists: Modelling with GIS. Pergamon, UK.
- Brodaric, B. and Gahegan, M. (2000) "Geoscience map data models, open systems GIS and semantics", GeoCanada2000.
- Camp, C. V. and Brown, M. C. (1993) "GIS procedure for developing three-dimensional subsurface profile", Journal of Computing in Civil Engineering, Vol. 7, No. 3, pp. 296-309.

References

- Cambridge Energy Research Associate (1996) Quiet revolution: Information technology and the reshaping of the oil and gas business.
- Caronna, S. (2006) Implementing XML for geotechnical Databases, *Geo-Engineering Data: Representation and Standardisation*, University of Nottingham, UK.
- CGDMWG (2000) Survey of Geoscience Agency Systems and Data Models, Project report, Holonics Data Management Group Ltd, Canada.
- Cheng, M. Y., Ko, C. H. and Chang, C. H. (2002) Computer-aided DSS for safety monitoring of geotechnical construction, Automation in Construction, Vol.11, pp. 375-390.
- Chiasson, P., Lafleur, J., Soulie, M. and Law, K. T. (1995) "Characterizing spatial variability of a clay by geostatistics", Can. Geotech. J., No. 32, pp. 1-10.
- Clark, P. J. and Prebharan, N. (1987) "Marina Bay station, Singapore excavation in Soft Clay", Fifth international Geotechnical Seminar: Case History in Soft Clay, Nanyang Technological Institute, Singapore.
- Cliff, A. D. and Ord, J. K. (1981) Spatial Processes: Models and Applications, London: Pion.
- Codd, E.F. (1970). "A Relational Model of Data for Large Shared Data Banks". Communications of the ACM Vol 13 , No. 6. pp.377-387.
- DeGroot, D. J. and Baecher, G. B. (1993) "Estimating autocovariance of in-situ soil properties", Journal of Geotechnical Engineering, Vol. 119, No. 1, pp. 147-166.
- Deutsch, C. V. and Journel, A. G. (1998) GSLIB geostatistical software library and user's guide, Oxford University Press, USA.
- ESRI (1997) Understanding GIS: The Arc/Info method, ESRI Press, Redland.
- Fenton, G. A. (1999) "Estimation for stochastic soil models", Journal of Geotechnical and Geoenvironmental Engineering, ASCE, Vol. 125, No. 6, pp. 470-485.
- Fenton, G. A. (1999) "Random field modelling of CPT data", Journal of Geotechnical and Geoenvironmental Engineering, ASCE, Vol. 125, No. 6, pp. 486-498.
- Garcia-Molina, H., Ullma, J. D. and Widom J. (2002) Database Systems: the complete book, Prentice Hall, New Jersey.
- Giles, D. (1994) "A Geographic Information System for Geotechnical and Ground Investigation Data Management and Analysis", Fifth European Conference and

References

- Exhibition of GIS, Vol. I, pp. 776-778.
- Goovaerts, P. (1997) Geostatistics for natural resource evaluation, Oxford University Press.
- Graves, M. (2002) Designing XML databases, Prentice Hall PTR., NJ.
- Hamel, J. V. (1998) "Geotechnical cross-sections re-visited", Geotechnical site characterization, Balkema, Rotterdam, pp.159-164.
- Hastings, J. and Brodaric, B. (2001) "Evolution of an object-oriented, NADM-based data model prototype for the USGS national geology map database project", Proceeding of International Association of Mathematical Geology Annual Meeting, Cancun, Mexico.
- Hellawell, E. E., Lamont-Black, J. Kemp, A. C. and Hughes, S. J. (2001) "GIS as a tool in geotechnical engineering", Proceedings of the Institution of Civil Engineers Geotechnical Engineering, 149 Issue2, pp. 85-93.
- Honjo, Y. and Kuroda, K. (1991) "A new look at fluctuating geotechnical data for reliability design", Soils and Foundations, Vol. 31, No. 1, pp. 110-120.
- Houlding, S. W. (1994) 3D Geoscience Modelling, Springer-Verlag, HongKong.
- Hutchison, R. J. and Chandler, R. J. (1999) "Sharing geotechnical data-The future today", Field Measurements in Geomechanics, Balkema, Rotterdam, pp.109-112.
- Isaaks, E. H. and Srivastava, R. M. (1989) An Introduction to Applied Geostatistics, Oxford University Press, USA.
- Jonathan, R. (2000) Multidimensional Geographic Information Science, Taylor & Francis, London.
- Johnson, B. R., Brodaric, B., Raines, G. L., Hastings, J. T. and Wahl, R. (1999) Digital Geologic Map Data Model, <http://geology.usgs.gov/dm/>
- Jones, N. J. and Wright, S. G. (1993) "Subsurface characterization with solid models", Journal of Geotechnical Engineering, ASCE, Vol. 119, No. 11, pp. 1823-1839.
- Journel, A. G. (1989) Fundamentals of Geostatistics in Five Lessons, American Geophysical Union, Washington D. C.
- Kimmance, J. P. (1999) "Geographical information system application to construction and geotechnical data management on large projects", Field Measurements in Geomechanics, Belkema, Rotterdam, pp.151-156.

References

- Kifer, M., Bernstein, A. and Lewis, P. M. (2005) Database System: An Application-Oriented Approach, Second Edition, Addison Wesley, New York.
- Lee, F. H., Tan, T. S., Karunaratne, G. P. and Lee, S. L. (1990) "Geotechnical data management system", Journal of Computing in Civil Engineering, Vol. 4, No. 3, pp.239-254.
- Li, X. X. and Hayashi, S. (1999) "Using geotechnical database for modelling spatial variability of soil properties", Geotechnical Engineering Journal, Vol. 30, No. 1, pp. 41-57.
- Luna, R., Hertel, T., Baker, H. and Fennessey, T. (2001) "Geotechnical database for emergency vehicle access routes in Missouri", Proceedings of the 80th annual meeting of the Transportation Research Board, Washington, D. C.
- Marble, D. F. and Peuquet, D. J. (1983) "Geographic Information system in remote sensing", Manual of Remote Sensing, Vol. 1, 2nd edition, VA: American Society of Photogrammetry, pp.923-957.
- Matheron, G. (1970) The theory of regionalized variables and its applications. Les Cahiers du Centre de Morphologie Mathematique de Fontainebleau, Ecole National Supérieure des Mines de Paris, No.5.
- McBrayer, M. C., Wiley, K. E. and Gilbert, R. B. (1998) "SimSite: Computer model to simulate site investigation for groundwater", Geotechnical Site Characterization, Balkema, Rotterdam, pp.189-194.
- McPhail, J. D. (2001) Electronic Storage and interchange of geotechnical data, <http://www.ejge.com/GML/>
- Moustafa, M. M. and Yomota, A. (1998) "Spatial modelling of soil properties for subsurface drainage projects", Journal of Irrigation and Drainage Engineering, Vol. 124, No. 4, pp.218-228.
- Muller, R. J. (1999) Database Design for Smarties: Using UML for Data Modelling, Morgan Kaufmann Publishers.
- Naiburg, E. J. & Maksimchuk, R. A. (2001) UML for Database Design, Addison-Wesley.
- Nathanail, C. P. and Rosenbaum, M. S. (1998) "Spatial management of geotechnical data for site selection", Engineering Geology, Vol. 50, pp.347-356.
- Nicholls, R. A., Pycroft, A. S. and Palmer, M. J. (1996) "From paper to silicon

References

- chip-the growing art of computerised data management”, Advances in Site Investigation Practice, Thomas Telford, London, pp.186-197.
- Nigbor, R. L., Swift, J. N., Bardet, J. P., and Hu, J. (2001) “The ROSRINE geotechnical engineering database and website, Invited Workshop on Archiving and Web Dissemination of Geotechnical Data, Cosmos publication No. cp-2001/3., pp. 41-58.
- Oliver, A. J. and Toll, D. G. (1996) “A computer system for site investigation data management and interpretation”, Advances in Site Investigation Practice, Thomas Telford, London.
- Oliver, M. A. and Webster, R. (1990) “Kriging: a method of interpolation for geographical information system”, International Journal of Geographical Information System, Vol. 4(3).
- Oloufa, A. A., Eltahan, A. A. and Papacostas, C. S. (1994) “Integrated GIS for construction site investigation”, Journal of Computing in Civil Engineering, pp. 211-222.
- Oloufa, A. A., Papacostas, C. S. and Espino, R. (1992) “Construction applications of relational data bases in three-dimensional GIS”, Journal of Computing in Civil Engineering, Vol. 6. No. 1, pp.72-84.
- O’Rourke, T. D. and Pease, J. W. (1997) “Mapping liquefiable layer thickness for seismic hazard assessment”, Journal of Geotechnical and Geoenvironmental Engineering, Vol. 123, No. 1, pp.46-56.
- Ozmutlu, S. and Hack, R. (2001) “Subsurface engineering problem and 3D modelling system: applications and requirement”, European Science Foundation Conference, Spa, Belgium.
- Page-Jones, M. (2000) Fundamentals of Object-Oriented Design in UML, Dorset House Publishing, New York.
- Parsons, R. L. and Frost, J. D. (2000) “Interactive analysis of spatial subsurface data using GIS-based tool”, Journal of Computing in Civil Engineering, Vol. 14, No. 4, pp. 215-222.
- Ponti, D. J. (2001) “USGS geologic site database and data acquisition for stratigraphical model development”, Invited Workshop on Archiving and Web Dissemination of Geotechnical Data, Cosmos publication No. cp-2001/3., pp. 73-92.
- Porsvig, M., Christensen, F. M. and Hansen, A. (1989) “Great Belt information system with a 3-D model”, Proceedings of the 12th International Conference on Soil Mechanics and Foundation Engineering, Rio de Janeiro, pp.475-478.

References

- Przewłócki, J. (2000) "Two-dimensional random field of mechanical soil properties", Journal of Geotechnical and Geoenvironmental Engineering, Vol. 126, No. 4, pp. 373-377.
- Real, C. R. and McGuire, T. (2001) "A geotechnical database to support seismic hazard mapping", Invited Workshop on Archiving and Web Dissemination of Geotechnical Data, Cosmos publication No. cp-2001/3., pp. 5-20.
- Satyanarayana, R. and Benoit, J. (2001) "World Wide Web User Query Interface for the National Geotechnical Experimentation Sites Database", Invited Workshop on Archiving and Web Dissemination of Geotechnical Data, Cosmos publication No. cp-2001/3., pp. 123-150
- SIAR(2002) SembCorp Industries Annual Report,
<http://www.sembcorp.com.sg/cgibin/annualreports/annualreports.pl?action=annualreport>
- Simons, B., Boisvert, E., Brodaric, B., Cox, S. Duffy, T. R., Johnson, B. R., Laxton, J. L. and Richard, S. (2006) GeosciML: Enabling the Exchange of Geological Map Data, Australian Earth Sciences Convention, Melbourne.
- Simsion, G. C. and Witt, G. C. (2005) Data Modelling Essentials, Elsevier Inc., San Francisco, CA.
- Steve, M. (2000) Building Oracle XML Applications, O'Reilly & Associates, Sebastopol.
- Stirling, R. M (2002) "Data management in the 21st century-Emerging technologies and their implication for hydrography", The International Federation of Surveyors (FIG) XXII International Congress, Washington, D. C. USA.
- Soulie, M., Montes, P. and Silvestri, V. (1990) "Modelling spatial variability of soil parameters", Can. Geotech. J., No. 27, pp. 617-630.
- Suwanwiwattana, P., Chantawarangul, K., Mairaing, W. and Apaphant, P. (2001) "The development of geotechnical database of Bangkok subsoil using GRASS-GIS", The 22nd Asian Conference on Remote Sensing, Singapore.
- Terzaghi, K., Peck, R. B., and Mesri, G. (1996) Soil Mechanics in Engineering Practice, 3rd edition, John Woley & Son, Inc., New York.
- Thalheim, B. (2000) Entity-Relationship Modelling: Foundation of Database Technology, Springer, Berlin.
- Toll, D. G. (1995) "The role of a knowledge-based system in interpreting geotechnical information", Geotechnique, Vol. 45, No. 3, pp.525-531.

References

- Toll, D. G. (1998) "A review of artificial intelligence systems for site characterization", Geotechnical Site Characterization, Balkema, Rotterdam, pp.327-332.
- Winkler, C., Bosquet, F., Cavin, X., and Paul, J. (1999) "Design and implementation of an immersive geoscience toolkit" Proceedings Visualization'99, San Francisco, California. pp.429-432.
- Wither, A., Pontin, S. and Black, M. (2000) "The geotechnical database for the Jubilee Line Extension project", Proc. Instn. Civ. Engrs. Geotech. Engng, 143, pp. 131-138.
- Yu, M. W., Park, H. D. and Song, W. D. (1999) "Application of borehole data management technique into site investigation" Proceedings of International Congress On Rock Mechanics, Vol. 2, Paris, pp.1315-1317.
- Zeiler, M. (2000) Modelling Our World: The ESRI Guide to Geodatabase Design, ERSI Press.
- Zlatanova, S. (2000) 3D GIS for Urban development, PhD Thesis, Graz University of Technology, Austria.

Appendix A An Example of DTD for Geotechnical Borehole at University of Durham (<http://www.dur.ac.uk/~des0www4/geotechml/>)

```

<?xml version="1.0"?>
<!DOCTYPE GEOTECHNICAL [

  <!ELEMENT GEOTECHNICAL_PROJECT (IDENTIFIER+, NAME+, DATE+,
    GEOTECH_ML_VERSION+,
    PROJECT_DETAILS+, SITE_INVESTIGATION, STRUCTURES, CONSTRUCTION,
    MONITORING)>
  <!ELEMENT IDENTIFIER (PCDATA?)>
  <!ELEMENT NAME (PCDATA?)>
  <!ELEMENT DATE (PCDATA?)>
  <!ELEMENT GEOTECH_ML_VERSION (PCDATA?)>

  <!ELEMENT PROJECT_DETAILS (PROJECT_LOCATION?, PROJECT_CLIENT?,
    CONTRACTOR?, ENGINEER?,
    PROJECT_COMMENTS?, PROJECT_DATE?)>
  <!ELEMENT PROJECT_LOCATION (PCDATA?)>
  <!ELEMENT PROJECT_CITY (PCDATA?)>
  <!ELEMENT PROJECT_COUNTY (PCDATA?)>
  <!ELEMENT PROJECT_COUNTRY (PCDATA?)>
  <!ELEMENT GRID_REFERENCE_EAST (PCDATA?)>
  <!ELEMENT GRID_REFERENCE_NORTH (PCDATA?)>
  <!ELEMENT GRID_REFERENCE_LETTER (PCDATA?)>
  <!ELEMENT CLIENT (PCDATA?)>
  <!ELEMENT CONTRACTOR (PCDATA?)>
  <!ELEMENT ENGINEER (PCDATA?)>
  <!ELEMENT START_DATE (PCDATA?)>
  <!ELEMENT END_DATE (PCDATA?)>
  <!ELEMENT PROJECT_COMMENTS (PCDATA?)>

  <!ELEMENT SITE_INVESTIGATION (IDENTIFIER+, NAME+, DATE+, PROJECT_DETAILS,
    DESK_STUDY, SITE_RECONNAISSANCE, GROUND_INVESTIGATION,
    FIELD_MONITORING)>
  <!ELEMENT DESK_STUDY (IDENTIFIER+, NAME+, DATE+, PROJECT_DETAILS)>
  <!ELEMENT SITE_RECONNAISSANCE (IDENTIFIER+, NAME+, DATE+,
    PROJECT_DETAILS)>
  <!ELEMENT GROUND_INVESTIGATION (IDENTIFIER+, NAME+, DATE+,
    PROJECT_DETAILS, BOREHOLE, TRIALPIT)>
  <!ELEMENT FIELD_MONITORING (IDENTIFIER+, NAME+, DATE+, PROJECT_DETAILS)>

  <!ELEMENT BOREHOLE (IDENTIFIER+, HOLE_DETAILS+, GROUND_CONDITIONS+,
    GROUND_DETAILS*,
    INSITU_TESTS*, SAMPLES*, LABORATORY_TESTS*, WATER_STRIKE*,
    PIEZOMETER*)>
  <!ELEMENT TRIALPIT (IDENTIFIER+, HOLE_DETAILS+, GROUND_CONDITIONS+,
    GROUND_DETAILS*,
    INSITU_TESTS*, SAMPLES*, LABORATORY_TESTS*, WATER_STRIKE*,
    PIEZOMETER*)>

```

Appendix A

An Example of DTD for Geotechnical Borehole

```

<!ELEMENT HOLE_DETAILS (TYPE?, FINAL_DEPTH?, LOGGED_BY?, CREW?, PLANT?,
COMMENTS?,
GRID_REFERENCE_LETTER?, GRID_REFERENCE_EAST?, GRID_REFERENCE_NORTH?,
GROUND_LEVEL?,
LOCAL_GRID_X?, LOCAL_GRID_Y?, LOCAL_GRID_Z?, LOCATION_IN_PROJECT?,
CLUSTER?,
LOCATION_METHOD?, START_DATE?, END_DATE?, BACKFILL_DATE?,
((DIAMETER*, CASING_DIAMETER*, ORIENTATION?, INCLINATION?, PROGRESS*,
CHISELLING*,
FLUSH*, DEPTH_RELATED_COMMENTS*)
| (LENGTH?, WIDTH?, TRAVERSE_END_EAST?, TRAVERSE_END_NORTH?,
TRAVERSE_END_LEVEL?,
SHORING?, STABILITY?)))>
<!ELEMENT GROUND_LEVEL (PCDATA?)>
<!ELEMENT TRAVERSE_END_EAST (PCDATA?)>
<!ELEMENT TRAVERSE_END_NORTH (PCDATA?)>
<!ELEMENT TRAVERSE_END_LEVEL (PCDATA?)>
<!ELEMENT LOCAL_GRID_X (PCDATA?)>
<!ELEMENT LOCAL_GRID_Y (PCDATA?)>
<!ELEMENT LOCAL_GRID_Z (PCDATA?)>
<!ELEMENT LOCATION_IN_PROJECT (PCDATA?)>
<!ELEMENT LOCATION_METHOD (PCDATA?)>
<!ELEMENT CLUSTER (PCDATA?)>
<!ELEMENT TYPE (PCDATA?)>
<!ELEMENT FINAL_DEPTH (PCDATA?)>
<!ELEMENT ORIENTATION (PCDATA?)>
<!ELEMENT INCLINATION (PCDATA?)>
<!ELEMENT LENGTH (PCDATA?)>
<!ELEMENT WIDTH (PCDATA?)>
<!ELEMENT BACKFILL_DATE (PCDATA?)>
<!ELEMENT LOGGED_BY (PCDATA?)>
<!ELEMENT CREW (PCDATA?)>
<!ELEMENT COMMENTS (PCDATA?)>
<!ELEMENT DIAMETER (PCDATA?)>
<!ELEMENT CASING_DIAMETER (PCDATA?)>
<!ELEMENT PROGRESS (PCDATA?)>
<!ELEMENT CHISELLING (PCDATA?)>
<!ELEMENT DEPTH_RELATED_COMMENTS (PCDATA?)>
<!ELEMENT FLUSH (PCDATA?)>
<!ELEMENT PLANT (PCDATA?)>
<!ELEMENT SHORING (PCDATA?)>
<!ELEMENT STABILITY (PCDATA?)>

<!ELEMENT GROUND_CONDITIONS (LAYER+, GROUND_DETAIL*, CORE*,
DISCONTINUITIES*, FRACTURES*,
WEATHERING*)>

<!ELEMENT LAYER (TOP+, BASE+, NAME?, DESCRIPTION?, GEOLOGICAL_ORIGIN?,
MATERIAL?, LEGEND?)>

<!ELEMENT TOP (PCDATA?)>
<!ELEMENT BASE (PCDATA?)>
<!ELEMENT DESCRIPTION (PCDATA?)>
<!ELEMENT GEOLOGICAL_ORIGIN (PCDATA?)>
<!ELEMENT MATERIAL (NAME+, TYPE?, MOISTURE_CONDITION?, CONSISTENCY?,
PLASTICITY?,

```



```

CONSTITUENT*, STRUCTURE*)>
<!ELEMENT MOISTURE_CONDITION (PCDATA?)>
<!ELEMENT CONSISTENCY (PCDATA?)>
<!ELEMENT PLASTICITY (PCDATA?)>
<!ELEMENT CONSTITUENT (TYPE+, AMOUNT?, MAIN_COLOUR?,
SECONDARY_COLOUR?, COLOUR_MODIFIER?,
GRAIN_SIZE?, SHAPE?, TEXTURE?)>
<!ELEMENT STRUCTURE (TYPE+, SPACING?, DIP?, ORIENTATION, SURFACE)>
<!ELEMENT LEGEND (PCDATA?)>

<!ELEMENT GROUND_DETAIL (TOP+, BASE+, DESCRIPTION?)>

<!ELEMENT CORE (PCDATA?)>
<!ELEMENT DISCONTINUITIES (PCDATA?)>
<!ELEMENT FRACTURES (PCDATA?)>

<!ELEMENT WEATHERING (TOP+, BASE+, WEATHERING_GRADE?, COMMENTS?)>
<!ELEMENT WEATHERING_GRADE (PCDATA?)>

<!ELEMENT INSITU_TESTS (CONE_PENETRATION_TEST*,
DYNAMIC_PROFILING_TEST*,
HORIZONTAL_PROFILING_GAUGE*, INSITU_CBR*, INSITU_DENSITY*,
SINGLE_POINT_INSTRUMENT*,
INSITU_PERMEABILITY_TEST*, INSITU_REDOX_TEST*, INSITU_RESISTIVITY_TEST*,
INSITU_VANE_TEST*,
PROFILING_TEST*, PRESSUREMETER_TEST*, PUMPING_TEST*,
STANDARD_PENETRATION_TEST*)>

<!ELEMENT CONE_PENETRATION_TEST (PCDATA?)>
<!ELEMENT DYNAMIC_PROFILING_TEST (PCDATA?)>
<!ELEMENT HORIZONTAL_PROFILING_GAUGE (PCDATA?,
HORIZONTAL_PROFILE_GAUGE_DATA*)>
<!ELEMENT INSITU_CBR (PCDATA?)>
<!ELEMENT INSITU_DENSITY (PCDATA?)>
<!ELEMENT SINGLE_POINT_INSTRUMENT (PCDATA?,
SINGLE_POINT_INSTRUMENT_DATA*)>
<!ELEMENT INSITU_PERMEABILITY_TEST (PCDATA?)>
<!ELEMENT INSITU_REDOX_TEST (PCDATA?)>
<!ELEMENT INSITU_RESISTIVITY_TEST (PCDATA?)>
<!ELEMENT INSITU_VANE_TEST (PCDATA?)>
<!ELEMENT PROFILING_TEST (PCDATA?, PROFILING_DATA*)>
<!ELEMENT PRESSUREMETER_TEST (PCDATA?, PRESSUREMETER_DATA*,
PRESSUREMETER_TEST_LOOPS_DATA*)>
<!ELEMENT PUMPING_TEST (PCDATA?)>

<!ELEMENT STANDARD_PENETRATION_TEST (TOP+, TYPE?, N_VALUE?,
SEATING_BLOWS?, MAIN_TEST_BLOWS?,
TOTAL_PENETRATION?, REPORTED_RESULT?, BLOWS_INCREMENT_1?,
BLOWS_INCREMENT_2?,
BLOWS_INCREMENT_3?, BLOWS_INCREMENT_4?, BLOWS_INCREMENT_5?,
BLOWS_INCREMENT_6?,
PENETRATION_INCREMENT_1?,
PENETRATION_INCREMENT_2?, PENETRATION_INCREMENT_3?,
PENETRATION_INCREMENT_4?, PENETRATION_INCREMENT_5?, PENETRATION_INCRE
MENT_6?,
CASING_DEPTH?, WATER_DEPTH?, COMMENTS?)>

```

```

<!ELEMENT N_VALUE (PCDATA?)>
<!ELEMENT SEATING_BLOWS (PCDATA?)>
<!ELEMENT MAIN_TEST_BLOWS (PCDATA?)>
<!ELEMENT TOTAL_PENETRATION (PCDATA?)>
<!ELEMENT REPORTED_RESULT (PCDATA?)>
<!ELEMENT BLOWS_INCREMENT_1 (PCDATA?)>
<!ELEMENT BLOWS_INCREMENT_2 (PCDATA?)>
<!ELEMENT BLOWS_INCREMENT_3 (PCDATA?)>
<!ELEMENT BLOWS_INCREMENT_4 (PCDATA?)>
<!ELEMENT BLOWS_INCREMENT_5 (PCDATA?)>
<!ELEMENT BLOWS_INCREMENT_6 (PCDATA?)>
<!ELEMENT PENETRATION_INCREMENT_1 (PCDATA?)>
<!ELEMENT PENETRATION_INCREMENT_2 (PCDATA?)>
<!ELEMENT PENETRATION_INCREMENT_3 (PCDATA?)>
<!ELEMENT PENETRATION_INCREMENT_4 (PCDATA?)>
<!ELEMENT PENETRATION_INCREMENT_5 (PCDATA?)>
<!ELEMENT PENETRATION_INCREMENT_6 (PCDATA?)>
<!ELEMENT CASING_DEPTH (PCDATA?)>
<!ELEMENT WATER_DEPTH (PCDATA?)>

<!ELEMENT DYNAMIC_PROFILING_DATA (PCDATA?,
DYNAMIC_PROFILING_DATA_DATA*)>
<!ELEMENT HORIZONTAL_PROFILE_GAUGE_DATA (PCDATA?)>
<!ELEMENT SINGLE_POINT_INSTRUMENT_DATA (PCDATA?)>
<!ELEMENT PROFILING_DATA (PCDATA?)>
<!ELEMENT PRESSUREMETER_DATA (PCDATA?)>
<!ELEMENT PRESSUREMETER_TEST_LOOPS_DATA (PCDATA?)>

<!ELEMENT SAMPLES (SAMPLE+)>

<!ELEMENT SAMPLE (TOP+, IDENTIFIER+, TYPE+, SAMPLE_DETAILS+)>

<!ELEMENT SAMPLE_DETAILS (DIAMETER?, BASE?, DESCRIPTION?, BLOW_COUNT?,
COMMENTS?, DATE?, TIME?,
BAROMETRIC_GAS_PRESSURE?, WATER_DEPTH?, TEMPERATURE?, GAS_PRESSURE?,
GAS_FLOW?)>
<!ELEMENT BLOW_COUNT (PCDATA?)>

<!ELEMENT TIME (PCDATA?)>
<!ELEMENT BAROMETRIC_GAS_PRESSURE (PCDATA?)>
<!ELEMENT TEMPERATURE (PCDATA?)>
<!ELEMENT GAS_PRESSURE (PCDATA?)>
<!ELEMENT GAS_FLOW (PCDATA?)>

<!ELEMENT LABORATORY_TESTS (CONSOLIDATION_TEST*, COMPACTION_TEST*,
CBR_TEST*, CHALK_TEST*,
CLASSIFICATION*, MOISTURE_CONDITION_VALUE_TEST*, PERMEABILITY_TEST*,
RELATIVE_DENSITY_TEST*,
ROCK*, SHEAR_BOX_TEST*, SUCTION_TEST*, TEN_PERCENT_FINES_TEST*,
TRIAXIAL_TEST*)>

<!ELEMENT CONSOLIDATION_TEST (SAMPLE_IDENTIFIER+, DEPTH+, SPECIMEN?,
PCDATA?, CONSOLIDATION_DATA+)>
<!ELEMENT COMPACTION_TEST (SAMPLE_IDENTIFIER+, DEPTH+, SPECIMEN?,
COMPACTION_DATA+)>
<!ELEMENT CBR_TEST (SAMPLE_IDENTIFIER+, DEPTH+, SPECIMEN?, CBR_DATA+)>

```



```

<!ELEMENT CHALK_TEST (SAMPLE_IDENTIFIER+, DEPTH+, SPECIMEN?, PCDATA?)>
<!ELEMENT CLASSIFICATION (SAMPLE_IDENTIFIER+, DEPTH+, SPECIMEN?,
CLASSIFICATION_PREPARATION?,
NATURAL_WATER_CONTENT?, LIQUID_LIMIT?, PLASTIC_LIMIT?, BULK_DENSITY?,
DRY_DENSITY?,
PARTICLE_DENSITY?, PERCENT_PASSING_425um?, SHRINKAGE_LIMIT?,
LINEAR_SHRINKAGE?,
HAND_VANE_PEAK_STRENGTH?, HAND_VANE_REMOULDED_STRENGTH?,
PENETROMETER_STRENGTH?,
LAB_VANE_PEAK_STRENGTH?, LAB_VANE_REMOULDED_STRENGTH?)>
<!ELEMENT MOISTURE_CONDITION_VALUE_TEST (SAMPLE_IDENTIFIER+, DEPTH+,
SPECIMEN?, MVC_DATA+)>
<!ELEMENT PERMEABILITY_TEST (SAMPLE_IDENTIFIER+, DEPTH+, SPECIMEN?,
PCDATA?)>
<!ELEMENT RELATIVE_DENSITY_TEST (SAMPLE_IDENTIFIER+, DEPTH+, SPECIMEN?,
PCDATA?)>
<!ELEMENT ROCK (SAMPLE_IDENTIFIER+, DEPTH+, SPECIMEN?, PCDATA?)>
<!ELEMENT SHEAR_BOX_TEST (SAMPLE_IDENTIFIER+, DEPTH+, SPECIMEN?,
SHEAR_BOX_DATA*)>
<!ELEMENT SUCTION_TEST (SAMPLE_IDENTIFIER+, DEPTH+, SPECIMEN?, PCDATA?)>
<!ELEMENT TEN_PERCENT_FINES_TEST (SAMPLE_IDENTIFIER+, DEPTH+, SPECIMEN?,
PCDATA?)>

<!ELEMENT TRIAXIAL_TEST (SAMPLE_IDENTIFIER+, DEPTH+, SPECIMEN?, TYPE?,
UNDRAINED_SHEAR_STRENGTH,
COHESION, FRICTION_ANGLE, SAMPLE_CONDITION?, COMMENTS,
TRIAxIAL_DATA*)>
<!ELEMENT TRIAXIAL_DATA (TEST_NUMBER+, DIAMETER, LENGTH,
WATER_CONTENT, BULK_DENSITY, DRY_DENSITY,
CELL_PRESSURE, INITIAL_POREWATER_PRESSURE, FAILURE_DEVIATOR_STRESS,
FAILURE_POREWATER_PRESSURE,
FAILURE_STRAIN, FAILURE_MODE)>

<!ELEMENT CONSOLIDATION_DATA (PCDATA?)>
<!ELEMENT COMPACTION_DATA (PCDATA?)>
<!ELEMENT MVC_DATA (PCDATA?)>
<!ELEMENT SHEAR_BOX_DATA (PCDATA?)>

<!ELEMENT NATURAL_WATER_CONTENT (PCDATA?)>
<!ELEMENT LIQUID_LIMIT (PCDATA?)>
<!ELEMENT PLASTIC_LIMIT (PCDATA?)>
<!ELEMENT BULK_DENSITY (PCDATA?)>
<!ELEMENT DRY_DENSITY (PCDATA?)>
<!ELEMENT PARTICLE_DENSITY (PCDATA?)>
<!ELEMENT PERCENT_PASSING_425um (PCDATA?)>
<!ELEMENT CLASSIFICATION_PREPARATION (PCDATA?)>
<!ELEMENT SHRINKAGE_LIMIT (PCDATA?)>
<!ELEMENT LINEAR_SHRINKAGE (PCDATA?)>
<!ELEMENT HAND_VANE_PEAK_STRENGTH (PCDATA?)>
<!ELEMENT HAND_VANE_REMOULDED_STRENGTH (PCDATA?)>
<!ELEMENT PENETROMETER_STRENGTH (PCDATA?)>
<!ELEMENT LAB_VANE_PEAK_STRENGTH (PCDATA?)>
<!ELEMENT LAB_VANE_REMOULDED_STRENGTH (PCDATA?)>

```


Appendix A

An Example of DTD for Geotechnical Borehole

```
<!ELEMENT PIEZOMETER (TIP_DEPTH+, DATE?, TYPE?, RESPONSE_ZONE_TOP?,  
RESPONSE_ZONE_BASE?, COMMENTS?, PIEZOMETER_DATA*)>  
<!ELEMENT PIEZOMETER_DATA (DATE+, TIME+, DEPTH?, HEAD?, COMMENTS?)>  
<!ELEMENT HEAD (PCDATA?)>  
  
<!ELEMENT WATER_STRIKE (DEPTH?, CASING_DEPTH?, DATE?, TIME?, FLOW_RATE,  
DEPTH_SEALED, WATER_STRIKE_DATA)>  
<!ELEMENT WATER_STRIKE_DATA (DEPTH_AFTER_WAIT?, WAIT_TIME?)>  
<!ELEMENT DEPTH_AFTER_WAIT (PCDATA?)>  
<!ELEMENT WAIT_TIME (PCDATA?)>  
>
```

Appendix B SQL for Building an Oracle 8i Geotechnical Database

List 1: Create holeinfo object and table

```
create or replace type holeinfo_type as object(
    holeid varchar2(15), exploretype varchar2(10), eastofstart number,
    northofstart number, groundleveltostart number, finaldepth number, startdate
    date, loggingperson varchar2(10), remark varchar2(60), eastofend number,
    northofend number, goundleveltoend number, reference varchar2(15), xlocal
    number, ylocal number, zlocal number, enddate date, backfilldate date, driller
    varchar2(20), orientation number, inclination number, plant varchar2(15),
    shoring varchar2(15), stability varchar2(30), length number, width number,
    locationmethod varchar2(10), originproject ref project_type
);
create table holeinfo_table of holeinfo_type;
```

List 2: Create project object and table

```
create or replace type project_type as object (
    title varchar2(35), location varchar2(35), client varchar2(35), contractor
    varchar2(35), engineer varchar2(35), remark varchar2(60), datetime date
);
create table project_table of project_type;
```

List 3: Create spt object and table

```
create type ispt_type as object(
    top number, totalseatblow int, totaltestblow int, totalpen number, sptvalue int,
    report varchar2(40), casingdep number, waterdep number, testtype
    varchar(10), remark varchar2(30), seatblow1 int, seatblow2 int, testblow1 int,
    testblow2 int, testblow3 int, testblow4 int, seatpen1 number, seatpen2 number,
    testpen1 number, testpen2 number, testpen3 number, testpen4 number,
    holeinfo ref holeinfo_type
);
create table ispt_table of ispt_type;
```

List 4: Create water strike object and table

```
create or replace type wstk_type as object(
    depth number, minutes number, casingdepth number, strikedate date,
    striketime date, postdepth number, remark varchar2(40), seal number
);
create table wstk_table of wstk_type;
```

List 5 : Create sample object and table

```
create type sample_type as object(
    depthtotop number, samplerrefnum int, sampletype varchar2(10), diameter
    number, depthtobase number, description varchar2(30), samplerblownum int,
    remark varchar2(30), sampleddate date, time date, barometricpressure number,
    waterdepth number, temperature number, gaspressure number, gasflow
    number, holeinfo ref holeinfo_type
);
Create table sample_table of sample_type;
```

List 6: Create classification object and table

```
create or replace type class_type as object(
    depthtosamplertop number, samplerrefnum int, sampletype varchar2(10),
    specimenrefnum int, specimendep number, moisturecontent number,
    liquidlimit number, plasticlimit number, bulkdensity number, drydensity
    number, partialdensity number, sieve425 number, preparationmethod
    varchar2(15), shrinkagelimit number, linearshrinkage number, peakhandvane
    number, remouldedhandvane number, pocketpenetrometer number,
    peaklabvane number, remouldedlabvane number, sample ref sample_type
);
create table class_table of class_type;
```

List 7: Create triaxial test object and table

```
create or replace type trig_type as object (
    depthtosamplertop number, samplerrefnum int, sampletype varchar2(10),
    specimenrefnum int, specimendep number, testtype varchar2(10), condition
    varchar2(10), remark varchar2(30), cu number, coh number, phi number,
    sample ref sample_type
);
create table trig_table of trig_type;
```


Appendix B**SQL for Building an Oracle8i Geotechnical Database**

List 8: Create project, holeinfo relation table

```
create or replace type holeinfo_ntb_type as table of ref holeinfo_type;
create table project_hole_table (
    project ref project_type scope is project_table,
    holes holeinfo_ntb_type
) nested table holes store as holes_ntb;
```

List 9: Create stratum object and table

```
create type stratum_type as object (
    top number, base number, description varchar2(60), geology varchar2(15),
    holeinfo ref holeinfo_type
);
create table stratum_table of stratum_type;
```

List10: Create source object and table

```
create or replace type source_type as object (
    engineer varchar2(15), project varchar2(20), definetime date
);
create table source_table of source_type;
```

List 11: Create connect object and table

```
create or replace type stratum_ntb_type as table of ref stratum_type;
create or replace type connect_type as object(
    source ref source_type, name varchar2(15), type varchar2(6), strata
    stratum_ntb_type
);
create table connect_table of connect_type
nested table strata store as connect_strata_ntb;
```

List 12: Create surface object and table

```
create or replace type connect_ntb_type as table of ref connect_type;
create or replace type surface_type as object (
    source ref source_type, name varchar2(15), type varchar2(6), connections
    connect_ntb_type, rasterURL varchar2(35)
);
create table surface_table of surface_type
nested table connections store as sur_connect_ntb;
```

Appendix B

SQL for Building an Oracle8i Geotechnical Database

List 13: Create layer object and table

```
create or replace type layer_type as object (
    source ref source_type, name varchar2(15), top ref surface_type, bottom ref
    surface_type, connections connect_ntb_type, description varchar2(80),
    remarks varchar2(6)
);
create table layer_table of layer_type
nested table connections store as lyr_connections_ntb;
```

List 14: Create stratigraphy table

```
create table stratigraphy_table (
    project varchar2(20), engineer varchar2 (20), stratigraphy clob
) lob (stratigraphy) store as strata_xml (storage (initial 80k next 80k pctincrease 0));
```

List 15: Create variogram table

```
create table variogram_table (
    layer ref layer_type, b datatype varchar2(12), variogram clob
) lob (variogram) store as variogram_xml (storage (initial 80k next 80k pctincrease 0));
```

Appendix C Program Codes for *CrystalGround*

The kernel program codes for *CrystalGround* are listed in this appendix. The server side is implemented using an Oracle 8i database. A set of store procedures has been developed using PL/SQL for handling geotechnical data processing.

The client side is developed using VB 6.0. The main form of *CrystalGround* has an ESRI Mapobject 2.0 ActiveX control. This map control is referred as *Map1* in the following list. It is worth to note that the *Estimator* module has a dynamic linked library called *Ktd3D.dll*. The FORTRAN code for building this library is also included for references.

1) PL/SQL Store Procedures:

```

CREATE or REPLACE FUNCTION getCLSS
(sample_ref ref sample_type)
RETURN xmldom.DOMNode
IS
    doc xmldom.DOMDocument;
    main_node xmldom.DomNode;
    node xmldom.DOMNode;
    nodes xmldom.DOMNodeList;
    one_node xmldom.DOMNode;
    user_node xmldom.DOMNode;
    item_node xmldom.DOMNode;
    temp_node xmldom.DOMNode;
    item_text xmldom.DOMText;

    CURSOR classes(sample_ref ref sample_type) is
        SELECT *
        FROM clss_table
        WHERE sample=sample_ref
        ORDER BY depthtosampletop DESC;
    clss classes%ROWTYPE;
    error EXCEPTION;
BEGIN

    main_node:=newCLSS;

    --obtain the CLASSIFICATION node
    nodes:=xslprocessor.selectNodes(main_node,
        'LABORATORY_TESTS/CLASSIFICATION');
    temp_node:=xmldom.item(nodes,0);
    --obtain the LABORATORY_TESTS node
    nodes:=xslprocessor.selectNodes(main_node,
        '//LABORATORY_TESTS');
    node:=xmldom.item(nodes,0);

    OPEN classes(sample_ref);
    LOOP
        fetch clsses INTO clss;
        exit when clsses%NOTFOUND ;

        --initialise node
        one_node:=xmldom.getFirstChild(node);
        item_node:=xmldom.cloneNode(temp_node, true);
        item_node:=xmldom.insertBefore(node, item_node, one_node);

        --add SPECIMEN--
        IF (clss.specimenrefnum IS NOT NULL) THEN
            nodes:=xslprocessor.selectNodes(item_node,
                '//SPECIMEN');
            user_node:=xmldom.item(nodes,0);
            item_text:=xmldom.createTextNode(xmldom.Makedocument(item_node),
                clss.specimenrefnum);

```

Appendix C

Program Code for CrystalGround

```

one_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));
END IF;

--add DEPTH--
IF (clss.depthtosampletop IS NOT NULL) THEN
nodes:=xslprocessor.selectNodes(item_node,
'./DEPTH');
user_node:=xmldom.item(nodes,0);
item_text:=xmldom.createTextNode(xmldom.Makedocument(item_node),
clss.depthtosampletop);
one_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));
dbms_output.put_line(clss.depthtosampletop);
END IF;

--add WATER_CONTENT--
IF ((clss.moisturecontent)IS NOT NULL) THEN
nodes:=xslprocessor.selectNodes(item_node,
'./NATURAL_WATER_CONTENT');
user_node:=xmldom.item(nodes,0);
item_text:=xmldom.createTextNode(xmldom.Makedocument(item_node),
clss.moisturecontent);
one_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));
END IF;

-- add LL--
IF ((clss.liquidlimit)IS NOT NULL) THEN
nodes:=xslprocessor.selectNodes(item_node,
'./LIQUID_LIMIT');
user_node:=xmldom.item(nodes,0);
item_text:=xmldom.createTextNode(xmldom.Makedocument(item_node),
clss.liquidlimit);
one_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));
END IF;

-- add PL--
IF ((clss.plasticlimit)IS NOT NULL) THEN
nodes:=xslprocessor.selectNodes(item_node,
'./PLASTIC_LIMIT');
user_node:=xmldom.item(nodes,0);
item_text:=xmldom.createTextNode(xmldom.Makedocument(item_node),
clss.plasticlimit);
one_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));
END IF;

END LOOP;
CLOSE clss;
--Clear the last CLSS node
temp_node:=xmldom.removeChild(node, temp_node);
RETURN main_node;

END;
```

```

CREATE or REPLACE PROCEDURE getGroundConditions
(hole_ref ref holeinfo_type, doc IN OUT xmldom.DOMDocument)
IS
  node xmldom.DOMNode;
  nodes xmldom.DOMNodeList;
  node_map xmldom.DOMNamedNodeMap;
  one_node xmldom.DOMNode;
  user_node xmldom.DOMNode;
  item_node xmldom.DOMNode;
  temp_node xmldom.DOMNode;
  item_text xmldom.DOMText;
  east number;
  north number;
  gl number;
  dep number;
  CURSOR strata(hole_ref ref holeinfo_type) is
    SELECT * FROM stratum_table
    WHERE holeinfo=hole_ref
    ORDER BY top DESC;
  stratum strata%ROWTYPE;
  error EXCEPTION;
BEGIN
  -- obtain the borehole information
  select eastofstart, northofstart,groundlevelstart,finaldepth
  into east, north, gl,dep
  from holeinfo_table a
  where ref(a)=hole_ref;
  /*****add hole deatail information*****/
  --East--
  nodes:=xslprocessor.selectNodes(xmldom.makeNode(doc),
    '//GRID_REFERENCE_EAST');
  node:=xmldom.item(nodes,0);
  item_text:=xmldom.createTextNode(doc,east);
  node:=xmldom.appendChild(node, xmldom.makenode(item_text));
  --North--
  nodes:=xslprocessor.selectNodes(xmldom.makeNode(doc),
    '//GRID_REFERENCE_NORTH');
  node:=xmldom.item(nodes,0);
  item_text:=xmldom.createTextNode(doc,north);
  node:=xmldom.appendChild(node, xmldom.makenode(item_text));
  --Ground level--
  nodes:=xslprocessor.selectNodes(xmldom.makeNode(doc),
    '//GROUND_LEVEL');
  node:=xmldom.item(nodes,0);
  item_text:=xmldom.createTextNode(doc,gl);
  node:=xmldom.appendChild(node, xmldom.makenode(item_text));
  --Final depth--
  nodes:=xslprocessor.selectNodes(xmldom.makeNode(doc),
    '//FINAL_DEPTH');
  node:=xmldom.item(nodes,0);
  item_text:=xmldom.createTextNode(doc,dep);
  node:=xmldom.appendChild(node, xmldom.makenode(item_text));
  /*****add Ground conditions information*****/
  --obtain the LAYER node
  nodes:=xslprocessor.selectNodes(xmldom.makeNode(doc),
    '//LAYER');

```


Appendix C

Program Code for CrystalGround

```

temp_node:=xmldom.item(nodes,0);
--obtain the GROUND_CONDITIONS node
nodes:=xslprocessor.selectNodes(xmldom.makeNode(doc),
    '//GROUND_CONDITIONS');
node:=xmldom.item(nodes,0);
OPEN strata(hole_ref);
LOOP
    fetch strata INTO stratum;
    exit when strata%NOTFOUND;
    --initialise node
    one_node:=xmldom.getFirstChild(node);
    item_node:=xmldom.cloneNode(temp_node, true);
    item_node:=xmldom.insertBefore(node, item_node, one_node);
    --add top--
    nodes:=xslprocessor.selectNodes(item_node,
        '//TOP');
    user_node:=xmldom.item(nodes,0);
    item_text:=xmldom.createTextNode(doc,stratum.top);
    user_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));
    --add base--
    nodes:=xslprocessor.selectNodes(item_node,
        '//BASE');
    user_node:=xmldom.item(nodes,0);
    item_text:=xmldom.createTextNode(doc,stratum.base);
    user_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));
    --add description--
    nodes:=xslprocessor.selectNodes(item_node,
        '//DESCRIPTION');
    user_node:=xmldom.item(nodes,0);
    item_text:=xmldom.createTextNode(doc,stratum.description);
    user_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));
    --add geology--
    nodes:=xslprocessor.selectNodes(item_node,
        '//GEOLOGICAL_ORIGIN');
    user_node:=xmldom.item(nodes,0);
    item_text:=xmldom.createTextNode(doc,stratum.geology);
    user_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));
END LOOP;
CLOSE strata;
--Clear the last GROUND_CONDITION node
one_node:=xmldom.removeChild(node, temp_node);
END;
```

```

CREATE or REPLACE FUNCTION getHoleRef(prjName VARCHAR2,
    contraName varchar2, hole_ID varchar2)
RETURN ref holeinfo_type
IS
    p_ref ref project_type;
    holes_nt holeinfo_ntb_type;
    hole_ref ref holeinfo_type;
    strID VARCHAR2(30);
BEGIN
    select ref(a) into p_ref
    from project_table a
    where title=prjName and contractor=contraName;
    select holes into holes_nt from project_hole_table
```

Appendix C**Program Code for CrystalGround**

```

where project=p_ref;
for i IN 1..holes_nt.count LOOP
  select holeid into strID from holeinfo_table a
  where ref(a)=holes_nt(i);
  IF strID=hole_ID THEN
    RETURN holes_nt(i);
  exit;
end if;
END LOOP;
END;

CREATE or REPLACE FUNCTION getHoleXML
  (prjName VARCHAR2, contraName VARCHAR2,
   hole_ID VARCHAR2)
RETURN CLOB
IS
  doc xmldom.DOMDocument;
  hole_ref ref holeinfo_type;
  error EXCEPTION;
  c clob;
BEGIN
  --obtain project infomation
  doc:=getProjectXML(prjName, contraName);

  --obtain the hole reference

  hole_ref:=getHoleRef(prjName, contraName, hole_ID);

  -- obtain the borehole information
  getGroundConditions(hole_ref, doc);

  --obtain samples

  getSample(hole_ref, doc);

  -- convert the xml document to CLOB
  DBMS_LOB.createTemporary(c, TRUE);
  xmldom.writeToClob(doc, c);
  xmldom.freeDocument(doc);
  RETURN c;
  -- DBMS_LOB.freeTemporary(c);
EXCEPTION
  WHEN error THEN
    xmldom.freeDocument(doc);
END;

CREATE OR REPLACE FUNCTION getLabTEST
  (sample_ref ref sample_type)
RETURN xmldom.DOMDocumentFragment
IS
  c clob;
  doc xmldom.DOMDOCUMENT;
  main_node xmldom.DomNode;

  node xmldom.DOMNode;

```

Appendix C

Program Code for CrystalGround

```

nodes xmldom.DOMNodeList;
TRIG_node xmldom.DOMNode;
CLSS_node xmldom.DOMNode;
root_node xmldom.DOMNode;
temp_node xmldom.DOMNode;

root_ele xmldom.DOMELEMENT;
buffer varchar2(600);
frag1 xmldom.domdocumentfragment;
frag3 xmldom.domdocumentfragment;
frag2 xmldom.domdocumentfragment;
sheet xslprocessor.stylesheet;
begin

--Build a document

doc:=xmldom.newDomDocument;
main_node:=xmldom.makenode(doc);
root_ele:=xmldom.createElement(doc, 'TESTS');
root_node:=xmldom.appendChild(main_node, xmldom.makenode(root_ele));

root_ele:=xmldom.createElement(doc, 'TRIGS');
TRIG_node:=xmldom.appendChild(root_node, xmldom.makenode(root_ele));
root_ele:=xmldom.createElement(doc, 'CLASSES');
CLSS_node:=xmldom.appendChild(root_node, xmldom.makenode(root_ele));

sheet:=identityXSLT;
--get TRIG
frag1:=xslprocessor.transformnode(gettrig(sample_ref), sheet);
--get clss
frag2:=xslprocessor.transformnode(getclss(sample_ref), sheet);

--add to document
temp_node:=xmldom.appendChild( TRIG_node, xmldom.makenode(frag1));
temp_node:=xmldom.appendChild( CLSS_node, xmldom.makenode(frag2));

--transform the document
sheet:=labTestXSLT;
frag3:=xslprocessor.transformnode(root_node, sheet);
RETURN frag3;
END;

CREATE or REPLACE FUNCTION getProjectXML
(prjName VARCHAR2, contraName VARCHAR2)
RETURN xmldom.DOMDocument
IS

doc xmldom.DOMDocument;
node xmldom.DOMNode;
nodes xmldom.DOMNodeList;
item_text xmldom.DOMText;

strClient VARCHAR2(30);
strLocation VARCHAR2(30);
error EXCEPTION;

```


Appendix C

Program Code for CrystalGround

```

BEGIN
  --initialise the xml document
  doc:=initialise();

  --obtain project infomation

  select client,location into strClient, strLocation
  from project_table
  where title=prjName and contractor=contraName;

  --add the project client:
  nodes:=xslprocessor.selectNodes(xmlDOM.makeNode(doc),
    '/GEOTECHNICAL_PROJECT/PROJECT_DETAILS/PROJECT_CLIENT');
  node:=xmlDOM.item(nodes,0);
  item_text:=xmlDOM.createTextNode(doc,strClient);
  node:=xmlDOM.appendChild(node, xmlDOM.makenode(item_text));

  -- add the project location:
  nodes:=xslprocessor.selectNodes(xmlDOM.makeNode(doc),
    '/GEOTECHNICAL_PROJECT/PROJECT_DETAILS/PROJECT_LOCATION');
  node:=xmlDOM.item(nodes,0);
  item_text:=xmlDOM.createTextNode(doc,strLocation);
  node:=xmlDOM.appendChild(node, xmlDOM.makenode(item_text));

  -- add the project title
  nodes:=xslprocessor.selectNodes(xmlDOM.makeNode(doc),
    '/GEOTECHNICAL_PROJECT/NAME');
  node:=xmlDOM.item(nodes,0);
  item_text:=xmlDOM.createTextNode(doc,prjName);
  node:=xmlDOM.appendChild(node, xmlDOM.makenode(item_text));

  -- add the contractor name
  nodes:=xslprocessor.selectNodes(xmlDOM.makeNode(doc),
    '/GEOTECHNICAL_PROJECT/PROJECT_DETAILS/CONTRACTOR');
  node:=xmlDOM.item(nodes,0);
  item_text:=xmlDOM.createTextNode(doc,contraName);
  node:=xmlDOM.appendChild(node, xmlDOM.makenode(item_text));

  RETURN doc;
  xmlDOM.freeDocument(doc);

EXCEPTION
  WHEN error THEN
    xmlDOM.freeDocument(doc);
END;

CREATE or REPLACE PROCEDURE getSample
(hole_ref ref holeinfo_type, doc IN OUT xmlDOM.DOMDocument)
IS
  node xmlDOM.DOMNode;
  nodes xmlDOM.DOMNodeList;
  node_map xmlDOM.DOMNamedNodeMap;
  one_node xmlDOM.DOMNode;
  user_node xmlDOM.DOMNode;
  item_node xmlDOM.DOMNode;
  temp_node xmlDOM.DOMNode;

```

Appendix C

Program Code for CrystalGround

```

item_text xmldom.DOMText;
frag xmldom.DOMDocumentFragment;
sample_ref ref sample_type;
value number;

CURSOR samples(hole_ref ref holeinfo_type) is
    SELECT *
    FROM sample_table
    WHERE holeinfo=hole_ref
    ORDER BY depthtotop DESC;
sample samples%ROWTYPE;
error EXCEPTION;
BEGIN

    /*****add SAMPLES *****/
    --obtain the sample node
    nodes:=xslprocessor.selectNodes(xmldom.makeNode(doc),
        '//SAMPLE');
    temp_node:=xmldom.item(nodes,0);
    --obtain the SMAPLES node
    nodes:=xslprocessor.selectNodes(xmldom.makeNode(doc),
        '//SAMPLES');
    node:=xmldom.item(nodes,0);

    OPEN samples(hole_ref);
    LOOP
        fetch samples INTO sample;
        exit when samples%NOTFOUND ;

        --initialise node
        one_node:=xmldom.getFirstChild(node);
        item_node:=xmldom.cloneNode(temp_node, true);
        item_node:=xmldom.insertBefore(node, item_node, one_node);

        --add IDENTIFIER--
        nodes:=xslprocessor.selectNodes(item_node,
            '//SAMPLE/IDENTIFIER');
        user_node:=xmldom.item(nodes,0);
        item_text:=xmldom.createTextNode(doc,sample.samplerefernum);
        user_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));

        --add top--
        nodes:=xslprocessor.selectNodes(item_node,
            '//SAMPLE/TOP');
        user_node:=xmldom.item(nodes,0);
        item_text:=xmldom.createTextNode(doc,sample.depthtotop);
        user_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));

        --add TYPE--

        nodes:=xslprocessor.selectNodes(item_node,
            '//SAMPLE/TYPE');
        user_node:=xmldom.item(nodes,0);
        item_text:=xmldom.createTextNode(doc,sample.sampletype);
        user_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));
        -- add base--

```

Appendix C

Program Code for CrystalGround

```

nodes:=xslprocessor.selectNodes(item_node,
'//SAMPLE/BASE');
user_node:=xmldom.item(nodes,0);
item_text:=xmldom.createTextNode(doc,sample.depthtobase);
user_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));

-- obtain the sample reference--
select ref(a) into sample_ref from
sample_table a
where a.depthtotop=sample.depthtotop and a.holeinfo=hole_ref;

--obtain the documentfragment of laboratory_test
frag:=getLabTEST(sample_ref);

--add lab tests--
nodes:=xslprocessor.selectNodes(item_node,
'//SAMPLE/LABORATORY_TESTS');
user_node:=xmldom.item(nodes,0);
one_node:=xmldom.insertBefore(item_node, xmldom.makenode(frag), user_node);
one_node:=xmldom.removeChild(item_node,user_node);
END LOOP;
CLOSE samples;
--Clear the last SAMPLE node
item_node:=xmldom.removeChild(node, temp_node);
END;
```

CREATE or REPLACE PROCEDURE getSPT
(hole_ref ref holeinfo_type, doc IN OUT xmldom.DOMDocument)

IS

```

node xmldom.DOMNode;
nodes xmldom.DOMNodeList;
node_map xmldom.DOMNamedNodeMap;
one_node xmldom.DOMNode;
user_node xmldom.DOMNode;
item_node xmldom.DOMNode;
temp_node xmldom.DOMNode;
item_text xmldom.DOMText;
value number;
```

CURSOR spts(hole_ref ref holeinfo_type) is

```

SELECT * FROM ispt_table
WHERE holeinfo=hole_ref
ORDER BY top DESC;
```

```

spt spts%ROWTYPE;
error EXCEPTION;
```

BEGIN

```

/*****add SPT information*****/
```

```

--obtain the LAYER node
```

```

nodes:=xslprocessor.selectNodes(xmldom.makeNode(doc),
'//STANDARD_PENETRATION_TEST');
```

```

temp_node:=xmldom.item(nodes,0);
```

```

--obtain the GROUND_CONDITIONS node
```

```

nodes:=xslprocessor.selectNodes(xmldom.makeNode(doc),
'//INSITU_TESTS');
```


Appendix C

Program Code for CrystalGround

```

node:=xmldom.item(nodes,0);

OPEN spts(hole_ref);
LOOP
  fetch spts INTO spt;
  exit when spts%NOTFOUND ;
  --initialise node
  one_node:=xmldom.getFirstChild(node);
  item_node:=xmldom.cloneNode(temp_node, true);
  item_node:=xmldom.insertBefore(node, item_node, one_node);

  --add top--
  nodes:=xslprocessor.selectNodes(xmldom.getFirstChild(node),
    '//STANDARD_PENETRATION_TEST/TOP');
  user_node:=xmldom.item(nodes,0);
  item_text:=xmldom.createTextNode(doc,spt.top);
  dbms_output.put_line (spt.top);
  user_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));

  --add N value--
  nodes:=xslprocessor.selectNodes(xmldom.getFirstChild(node),
    '//STANDARD_PENETRATION_TEST/N_VALUE');
  user_node:=xmldom.item(nodes,0);
  item_text:=xmldom.createTextNode(doc,spt.sptvalue);
  user_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));

END LOOP;
CLOSE spts;
--Clear the last SPT node
item_node:=xmldom.removeChild(node, temp_node);
END;

CREATE or REPLACE FUNCTION getTRIG
(sample_ref ref sample_type)
RETURN xmldom.DOMNode
IS
  main_node xmldom.DomNode;
  node xmldom.DOMNode;
  nodes xmldom.DOMNodeList;
  one_node xmldom.DOMNode;
  user_node xmldom.DOMNode;
  item_node xmldom.DOMNode;
  temp_node xmldom.DOMNode;
  item_text xmldom.DOMText;

  CURSOR trigs(sample_ref ref sample_type) is
    SELECT *
      FROM trig_table
     WHERE sample=sample_ref
     ORDER BY depthtosampletop DESC;
  trig trigs%ROWTYPE;
  error EXCEPTION;
BEGIN
  main_node:=newTRIG;

  --obtain the TRIAXIAL_TEST node

```

Appendix C

Program Code for CrystalGround

```

nodes:=xslprocessor.selectNodes(main_node,
    'LABORATORY_TESTS/TRIAXIAL_TEST');
temp_node:=xmldom.item(nodes,0);
--obtain the LABORATORY_TESTS node
nodes:=xslprocessor.selectNodes(main_node,
    '//LABORATORY_TESTS');
node:=xmldom.item(nodes,0);

OPEN trigs(sample_ref);
LOOP
    fetch trigs INTO trig;
    exit when trigs%%NOTFOUND ;

    --initialise node
    one_node:=xmldom.getFirstChild(node);
    item_node:=xmldom.cloneNode(temp_node, true);
    item_node:=xmldom.insertBefore(node, item_node, one_node);

    --add SPECIMEN--
    IF (trig.specimenrefernum IS NOT NULL) THEN
        nodes:=xslprocessor.selectNodes(item_node,
            '//SPECIMEN');
        user_node:=xmldom.item(nodes,0);
        item_text:=xmldom.createTextNode(xmldom.Makedocument(item_node),
            trig.specimenrefernum);
        one_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));
    END IF;

    --add DEPTH--
    IF (trig.depthtosamplertop IS NOT NULL) THEN
        nodes:=xslprocessor.selectNodes(item_node,
            '//DEPTH');
        user_node:=xmldom.item(nodes,0);
        item_text:=xmldom.createTextNode(xmldom.Makedocument(item_node),
            trig.depthtosamplertop);
        one_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));
    END IF;

    --add TYPE--
    IF ((trig.testtype)IS NOT NULL) THEN
        nodes:=xslprocessor.selectNodes(item_node,
            '//TYPE');
        user_node:=xmldom.item(nodes,0);
        item_text:=xmldom.createTextNode(xmldom.Makedocument(item_node),
            trig.testtype);
        one_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));
    END IF;

    -- add COHESION--
    IF ((trig.coh)IS NOT NULL) THEN
        nodes:=xslprocessor.selectNodes(item_node,
            '//COHESION');
        user_node:=xmldom.item(nodes,0);
        item_text:=xmldom.createTextNode(xmldom.Makedocument(item_node),
            trig.coh);
        one_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));
    END IF;

```

Appendix C

Program Code for CrystalGround

```

END IF;

-- add FRICTION_ANGLE--
IF ((trig.phi)IS NOT NULL) THEN
nodes:=xslprocessor.selectNodes(item_node,
'./FRICTION_ANGLE');
user_node:=xmldom.item(nodes,0);
item_text:=xmldom.createTextNode(xmldom.Makedocument(item_node),
trig.phi);
one_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));
END IF;

-- add UNDRAIN_SHEAR_STRENGTH/--
IF ((trig.cu)IS NOT NULL) THEN
nodes:=xslprocessor.selectNodes(item_node,
'./UNDRAIN_SHEAR_STRENGTH');
user_node:=xmldom.item(nodes,0);
item_text:=xmldom.createTextNode(xmldom.Makedocument(item_node),
trig.cu);
one_node:=xmldom.appendChild(user_node, xmldom.makenode(item_text));
END IF;

END LOOP;
CLOSE trigs;
--Clear the last SPT node
temp_node:=xmldom.removeChild(node, temp_node);
RETURN node;

END;

CREATE or REPLACE FUNCTION identityXSLT
RETURN xslprocessor.Stylesheet
IS
    sheet xslprocessor.Stylesheet;
    parser xmlparser.Parser;
    node xmldom.DOMNode;
    xml VARCHAR2(600);
    error EXCEPTION;

BEGIN
    --initialise the xml document
    parser:=xmlparser.newParser;

    xml:='EXCEPTION

```


*Appendix C**Program Code for CrystalGround*

```

    WHEN error THEN
        xmlparser.freeParser(parser);
    END;

CREATE or REPLACE FUNCTION initialise
RETURN xmldom.DOMDocument
IS

    parser xmlparser.Parser;
    doc xmldom.DOMDocument;
    xml VARCHAR2(6000);
    error EXCEPTION;

    BEGIN
        --initialise the xml document
        parser:=xmlparser.newParser;

        xml:='GEOTECHNICAL_PROJECT>
            <NAME></NAME>
            <DATE></DATE>
            <GEOTECHNICAL_ML_VERSION>1.0</GEOTECHNICAL_ML_VERSION>
            <PROJECT_DETAILS>
                <PROJECT_LOCATION/>
                <PROJECT_CLIENT/>
                <CONTRACTOR/>
                <ENGINEER/>
            </PROJECT_DETAILS>
            <SITE_INVESTIGATION>
                <GROUND_INVESTIGATION>
                    <BOREHOLE>
                        <IDENTIFIER/>
                        <HOLE_DETAILS>
                            <TYPE/>
                            <GRID_REFERENCE_EAST/>
                            <GRID_REFERENCE_NORTH/>
                            <GROUND_LEVEL/>
                            <FINAL_DEPTH/>
                            <DTAE_START/>
                        </HOLE_DETAILS>

                    <GROUND_CONDITIONS>
                        <LAYER>
                            <TOP/>
                            <BASE/>
                            <NAME/>
                            <DESCRIPTION/>
                            <GEOLOGICAL_ORIGIN/>
                        </LAYER>
                    </GROUND_CONDITIONS>
                <INSITU_TESTS>
                    <STANDARD_PENETRATION_TEST>
                        <TOP/>
                        <N_VALUE/>
                        <TYPE/>
                    </STANDARD_PENETRATION_TEST>
                </INSITU_TESTS>
            </SITE_INVESTIGATION>
        </GEOTECHNICAL_PROJECT>';

```

Appendix C

Program Code for CrystalGround

```

        <SAMPLES>
        <SAMPLE>
        <IDENTIFIER/>
        <TOP/>
        <BASE/>
        <TYPE/>
        <LABORATORY_TESTS/>
        </SAMPLE>
    </SAMPLES>
    <WATER_STRIKE>
    <DEPTH/>
    </WATER_STRIKE>
</BOREHOLE>
</GROUND_INVESTIGATION>
</SITE_INVESTIGATION>
</GEOTECHNICAL_PROJECT>;
xmlparser.parseBuffer(parser, xml);
doc:=xmlparser.getDocument(parser);
xmlparser.freeParser(parser);
return doc;
EXCEPTION
    WHEN error THEN
        xmlparser.freeParser(parser);
END;

CREATE or REPLACE FUNCTION labTestXSLT
RETURN xslprocessor.Stylesheet
IS
    sheet xslprocessor.Stylesheet;
    parser xmlparser.Parser;
    node xmldom.DOMNode;
    xml VARCHAR2(600);
    error EXCEPTION;

BEGIN
    --initialise the xml document
    parser:=xmlparser.newParser;

    xml:='<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="TESTS"> <LABORATORY_TESTS><xsl:apply-templates/>
    </LABORATORY_TESTS></xsl:template>
    <xsl:template match="//TRIAXIAL_TEST//CLASSIFICATION">
    <xsl:copy-of select="."/>
    </xsl:template></xsl:stylesheet>';

    xmlparser.parseBuffer(parser, xml);
    sheet:= xslprocessor.newStylesheet(
        xmlparser.getDocument(parser), NULL);
    xmlparser.freeParser(parser);
    return sheet;
EXCEPTION
    WHEN error THEN
        xmlparser.freeParser(parser);
END;

```

Appendix C

Program Code for CrystalGround

CREATE or REPLACE FUNCTION newCLSS
RETURN xmldom.DOMNode
IS

```

parser xmlparser.Parser;
node xmldom.DOMNode;
xml VARCHAR2(600);
error EXCEPTION;

```

BEGIN

```

--initialise the xml document
parser:=xmlparser.newParser;

```

```

xml:='<LABORATORY_TESTS>
    <CLASSIFICATION>
        <SPECIMEN/>
        <DEPTH/>
        <NATURAL_WATER_CONTENT/>
        <LIQUID_LIMIT/>
        <PLASTIC_LIMIT/>
        <BULK_DENSITY/>
    </CLASSIFICATION>
</LABORATORY_TESTS>';

```

```

xmlparser.parseBuffer(parser, xml);
node:=xmldom.makeNode(xmlparser.getDocument(parser));
xmlparser.freeParser(parser);
return node;

```

EXCEPTION

```

    WHEN error THEN
        xmlparser.freeParser(parser);

```

END;

CREATE or REPLACE FUNCTION newTRIG
RETURN xmldom.DOMNode
IS

```

parser xmlparser.Parser;
node xmldom.DOMNode;
xml VARCHAR2(500);
error EXCEPTION;

```

BEGIN

```

--initialise the xml document
parser:=xmlparser.newParser;

```

```

xml:='<LABORATORY_TESTS><TRIAXIAL_TEST>
    <SPECIMEN/>
    <DEPTH/>
    <TYPE/>
    <UNDRAIN_SHEAR_STRENGTH/>
    <COHESION/>
    <FRICTION_ANGLE/>
</TRIAXIAL_TEST></LABORATORY_TESTS>';

```

```

xmlparser.parseBuffer(parser, xml);

```


Appendix C

Program Code for CrystalGround

```
node:=xmldom.makeNode(xmlparser.getDocument(parser));
xmlparser.freeParser(parser);
return node;
EXCEPTION
  WHEN error THEN
    xmlparser.freeParser(parser);
END;
```

2) VB Program

Form: FrmMain

```

Option Explicit
Private Enum MapFunction
    MO_FULLEXTENT = 0
    MO_ZOOMIN = 1
    MO_ZOOMOUT = 2
    MO_DRAWPL = 3
    MO_HOIELOG = 4
    MO_CHOOSEPOINTS = 5
    MO_PAN = 6
End Enum

Dim strataXML As New msxml.DOMDocument           'Stratigraphy XML
Dim pts As New MapObjects2.points
Dim rubberL As MapObjects2.line                 'store baseline for section generation
Dim nameProject As String                       'store the name of this project
Dim iProfCount As Integer
Dim iEstCount As Integer
Private m_action As MapFunction

Dim cc() As Double
Dim aa() As Double
Dim ang1() As Double
Dim ang2() As Double
Dim ang3() As Double
Dim anis1() As Double
Dim anis2() As Double
Dim it() As Long
Dim c0 As Double
Dim nst As Long

Private Sub Form_Load()
    LoadPlan

    'Create two storage files in server's "\shape" folder if they don't exist
    NewLineShpFile "plan"                        ' for storage of the input lines
    NewLineShpFile "profile"                     ' for storage of the output profiles
    NewPointShpFile "points"                     ' for storage of the estimation points

    'Initialize the count number
    iProfCount = 0
    iEstCount = 0
    nameProject = "DemoAnalysis"                 'name of project
    Set strataXML = queryStratigraphy(nameProject) 'load the stratigraphy data of this project
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Map1.Layers.Clear
    Set FrmMain = Nothing
    Set pts = Nothing

```

Appendix C**Program Code for CrystalGround**

```

If MsgBox("do you want to clear the shapefiles?", vbYesNo) = vbYes Then
    ClearShapeFile "plan"
    ClearShapeFile "profile"
    ClearShapeFile "points"
End If
End Sub

Public Sub DoCreateSection()
    If rubberL Is Nothing Then
        MsgBox "The postion of section is not defined"
        Exit Sub
    Else
        Set pts = rubberL.Parts.Item(0)
        End If
        Set rubberL = Nothing
        If pts Is Nothing Then
            MsgBox "The position of the section is not defined"
            Exit Sub
        End If
        'now densify this line using a default distance
        DensifyPoints (2#) 'the default parameter value is for the specific demo case
        CreateChainage
    End Sub

Private Sub DensifyPoints(ByVal measure As Double)
    Dim point1 As New MapObjects2.point
    Dim point2 As New MapObjects2.point
    Dim temPt As New MapObjects2.point
    Dim points As New MapObjects2.points
    Dim i As Integer
    Dim j As Integer
    Dim deltaX As Double
    Dim deltaY As Double
    Dim dis As Double
    Dim Count As Integer
    With pts
        For i = 0 To .Count - 2 'check the distance between two neighbor points
            Set point1 = .Item(i)
            Set point2 = .Item(i + 1)
            dis = point1.DistanceTo(point2)
            Count = Fix(dis / measure)
            points.Add point1
            If Count > 0 Then
                deltaX = point2.x - point1.x
                deltaY = point2.y - point1.y
                For j = 0 To Count - 1
                    temPt.x = point1.x + (j + 1) * measure * deltaX / dis
                    temPt.y = point1.y + (j + 1) * measure * deltaY / dis
                    If Not (temPt.DistanceTo(point2) = 0) Then points.Add temPt
                Next j
            End If
        Next i
        'add the last point to the points
        points.Add point2
    End With

```


Appendix C

Program Code for CrystalGround

```

'Store the points before densifying it
StoreLine pts, "Section_" & CStr(iProfCount), "plan"
StoreLine points, "Densify_" & CStr(iProfCount), "plan"
Set pts = points

'cleaning
Set points = Nothing
Set point1 = Nothing
Set point2 = Nothing
Set temPt = Nothing
End Sub

Public Sub CreateChainage(Optional ByVal scaleX As Double = 0.5, Optional ByVal scaleY As
                        Double = 1#)
    Dim list() As String
    Dim pt1 As New MapObjects2.point
    Dim pt2 As New MapObjects2.point
    Dim points As New MapObjects2.points
    Dim Xmark As Double
    Dim i As Integer
    Dim strInput As String
    Dim line As New MapObjects2.line
    Dim icount As Integer
    Dim gd As MapObjects2.GeoDataset
    Dim db As New MapObjects2.DataConnection
    Dim lyr As New MapObjects2.MapLayer
    Dim File As String
    Dim surfaceFiles() As String 'store surface file name
    Dim eleDom As msxml.IXMLDOMElement
    Dim Count As Integer

    'obtain the names of the surface files
    Count = 0
    For i = 0 To strataXML.getElementsByTagName("LAYER").length - 1
        Set eleDom = strataXML.getElementsByTagName("LAYER").Item(i)
        If (eleDom.Attributes.getNamedItem("bottomsurface") Is Nothing) Then
            Count = Count + 1
            ReDim Preserve surfaceFiles(Count)
            surfaceFiles(Count - 1) = eleDom.Attributes.getNamedItem("topsurface").nodeValue
        Else
            Count = Count + 2
            ReDim Preserve surfaceFiles(Count)
            surfaceFiles(Count - 2) = eleDom.Attributes.getNamedItem("topsurface").nodeValue
            surfaceFiles(Count - 1) = eleDom.Attributes.getNamedItem("bottomsurface").nodeValue
        End If
    Next i
    For i = 0 To UBound(surfaceFiles) - 1
        Next i

    'Map1.Layers.Clear
    'build a string for the elevation query
    strInput = ""
    With pts
        For i = 0 To .Count - 1
            strInput = strInput & CStr(.Item(i).x) & "," & CStr(.Item(i).y)
            If Not (i = .Count - 1) Then strInput = strInput & ","
        Next i
    End With

```

Appendix C

Program Code for CrystalGround

```

Next i
End With

db.Database = App.Path & "\shape"
db.Connect
Set gd = db.FindGeoDataset("profile")
If gd Is Nothing Then MsgBox "the file is not here"
Set lyr.GeoDataset = gd

'Start loop for each surface
For icount = 0 To UBound(surfaceFiles) - 1
    Set points = Nothing
    Erase list
    File = App.Path & "\ascii\" & surfaceFiles(icount)
    Tokenize list, QueryGrid(File, strInput), ","

    With pts
        ' add the points
        Xmark = .Item(0).x
        For i = 0 To .Count - 1
            If Not list(i) = "-9999" Then
                pt1.x = Xmark
                pt1.y = Val(list(i))
                ' using z to record the position in the pts
                pt1.z = i
                points.Add pt1
            End If
            If Not i = .Count - 1 Then Xmark = Xmark + .Item(i).DistanceTo(.Item(i + 1))
        Next i
    End With

    If points.Count > 0 Then
        line.Parts.Add points

        With lyr.Records
            .AddNew
            .Fields("Shape").Value = line
            .Fields("Name").Value = surfaceFiles(icount) & "_" & CStr(iProfCount)
            .Update
            .StopEditing
        End With
    End If
Next icount
lyr.Records.StopEditing
db.Disconnect

LoadProfile (iProfCount)

'clear pts for next usage
Set pts = Nothing
Set lyr = Nothing
Set line = Nothing
Set points = Nothing
Set pt1 = Nothing
Set pt2 = Nothing
Set gd = Nothing

```

Appendix C**Program Code for CrystalGround**

```

    Set db = Nothing
End Sub

Private Sub NewLineShpFile(ByVal name As String)

    Dim db As New MapObjects2.DataConnection
    Dim desc As New MapObjects2.TableDesc

    With desc
        ' define three additional fields
        .FieldCount = 3

        'set the field names
        .FieldName(0) = "Name"
        .FieldName(1) = "scaleX"
        .FieldName(2) = "scaleY"

        ' set the type of field
        .FieldType(0) = moString
        .FieldType(1) = moDouble
        .FieldType(2) = moDouble

        ' set the length of a character field
        .FieldLength(0) = 16

        ' set the number of digits used in the field
        .FieldPrecision(1) = 15
        .FieldPrecision(2) = 15

        ' set the number of digits to the right of the decimal point
        .FieldScale(1) = 3
        .FieldScale(2) = 3
    End With

    db.Database = App.Path & "\shape" ' create the shape file in the shape folder
    db.Connect
    If db.FindGeoDataset(name) Is Nothing Then
        db.AddGeoDataset name, moShapeTypeLine, desc, True
    End If
    db.Disconnect
    Set db = Nothing
    Set desc = Nothing
End Sub

Private Sub DeleteShapeFile(ByVal name As String)
    Dim strFile As String
    strFile = App.Path & "\Shape\" & name
    KillFile strFile & ".shp"
    KillFile strFile & ".shx"
    KillFile strFile & ".dbf"
End Sub

Private Sub ClearShapeFile(ByVal name As String)
    Dim gd As MapObjects2.GeoDataset
    Dim db As New MapObjects2.DataConnection
    Dim lyr As New MapObjects2.MapLayer

```


Appendix C

Program Code for CrystalGround

```

db.Database = App.Path & "\shape"           ' create the shape file in the shape folder
db.Connect
Set gd = db.FindGeoDataset(name)
Set lyr.GeoDataset = gd
With lyr.Records
    .MoveFirst
    Do While Not .EOF
        .Delete
        .MoveNext
    Loop
End With

db.Disconnect
Set lyr = Nothing
Set gd = Nothing
Set db = Nothing
End Sub

Private Sub DoEstimation(ByVal strPoints As String)
    Dim pt As New MapObjects2.point
    Dim pts3D As New MapObjects2.points
    Dim xTem As Double
    Dim yTem As Double
    Dim i As Integer
    Dim list() As String
    Dim dTem As Double
    Dim v As Double

    'tokenize the input points first
    Set pts = Nothing
    Tokenize list, strPoints, ","
    For i = 1 To (UBound(list) + 1) / 2
        xTem = Val(list(2 * (i - 1)))
        yTem = Val(list(2 * (i - 1) + 1))
        Set pt = Map1.ToMapPoint(xTem, yTem)
        pts.Add pt
    Next i

    Erase list
    'transform these points from profile X, Y to real (x,y,z)
    Set pts3D = FromChainageToXYZ("Densify_" & CStr(iProfCount))
    If pts3D Is Nothing Then MsgBox "the transformation is not successful"

    Tokenize list, GetStraInfo(pts3D, "phase1"), ","

    For i = 0 To pts3D.Count - 1           'loop through each point to estimate the vlaue
        v = kriging(pts3D(i), "Phase1 showcase", list(i))       'check the strata information
    Next i
End Sub

Private Sub StoreLine(ByVal points As MapObjects2.points, _
    ByVal lname As String, ByVal fname As String)
    Dim gd As MapObjects2.GeoDataset
    Dim db As New MapObjects2.DataConnection

```

*Appendix C**Program Code for CrystalGround*

```

Dim lyr As New MapObjects2.MapLayer
Dim line As New MapObjects2.line
line.Parts.Add points
db.Database = App.Path & "\shape"
db.Connect
Set gd = db.FindGeoDataset(fname)
If gd Is Nothing Then MsgBox "the file is not here"
Set lyr.GeoDataset = gd

With lyr.Records
    .AddNew
    .Fields("Shape").Value = line
    .Fields("Name").Value = lname
    .Update
End With

db.Disconnect
Set line = Nothing
Set gd = Nothing
Set db = Nothing
Set lyr = Nothing
End Sub

Private Function FromChainageToXYZ(ByVal planNameRef As String) As
MapObjects2.points 'planNameRef is the densified line

Dim gd As MapObjects2.GeoDataset
Dim db As New MapObjects2.DataConnection
Dim lyr As New MapObjects2.MapLayer
Dim line As New MapObjects2.line
Dim ptsRef As New MapObjects2.points
Dim xStart As Double
Dim xDis As Double
Dim xDisRef As Double
Dim pt As New MapObjects2.point
Dim points As New MapObjects2.points
Dim xTem As Double
Dim i As Integer
Dim icount As Integer

db.Database = App.Path & "\shape"
db.Connect
Set gd = db.FindGeoDataset("plan")
If gd Is Nothing Then MsgBox "the plan file is not here"
Set lyr.GeoDataset = gd
With lyr.Records
    .MoveFirst
    Do While Not .Fields("Name") = planNameRef And Not .EOF
        .MoveNext
    Loop
    If Not .EOF Then Set line = .Fields("Shape").Value
End With
If line Is Nothing Then MsgBox "the reference line cannot be found"
Set ptsRef = line.Parts.Item(0)

'cleaning

```

Appendix C

Program Code for CrystalGround

```

Set gd = Nothing
Set db = Nothing
Set lyr = Nothing
Set line = Nothing
'transformation
xStart = ptsRef.Item(0).x
For i = 0 To pts.Count - 1 'loop through each point
    xTem = pts.Item(i).x
    pt.z = pts.Item(i).y
    xDis = xTem - xStart
    xDisRef = 0#
    icount = 0
    With ptsRef
        Do While icount < .Count - 2 And _
            (xDisRef + .Item(icount).DistanceTo(.Item(icount + 1))) < xDis
            icount = icount + 1
            xDisRef = xDisRef + .Item(icount).DistanceTo(.Item(icount + 1))
        Loop
        pt.x = .Item(icount).x + (xDis - xDisRef) * _
            (.Item(icount + 1).x - .Item(icount).x) / .Item(icount).DistanceTo(.Item(icount + 1))

        pt.y = .Item(icount).y + (xDis - xDisRef) * _
            (.Item(icount + 1).y - .Item(icount).y) / .Item(icount).DistanceTo(.Item(icount + 1))
    End With
    points.Add pt
    Set pt = Nothing
Next i

Set FromChainageToXYZ = points
Set points = Nothing
Set ptsRef = Nothing

```

End Function**Private Sub NewPointShpFile(ByVal name As String)**

```

Dim db As New MapObjects2.DataConnection
Dim desc As New MapObjects2.TableDesc
    With desc
        ' define three additional fields
        .FieldCount = 2
        'set the field names
        .FieldName(0) = "Name"
        .FieldName(1) = "stratum"
        ' set the type of field
        .FieldType(0) = moString
        .FieldType(1) = moString
        ' set the length of a character field
        .FieldLength(0) = 16
        .FieldLength(1) = 16
    End With

    db.Database = App.Path & "\shape" ' create the shape file in the shape folder
    db.Connect
    If db.FindGeoDataset(name) Is Nothing Then
        db.AddGeoDataset name, moShapeTypePoint, desc, True
    End If
    db.Disconnect

```


Appendix C

Program Code for CrystalGround

```

Set db = Nothing
Set desc = Nothing
End Sub

Private Function kriging(ByVal point As MapObjects2.point, ByVal prj_name, _
                        ByVal lyr_name As String) As Double
    Dim x() As Double
    Dim y() As Double
    Dim z() As Double
    Dim v() As Double
    Dim est As Double
    Dim estv As Double

    Dim nd As Long
    Dim nx As Long
    Dim ny As Long
    Dim nz As Long

    Dim xmn As Double
    Dim xsiz As Double
    Dim ymn As Double
    Dim ysiz As Double
    Dim zmn As Double
    Dim zsiz As Double

    Dim radius As Double
    Dim radius1 As Double
    Dim radius2 As Double

    Dim xp As Double
    Dim yp As Double
    Dim zp As Double

    Dim sang1 As Double
    Dim sang2 As Double
    Dim sang3 As Double
    ' initial the sample values

    lyr_name = "upperMC"
    queryData prj_name, lyr_name, x(), y(), z(), v()      'build up dataset based on soil layer name
    ' the following codes assign the variograph parameters, and should be upgraded
    ' later to an subroutine, now it works only for specific project
    nd = UBound(x)
    xmn = 0#
    ymn = 0#
    zmn = 0#
    xsiz = 25#
    ysiz = 25#
    zsiz = 5
    radius = 50#
    radius1 = 50#
    radius2 = 10#

    sang1 = 0#
    sang2 = 0#
    sang3 = 0#

```

Appendix C**Program Code for CrystalGround**

```

nx = 20
ny = 20
nz = 10

queryVariogram prj_name, layer_name

xp = point.x
yp = point.y
zp = point.z

est = -99#
estv = -99#

Select Case UCase(lyr_name)
  Case "UPPERMC"
    cc(0) = 7.5
    aa(0) = 30.5
    kt3dp nd, x(0), y(0), z(0), v(0), _
    nx, xmn, xsiz, ny, ymn, ysiz, nz, zmn, zsiz, c0, _
    nst, it(0), cc(0), aa(0), ang1(0), ang2(0), ang3(0), anis1(0), anis2(0), _
    radius, radius1, radius2, _
    sang1, sang2, sang3, xp, yp, zp, est, estv
  Case Else
    est = -99.9
End Select

Erase x
Erase y
Erase z
Erase v
Erase cc
Erase aa
Erase it
Erase ang1
Erase ang2
Erase ang3
Erase anis1
Erase anis2
kriging = est
End Function

Private Sub FullExtent_bt_Click()
  m_action = MO_FULLEXTENT
  ZoomIn_bt.Enabled = True
  ZoomOut_bt.Enabled = True
  Map1.Extent = Map1.FullExtent
End Sub

Private Sub HoleLog_bt_Click()
  m_action = MO_HOLELOG
End Sub

Private Sub LoadPlan_bt_Click()
  LoadPlan
End Sub

```

Appendix C**Program Code for CrystalGround****Private Sub Map1_DblClick()**

```
Select Case m_action
Case MO_DRAWPL
    m_action = MO_FULLEXTENT
End Select
```

End Sub**Private Sub ChoosePts_bt_Click()**

```
If Not m_action = MO_CHOOSEPOINTS Then
    ZoomIn_bt.Enabled = False
    ZoomOut_bt.Enabled = False
    m_action = MO_CHOOSEPOINTS
Else
    ZoomIn_bt.Enabled = True
    ZoomOut_bt.Enabled = True
    m_action = MO_FULLEXTENT
End If
```

End Sub**Private Sub Command_Click()**

```
Select Case m_action
Case MO_DRAWPL
    ' iProfCount = iProfCount + 1
    ' DoCreateSection
    ' m_action = MO_FULLEXTENT
Case MO_HOIELOG
    If (pts Is Nothing) Then
        ' MsgBox "need define the position of borehole log"
        Exit Sub
    Else
        DoCreateHoleLog
    End If
End Select
```

End Sub**Private Sub DrawPL_bt_Click()**

```
m_action = MO_DRAWPL
```

End Sub**Private Sub Map1_MouseDown(Button As Integer, Shift As Integer, x As Single, y As Single)**

```
Dim pt As New MapObjects2.point
Dim line As New MapObjects2.line
Select Case m_action
Case MO_DRAWPL And Button = 1

    Set rubberL = Map1.TrackLine
    Map1.TrackingLayer.Refresh True
Case MO_HOIELOG And Button = 1
    Set pt = Map1.ToMapPoint(x, y)
    pts.Add pt
Case MO_HOIELOG And Button = 2
    If (pts.Count > 1) Then
        pts.Remove (pts.Count - 1)
    Else
        Set pts = Nothing
    End If
```


Appendix C**Program Code for CrystalGround**

```

End Select
Set pt = Nothing
Set line = Nothing
End Sub

Private Sub Map1_MouseUp(Button As Integer, Shift As Integer, x As Single, y As Single)
    Dim oNewExtent As New MapObjects2.Rectangle
    Dim pt As New MapObjects2.point
    Select Case m_action
    Case MO_PAN
        With Map1
            Set pt = .ToPoint(x, y)
            Set oNewExtent = .Extent
        End With
        With oNewExtent
            .offset pt.x - .Center.x, pt.y - .Center.y
        End With
        Map1.Extent = oNewExtent
        Set oNewExtent = Nothing
        Set pt = Nothing
    End Select
End Sub

Private Sub Pan_bt_Click()
    m_action = MO_PAN
End Sub

Private Sub ZoomIn_bt_MouseDown(Button As Integer, Shift As Integer, x As Single, y As Single)
    Dim newExtent As New Rectangle
    Set newExtent = Map1.Extent
    newExtent.ScaleRectangle 0.6
    Map1.Extent = newExtent
    m_action = MO_ZOOMIN
    Set newExtent = Nothing
End Sub

Private Sub ZoomOut_bt_MouseDown(Button As Integer, Shift As Integer, x As Single, y As Single)
    Dim newExtent As New Rectangle
    Set newExtent = Map1.Extent
    newExtent.ScaleRectangle 1.3
    Map1.Extent = newExtent
    m_action = MO_ZOOMOUT
    Set newExtent = Nothing
End Sub

Private Sub Map1_AfterTrackingLayerDraw(ByVal hDC As stdole.OLE_HANDLE)
    If Not rubberL Is Nothing Then
        Dim sym As New MapObjects2.Symbol
        sym.SymbolType = moLineSymbol
        sym.Style = moSolidLine
        sym.Color = moRed
        sym.Size = 2
        Map1.DrawShape rubberL, sym
    End If

```

Appendix C

Program Code for CrystalGround

End Sub

Private Sub LoadPlan()

```

    Dim db As New MapObjects2.DataConnection
    ' With db
    ' .Server = "sde81:csu006"
    ' .User = ""
    ' .Password = "borehole"
    ' .Database = "sde.CSU006"
    ' End With
    db.Database = App.Path & "\shape\"
    db.Connect
    Dim lyr As New MapObjects2.MapLayer
    Dim sym As New MapObjects2.Symbol

    Map1.Layers.Clear
    'add scope
    If db.Connected Then
        lyr.GeoDataset = db.FindGeoDataset("scope")

        If lyr.Valid Then
            Map1.Layers.Add lyr
        Else
            MsgBox "Invalid layer", vbCritical, "Layer error"
        End If
    End If
    Set sym = Map1.Layers.Item("scope").Symbol
    sym.SymbolType = moFillSymbol
    sym.Color = moLightGray
    sym.Style = moLightGrayFill

    Set lyr = Nothing
    If db.Connected Then
        lyr.GeoDataset = db.FindGeoDataset("planmap")
        If lyr.Valid Then
            Map1.Layers.Add lyr
        Else
            MsgBox "Invalid layer", vbCritical, "Layer error"
        End If
    End If
    Set sym = Map1.Layers.Item("planmap").Symbol
    sym.SymbolType = moFillSymbol
    sym.Color = moBlack
    sym.Style = moCircleMarker

    Map1.Extent = Map1.FullExtent

    FrmMain.Caption = "Boreholes Plan"
    ' FrmMain.ScaleMode = vbPixels
    ' FrmMain.Height = 6500
    Me.Caption = "Plan Map"
    db.Disconnect
    Set db = Nothing
    Set lyr = Nothing
    Set pts = Nothing 'clear the variable

```

End Sub

Appendix C**Program Code for CrystalGround****Private Sub LoadProfile(profileNum As Integer)**

```

Dim db As New MapObjects2.DataConnection
Dim gd As MapObjects2.GeoDataset
Dim lyr As New MapObjects2.MapLayer
Dim strFilter As String
Dim i As Integer
Dim Count As Integer
Dim surfaceFiles() As String
Dim eleDom As msxml.IXMLDOMElement

'obtain the names of the surface files
Count = 0
For i = 0 To strataXML.getElementsByTagName("LAYER").length - 1
    Set eleDom = strataXML.getElementsByTagName("LAYER").Item(i)
    If (eleDom.Attributes.getNamedItem("bottomsurface") Is Nothing) Then
        Count = Count + 1
        ReDim Preserve surfaceFiles(Count)
        surfaceFiles(Count - 1) = eleDom.Attributes.getNamedItem("topsurface").nodeValue
    Else
        Count = Count + 2
        ReDim Preserve surfaceFiles(Count)
        surfaceFiles(Count - 2) = eleDom.Attributes.getNamedItem("topsurface").nodeValue
        surfaceFiles(Count - 1) = eleDom.Attributes.getNamedItem("bottomsurface").nodeValue
    End If
Next i

'build the filter string
strFilter = ""
For i = 0 To UBound(surfaceFiles) - 2
    strFilter = strFilter & "Name=" & surfaceFiles(i) & " _" & CStr(profileNum) & " or "
Next i
strFilter = strFilter & "Name=" & surfaceFiles(i) & " _" & CStr(profileNum) & ""
MsgBox strFilter
db.Database = App.Path & "\shape\"
db.Connect
Set gd = db.FindGeoDataset("profile")
Set lyr.GeoDataset = gd

lyr.FilterExpression = strFilter
Map1.Layers.Clear
Map1.Layers.Add lyr
Me.Caption = "Profile"

db.Disconnect
Set eleDom = Nothing
Set lyr = Nothing
Set gd = Nothing
Set db = Nothing

```

End Sub**Private Sub DoCreateHoleLog()**

```

Dim i As Integer
Dim rs As MapObjects2.Recordset
Dim lyr As New MapObjects2.MapLayer
Dim prjTitle As String
Dim contractorId As String

```


Appendix C**Program Code for CrystalGround**

```
Dim holeID As String
Dim holeXML As New msxml.DOMDocument

Set lyr = Map1.Layers("planmap")
For i = 0 To pts.Count - 1

    Set rs = lyr.SearchByDistance(pts(i), 1#, "")
    If Not rs.EOF Then
        prjTitle = rs.Fields("PROJECT_NA").ValueAsString
        contractorId = rs.Fields("CONTRACTOR").ValueAsString
        holeID = rs.Fields("BOREHOLE_I").ValueAsString 'only one record available
                                                'no loop needed
        MsgBox prjTitle & "::" & contractorId
        Set holeXML = QueryHoleLog(prjTitle, contractorId, holeID)
    Else
        'holeXML = CreatePsedoLog(pts(i))

    End If
Next
End Sub
```

Module: DataManager

Option Explicit

```
Sub queryData(ByVal prjName As String, ByVal lyrName As String, x() As Double, _
              y() As Double, z() As Double, v() As Double)
```

```
    Dim OraSession As Object 'Declare variables as OLE Objects
```

```
    Dim OraDatabase As Object
```

```
    Dim OraDynaset As Object
```

```
    Dim strSelect As String
```

```
    Dim xArray As OraParamArray
```

```
    Dim yArray As OraParamArray
```

```
    Dim zArray As OraParamArray
```

```
    Dim vArray As OraParamArray
```

```
    Dim num As Integer
```

```
    Dim i As Integer
```

```
    Set OraSession = CreateObject("OracleInProcServer.XOraSession")
```

```
    Set OraDatabase = OraSession.DbOpenDatabase("ntugis", "showcase/borehole", 0&)
```

```
    OraDatabase.Parameters.Add "project_name", prjName, ORAPARM_INPUT
```

```
    OraDatabase.Parameters("project_name").serverType = ORATYPE_VARCHAR2
```

```
    OraDatabase.Parameters.Add "layer_name", lyrName, ORAPARM_INPUT
```

```
    OraDatabase.Parameters("layer_name").serverType = ORATYPE_VARCHAR2
```

```
    OraDatabase.Parameters.AddTable "xs", ORAPARM_OUTPUT, ORATYPE_NUMBER, 1000,
22
```

```
    OraDatabase.Parameters.AddTable "ys", ORAPARM_OUTPUT, ORATYPE_NUMBER, 1000,
22
```

```
    OraDatabase.Parameters.AddTable "zs", ORAPARM_OUTPUT, ORATYPE_NUMBER, 1000,
22
```

```
    OraDatabase.Parameters.AddTable "vs", ORAPARM_OUTPUT, ORATYPE_NUMBER, 1000,
22
```

```
    OraDatabase.Parameters.Add "ct", 0, ORAPARM_OUTPUT
```

```
    OraDatabase.Parameters("ct").serverType = ORATYPE_NUMBER
```

```
    strSelect = "BEGIN
```

```
HOLEINFO_PACK.GETVARIABLES(:project_name, :layer_name, :xs, :ys, :zs, :vs, :ct);END;"
```

```
    Set xArray = OraDatabase.Parameters("xs")
```

```
    Set yArray = OraDatabase.Parameters("ys")
```

```
    Set zArray = OraDatabase.Parameters("zs")
```

```
    Set vArray = OraDatabase.Parameters("vs")
```

```
    OraDatabase.ExecuteSQL (strSelect)
```

```
    num = OraDatabase.Parameters("ct").Value
```

```
    'loop the array;
```

```
    ReDim Preserve x(num)
```

```
    ReDim Preserve y(num)
```

```
    ReDim Preserve z(num)
```

```
    ReDim Preserve v(num)
```

```
    For i = 0 To num - 1
```

```
        x(i) = xArray(i + 1) 'OraArray start from 1
```

```
        y(i) = yArray(i + 1)
```

```
        z(i) = zArray(i + 1)
```

```
        v(i) = vArray(i + 1)
```

Appendix C**Program Code for CrystalGround**

```

Next

OraDatabase.Close
Set OraDatabase = Nothing
Set OraSession = Nothing
Set xArray = Nothing
Set yArray = Nothing
Set zArray = Nothing
Set vArray = Nothing
End Sub

Public Function queryStratigraphy(ByVal prjName As String) As msxml.DOMDocument
    Dim OraSession As Object    'Declare variables as OLE Objects
    Dim OraDatabase As Object
    Dim OraDynaset As Object
    Dim strSelect As String
    Dim buffer As Variant
    Dim obj As OraClob
    Dim strXML As String
    Dim objDom As New msxml.DOMDocument
    Dim amount As Long

    Set OraSession = CreateObject("OracleInProcServer.XOraSession")
    Set OraDatabase = OraSession.DbOpenDatabase("ntugis", "demo/borehole", 0&)
    OraDatabase.Parameters.Add "project_name", prjName, ORAPARM_INPUT
    OraDatabase.Parameters("project_name").serverType = ORATYPE_VARCHAR2
    OraDatabase.Parameters.Add "ct", 0, ORAPARM_OUTPUT
    OraDatabase.Parameters("ct").serverType = ORATYPE_CLOB
    strSelect = "begin select stratigraphy into :ct from stratigraphy_table where
project=:project_name;end;"
    OraDatabase.ExecuteSQL (strSelect)
    Set obj = OraDatabase.Parameters("ct").Value
    amount = obj.Read(buffer)
    strXML = buffer

    objDom.async = False
    If (objDom.loadXML(strXML)) Then
        Debug.Print " the xml is loaded"
    Else
        Debug.Print " the xml loading is failed"
    End If
    Set queryStratigraphy = objDom
    Set objDom = Nothing
    OraDatabase.Close
    Set OraDatabase = Nothing
    Set OraSession = Nothing
End Function

Public Function QueryHoleLog(prjName As String, contractor As String, _
holeID As String) As msxml.DOMDocument
    Dim holeDom As New msxml.DOMDocument
    Dim OraSession As Object    'Declare variables as OLE Objects
    Dim OraDatabase As Object
    Dim OraDynaset As Object
    Dim strSelect As String
    Dim buffer As Variant

```


Appendix C

Program Code for CrystalGround

```

Dim obj As OraClob
Dim strXML As String
Dim objDom As New msxml.DOMDocument
Dim amount As Long

Set OraSession = CreateObject("OracleInProcServer.XOraSession")
Set OraDatabase = OraSession.DbOpenDatabase("ntugis", "demo/borehole", 0&)
' Obtain the information about the borehole
OraDatabase.Parameters.Add "project_name", prjName, ORAPARM_INPUT
OraDatabase.Parameters("project_name").serverType = ORATYPE_VARCHAR2
OraDatabase.Parameters.Add "contractor_name", contractor, ORAPARM_INPUT
OraDatabase.Parameters("contractor_name").serverType = ORATYPE_VARCHAR2
OraDatabase.Parameters.Add "hole_ID", holeID, ORAPARM_INPUT
OraDatabase.Parameters("hole_ID").serverType = ORATYPE_VARCHAR2
OraDatabase.Parameters.Add "ct", 0, ORAPARM_OUTPUT
OraDatabase.Parameters("ct").serverType = ORATYPE_CLOB
strSelect = "declare c clob;begin dbms_lob.createTemporary(c, true);" & _
    "c:=getHoleXML(:project_name,:contractor_name,:hole_ID);" & _
    "Delete from temp_xml_table;" & _
    "insert into temp_xml_table values " & _
    " ('doc',c); dbms_lob.freeTemporary(c);" & _
    "select xmldoc into :ct from temp_xml_table " & _
    "where name='doc';end;"

OraDatabase.ExecuteSQL (strSelect)
Set obj = OraDatabase.Parameters("ct").Value
amount = obj.Read(buffer)
strXML = buffer

objDom.async = False
If (objDom.loadXML(strXML)) Then
    Debug.Print objDom.xml
Else
    Debug.Print " the xml loading is failed"
End If
Set QueryHoleLog = objDom

Set objDom = Nothing
OraDatabase.Close
Set OraDatabase = Nothing
Set OraSession = Nothing
End Function

Public Function ObtainLyrName(strataXML As msxml.DOMDocument, _
    pt As MapObjects2.point, prjName As String) As String
    Dim nodeDom As msxml.IXMLDOMNode
    Dim nodeNum As Integer
    Dim i As Integer
    Dim lyrName As String
    Dim result As Boolean

    lyrName = ""
    'Get Stratigraphy information
    If strataXML Is Nothing Then
        MsgBox "no stratigraphy information is available"
    
```

Appendix C

Program Code for CrystalGround

```

Exit Function
End If
nodeNum = strataXML.selectNodes("STRATIGRAPHY").Item(0).childNodes.length
For i = 0 To nodeNum - 1
    Set nodeDom = strataXML.selectNodes("STRATIGRAPHY").Item(0).childNodes.Item(i)
    Debug.Print nodeDom.Attributes.length
    MsgBox nodeDom.Attributes.getNamedItem("name").nodeValue
    If (IsInNode(pt, nodeDom, prjName, lyrName)) Then
        ObtainLyrName = lyrName
    Exit Function
    Else
        Debug.Print "this point is not in the above layer"
    End If
Next i
MsgBox "The point is not in the study domain"
End Function

Public Function IsInNode(pt As MapObjects2.point, oNode As msxml.IXMLDOMNode, _
    prjName As String, layerName As String) As Boolean
    Dim nodeName As String
    Dim topName As String
    Dim bottomName As String
    Dim strInput As String
    Dim strOutput As String
    Dim File As String
    Dim i As Integer
    Dim result As Boolean
    Dim top As Double
    Dim bottom As Double

    'Obtain the top and bottom surface name from the node
    nodeName = oNode.Attributes.getNamedItem("name").nodeValue 'obtain the node name
    MsgBox oNode.xml
    strInput = CStr(pt.x) & "," & CStr(pt.y)

    If (oNode.Attributes.getNamedItem("bottomsurface") Is Nothing) Then
        ' this scenary happens for the final layer where
        ' only top surface is given
    'Get the top value of layer
    topName = oNode.Attributes.getNamedItem("topsurface").nodeValue
    File = App.Path & "\ASCII\" & topName

    strOutput = QueryGrid(File, strInput)
    If strOutput = "NODATA" Then
        top = -9999
    Else
        top = Val(strOutput)
    End If
    If (top = -9999 Or top < pt.z) Then
        IsInNode = False
        Exit Function
    Else
        IsInNode = True
        layerName = nodeName
    End If
Else ' the layer has top surface and bottom surface

```

Appendix C**Program Code for CrystalGround**

```

topName = oNode.Attributes.getNamedItem("topsurface").nodeValue
bottomName = oNode.Attributes.getNamedItem("bottomsurface").nodeValue
'Get the top value of layer
File = App.Path & "\ASCII\" & topName
MsgBox "the topsurface file is " & File
strOutput = QueryGrid(File, strInput)
MsgBox " the top surface is " & strOutput
If strOutput = "NODATA" Then
    top = -9999
Else
    top = Val(strOutput)
End If
'Get the bottom value of layer
File = App.Path & "\ASCII\" & bottomName
MsgBox " the bottom file is " & File
strOutput = QueryGrid(File, strInput)
If strOutput = "NODATA" Then
    bottom = -9999
Else
    bottom = Val(strOutput)
End If
MsgBox " the bottomsurface is " & strOutput

'Check if the point is inside this layer
If (top = -9999 Or pt.z > top Or pt.z < bottom) Then
    IsInNode = False

    Exit Function
Else
    IsInNode = True
    layerName = nodeName
End If
End If

'Check if the layer has sublayer
If (oNode.childNodes.length > 1) Then ' this node has child node
    For i = 0 To oNode.childNodes.length - 1
        result = IsInNode(pt, oNode.childNodes.Item(i), prjName, layerName)
    Next i
End If -

```

End Function

Module: Estimator

Option Explicit

```

Declare Sub kt3dp Lib "C:\gshan\TestServer\DLL\kt3dp.dll" (nd As Long, _
    x As Double, y As Double, z As Double, vr As Double, _
    nx As Long, xmn As Double, xsiz As Double, _
    ny As Long, ymn As Double, ysiz As Double, _
    nz As Long, zmn As Double, zsiz As Double, _
    c0 As Double, nst As Long, it As Long, cc As Double, aa As Double, _
    ang1 As Double, ang2 As Double, ang3 As Double, _
    anis1 As Double, anis2 As Double, _
    radius As Double, radius1 As Double, radius2 As Double, _
    sang1 As Double, sang2 As Double, sang3 As Double, _
    xp As Double, yp As Double, zp As Double, _
    est As Double, estv As Double)

```

```

+   subroutine kt3dp (nd,x,y,z,vr,nx,xmn,xsiz,ny,ymn,ysiz,nz,zmn,zsiz,
+   c0,nst,it,cc,aa,ang1,ang2,ang3,anis1,anis2, radius,radius1,radius2,
+   sang1,sang2,sang3, xp,yp,zp,est,estv)

```

implicit none

!DEC\$ ATTRIBUTES DLLEXPORT::kt3dp

!DEC\$ ATTRIBUTES ALIAS: "kt3dp"::kt3dp

INTERFACE

subroutine setrot(ang1,ang2,ang3,anis1,anis2,ind,MAXROT,rotmat)

implicit none

real*8, intent(in):: ang1

real*8, intent(in):: ang2

real*8, intent(in):: ang3

real*8, intent(in):: anis1

real*8, intent(in):: anis2

integer, intent(in):: ind

integer, intent(in):: MAXROT

real*8, dimension (MAXROT,3,3), intent(inout)::rotmat

end subroutine setrot

```

+   subroutine setsupr(nx,xmn,xsiz,ny,ymn,ysiz,nz,zmn,zsiz,nd,x,y,z,
+   vr,tmp,MAXSBX,MAXSBY,
+   MAXSBZ,nisb,nxsup,xmnsup,xsizsup,nysup,ymnsup,
+   ysizsup,nzsup,zmnsup,zsizsup)

```

implicit none

integer, intent(in):: nx

integer, intent(in):: ny

integer, intent(in):: nz

integer, intent(in):: nd

integer, intent(in):: MAXSBX

integer, intent(in):: MAXSBY

integer, intent(in):: MAXSBZ

real*8, intent(in):: xmn

real*8, intent(in):: xsiz

real*8, intent(in):: ymn

real*8, intent(in):: ysiz

Appendix C

Program Code for CrystalGround

```

real*8, intent(in)::zmn
real*8, intent(in)::zsiz
real*8, dimension(:), intent(inout)::x
real*8, dimension(:), intent(inout)::y
real*8, dimension(:), intent(inout)::z
real*8, dimension(:), intent(inout)::vr
real*8, dimension(:), intent(inout)::tmp
integer, intent(out):: nxsup
integer, intent(out):: nysup
integer, intent(out):: nzsup
real*8, intent(out)::xmnsup
real*8, intent(out)::ymnsup
real*8, intent(out)::zmnsup
real*8, intent(out)::xsizsup
real*8, intent(out)::ysizsup
real*8, intent(out)::zsizsup
integer, dimension(:), intent(out)::nisb
end subroutine setsupr

+ subroutine picksup(nxsup,xsizsup,nysup,ysizsup,nzsup,zsizsup,
    irot,MAXROT,rotmat,radsqd,nsbtosr,ixsbtsr, iysbtsr,izsbtsr)
    implicit none
    integer, intent(in):: nxsup
    integer, intent(in)::nysup
    integer, intent(in)::nzsup
    integer, intent(in)::irot
    integer, intent(in)::MAXROT
    integer, intent(out)::nsbtosr
    real*8, intent(in)::xsizsup
    real*8, intent(in)::ysizsup
    real*8, intent(in)::zsizsup
    real*8, intent(in)::radsqd
    integer, dimension(:), intent(out):: ixsbtsr
    integer, dimension(:), intent(out)::iysbtsr
    integer, dimension(:), intent(out)::izsbtsr
    real*8, dimension(MAXROT,3,3), intent(in):: rotmat
end subroutine

+ subroutine cova3(x1,y1,z1,x2,y2,z2,ivarg,nst,MAXNST,c0,it,cc,aa,
    irot,MAXROT,rotmat,cmax,cova)
    implicit none
    real*8, intent(in)::x1
    real*8, intent(in)::y1
    real*8, intent(in)::z1
    real*8, intent(in)::x2
    real*8, intent(in)::y2
    real*8, intent(in)::z2
    real*8, intent(in)::c0
    integer, intent(in)::ivarg
    integer, intent(in):: nst
    integer, intent(in):: MAXNST
    integer, intent(in)::irot
    integer, intent(in)::MAXROT
    integer, dimension(nst), intent(in)::it
    real*8, dimension(nst),intent(in)::cc
    real*8, dimension(nst),intent(in)::aa

```

Appendix C

Program Code for CrystalGround

```

real*8, dimension(MAXROT,3,3),intent(in)::rotmat
real*8, intent(out)::cmax
real*8, intent(out)::cova
end subroutine cova3

subroutine srchsupr(xloc,yloc,zloc,radsqd,irotd,MAXROT,rotmat,
+      nsbtosr,ixsbtsr,iysbtsr,izsbtsr,noct,nd,
+      x,y,z,tmp,nisb,nxsup,ymnsup,xsizsup,
+      nysup,ymnsup,ysizsup,nzsup,zmnsup,zsizsup,
+      nclose,close1,infect)
implicit none
real*8, intent(in):: xloc
real*8, intent(in)::yloc
real*8, intent(in)::zloc
real*8, intent(in)::radsqd
real*8, intent(in):: xmsup
real*8, intent(in)::ymnsup
real*8, intent(in)::zmnsup
real*8, intent(in)::xsizsup
real*8, intent(in)::ysizsup
real*8, intent(in)::zsizsup
integer, intent(in):: irotd
integer, intent(in):: MAXROT
integer, intent(in)::nsbtosr
integer, intent(in)::noct
integer, intent(in)::nd
integer, intent(in)::nxsup
integer, intent(in)::nysup
integer, intent(in)::nzsup
integer, intent(out)::nclose
integer, intent(out)::infect
real*8, dimension(MAXROT,3,3), intent(in)::rotmat
integer, dimension(:),intent(in)::ixsbtsr
integer, dimension(:),intent(in)::iysbtsr
integer, dimension(:),intent(in)::izsbtsr
real*8, dimension(:),intent(in)::x
real*8, dimension(:),intent(in)::y
real*8, dimension(:),intent(in)::z
real*8, dimension(:),intent(inout)::tmp
integer, dimension(:),intent(in)::nisb
integer, dimension(:),intent(out)::close1
end subroutine srchsupr

subroutine ktsol(n,ns,nv,a,b,x,ktilt,maxeq)
implicit none
integer, intent(in)::n
integer, intent(in)::ns
integer, intent(in)::nv
integer, intent(out)::ktilt
integer, intent(in)::maxeq
real*8, dimension(maxeq), intent(inout)::x
real*8, dimension(maxeq*maxeq),intent(inout)::a
real*8, dimension(maxeq),intent(inout)::b
end subroutine ktsol

```

END INTERFACE

Appendix D Datasets for Synthetic Stratigraphy Generation

This appendix presents the datasets used for the synthetic stratigraphy generation. There are 5 datasets altogether. Each dataset has 50 points. Each point is defined by its coordination x , y , z . The origin of the coordination is at the front-lower-left corner of the 3D domain and all the x , y , z values are positive.

The 5 datasets are used to generate surfaces for soil layers used in the case study of Chapter 7. They are listed as following:

- Dataset for the bottom surface of SandFill layer
- Dataset for the top surface of F2 layer (Part I)
- Dataset for the bottom surface of F2 layer (Part I)
- Dataset for the top surface of F2 layer (Part II)
- Dataset for the bottom surface of F2 layer (Part II)
- Dataset for the bottom surface of MC layer

Generating raster surface from the sample points is conducted using ESRI 3D Raster program. The interpretation method used is the regular spline interpretation with the following parameters: weight=5, sample number=30 and cell=5.

Dataset for bottom surface of SandFill layer

x	y	z	x	y	z
0.0	345.0	39.25	225.0	20.0	43.50
5.0	430.0	42.75	225.0	435.0	42.50
5.0	495.0	40.25	230.0	355.0	40.75
10.0	135.0	40.25	235.0	75.0	41.50
10.0	365.0	40.00	245.0	340.0	40.75
15.0	60.0	42.00	250.0	125.0	41.75
20.0	115.0	42.25	250.0	405.0	42.00
25.0	130.0	42.25	255.0	85.0	42.25
30.0	80.0	41.00	255.0	255.0	43.00
30.0	235.0	41.00	270.0	245.0	42.00
45.0	390.0	40.00	270.0	350.0	42.25
50.0	325.0	41.00	275.0	60.0	40.50
55.0	35.0	42.00	275.0	265.0	43.00
55.0	205.0	41.75	275.0	480.0	43.00
55.0	225.0	40.25	280.0	160.0	41.25
80.0	115.0	40.75	280.0	330.0	42.50
80.0	440.0	39.50	280.0	385.0	41.50
85.0	265.0	42.25	285.0	300.0	41.00
85.0	320.0	42.75	285.0	485.0	41.25
90.0	350.0	41.25	300.0	120.0	41.50
90.0	395.0	42.75	300.0	215.0	41.00
95.0	210.0	40.75	300.0	270.0	40.75
95.0	240.0	40.50	305.0	475.0	43.00
95.0	270.0	42.75	315.0	70.0	43.50
100.0	270.0	42.50	320.0	265.0	41.75
100.0	390.0	41.00	330.0	210.0	41.25
110.0	200.0	41.00	335.0	55.0	39.75
110.0	490.0	42.00	350.0	325.0	40.50
120.0	485.0	39.50	355.0	395.0	41.75
130.0	175.0	41.00	385.0	250.0	41.75
140.0	15.0	41.00	390.0	35.0	41.00
150.0	75.0	43.75	390.0	80.0	40.50
175.0	110.0	42.00	400.0	195.0	40.00
175.0	255.0	43.00	400.0	265.0	41.50
185.0	100.0	41.25	400.0	485.0	43.50
185.0	370.0	40.75	405.0	375.0	41.25
185.0	435.0	41.25	410.0	30.0	42.75
190.0	380.0	40.50	410.0	155.0	39.75
195.0	95.0	41.75	430.0	180.0	40.00
195.0	295.0	42.75	435.0	125.0	41.25
195.0	425.0	42.25	440.0	160.0	41.50
195.0	455.0	43.75	440.0	260.0	42.25
210.0	40.0	42.00	440.0	305.0	42.75
210.0	45.0	42.25	455.0	220.0	40.50
210.0	380.0	43.00	475.0	20.0	42.00
210.0	430.0	42.50	475.0	300.0	42.50
215.0	315.0	42.00	480.0	25.0	43.25

Appendix D

Datasets for Synthetic Stratigraphy Generation

x	y	z	x	y	z
485.0	285.0	41.00	490.0	245.0	42.25
490.0	150.0	41.75	490.0	425.0	41.50

Dataset for top surface of F2 layer (Part I)

x	y	z	x	y	z
5.0	160.0	34.50	225.0	500.0	30.00
10.0	330.0	35.25	240.0	205.0	28.50
15.0	145.0	34.75	255.0	455.0	29.00
35.0	495.0	34.00	260.0	280.0	27.25
40.0	30.0	33.00	265.0	300.0	29.00
50.0	115.0	34.75	270.0	345.0	27.75
60.0	370.0	32.25	270.0	425.0	29.25
65.0	145.0	35.00	285.0	5.0	27.50
70.0	160.0	34.50	285.0	295.0	26.50
85.0	300.0	31.75	295.0	30.0	28.75
85.0	390.0	31.75	305.0	180.0	26.25
90.0	400.0	33.00	310.0	480.0	27.25
105.0	120.0	31.25	315.0	35.0	29.00
105.0	405.0	32.75	315.0	330.0	30.00
110.0	85.0	32.25	325.0	55.0	28.25
115.0	215.0	32.00	325.0	75.0	27.25
130.0	260.0	31.50	325.0	200.0	28.75
130.0	285.0	31.50	325.0	345.0	26.50
130.0	315.0	33.25	325.0	485.0	28.50
135.0	400.0	32.25	335.0	140.0	29.00
140.0	200.0	32.75	335.0	410.0	28.00
145.0	35.0	30.75	340.0	25.0	27.50
145.0	80.0	31.00	340.0	145.0	29.25
150.0	0.0	30.75	345.0	65.0	25.25
155.0	105.0	31.75	345.0	240.0	28.25
155.0	130.0	31.50	345.0	490.0	28.50
170.0	185.0	33.00	370.0	450.0	27.25
170.0	470.0	32.50	380.0	325.0	27.75
175.0	260.0	31.25	380.0	350.0	27.75
175.0	435.0	30.75	385.0	370.0	26.50
185.0	390.0	29.75	390.0	5.0	26.25
185.0	445.0	31.00	390.0	320.0	26.25
190.0	345.0	31.25	405.0	130.0	28.00
190.0	415.0	30.00	405.0	195.0	27.75
190.0	480.0	31.00	410.0	50.0	26.00
195.0	270.0	31.50	410.0	230.0	27.50
205.0	325.0	32.75	415.0	415.0	23.75
210.0	165.0	30.00	420.0	150.0	25.25
220.0	290.0	30.00	420.0	155.0	26.75
220.0	405.0	32.50	420.0	235.0	24.00
225.0	140.0	27.50	425.0	155.0	26.00

Appendix D

Datasets for Synthetic Stratigraphy Generation

x	y	z	x	y	z
225.0	275.0	30.50	445.0	60.0	25.00
445.0	360.0	27.50	470.0	270.0	23.75
450.0	140.0	25.25	470.0	440.0	25.00
455.0	255.0	27.75	485.0	120.0	24.50
455.0	280.0	25.75	485.0	155.0	23.25
460.0	330.0	26.50	485.0	275.0	25.75
460.0	350.0	26.75	490.0	220.0	26.25
465.0	55.0	25.00	490.0	225.0	23.25
465.0	495.0	22.75	500.0	265.0	25.25

Dataset for bottom surface of F2 layer (Part I)

x	y	z	x	y	z
0.0	295.0	28.00	170.0	140.0	30.75
10.0	90.0	28.50	185.0	460.0	32.25
10.0	310.0	28.00	190.0	220.0	31.75
10.0	455.0	27.25	190.0	470.0	32.00
15.0	110.0	27.50	200.0	155.0	32.00
15.0	315.0	27.75	215.0	25.0	31.25
25.0	65.0	27.25	220.0	290.0	31.75
25.0	470.0	28.75	230.0	445.0	32.25
30.0	30.0	28.50	240.0	65.0	32.25
30.0	55.0	27.25	250.0	485.0	31.75
30.0	430.0	28.00	255.0	190.0	33.00
30.0	495.0	27.75	255.0	290.0	32.75
40.0	390.0	29.25	265.0	120.0	33.00
45.0	175.0	29.00	270.0	385.0	33.25
45.0	440.0	28.50	275.0	230.0	32.50
45.0	500.0	28.50	275.0	495.0	32.50
55.0	95.0	28.50	280.0	320.0	33.75
60.0	215.0	29.50	290.0	335.0	33.25
60.0	375.0	28.75	295.0	210.0	32.75
65.0	105.0	28.50	300.0	40.0	33.50
85.0	5.0	29.00	300.0	90.0	33.50
100.0	315.0	29.75	305.0	290.0	34.25
105.0	405.0	28.75	310.0	45.0	34.00
105.0	420.0	29.50	310.0	140.0	32.00
105.0	445.0	28.75	320.0	390.0	33.50
110.0	150.0	29.00	330.0	325.0	33.75
110.0	275.0	30.50	335.0	340.0	34.50
120.0	500.0	30.25	345.0	260.0	34.75
125.0	150.0	29.75	345.0	380.0	34.50
135.0	425.0	29.50	350.0	185.0	34.00
140.0	150.0	31.75	350.0	275.0	34.25
140.0	475.0	31.50	355.0	305.0	34.75
145.0	235.0	31.25	360.0	240.0	34.50
155.0	355.0	30.00	365.0	5.0	33.75

Appendix D

Datasets for Synthetic Stratigraphy Generation

x	y	z	x	y	z
160.0	205.0	31.50	365.0	195.0	33.75
165.0	440.0	30.50	375.0	90.0	34.50
375.0	235.0	35.25	460.0	345.0	36.75
400.0	120.0	35.75	465.0	150.0	37.00
405.0	335.0	36.00	470.0	180.0	36.25
405.0	355.0	36.50	470.0	355.0	36.75
405.0	500.0	36.00	475.0	240.0	37.75
410.0	355.0	35.00	480.0	165.0	37.50
425.0	160.0	36.50	480.0	260.0	37.75
430.0	0.0	37.00	480.0	405.0	37.50
430.0	175.0	36.25	485.0	240.0	37.25
440.0	200.0	37.75	490.0	495.0	37.50
440.0	220.0	36.25	495.0	25.0	37.25
440.0	370.0	36.75	500.0	80.0	36.50
445.0	60.0	36.75	500.0	275.0	38.00
450.0	365.0	36.00	500.0	420.0	37.50

Dataset for top surface of F2 layer (Part II)

x	y	z	x	y	z
0.0	470.0	29.00	210.0	105.0	29.75
5.0	365.0	28.50	215.0	485.0	30.25
10.0	170.0	28.25	225.0	295.0	29.50
20.0	25.0	28.50	225.0	355.0	30.00
35.0	160.0	29.50	225.0	440.0	29.75
35.0	425.0	29.25	230.0	40.0	29.75
40.0	120.0	28.50	230.0	165.0	29.75
45.0	255.0	28.75	230.0	450.0	29.50
50.0	490.0	28.00	235.0	190.0	29.75
60.0	20.0	29.00	240.0	0.0	29.75
60.0	150.0	29.25	245.0	405.0	29.75
65.0	125.0	29.50	245.0	425.0	30.00
80.0	390.0	29.25	250.0	390.0	30.00
95.0	210.0	29.00	255.0	405.0	30.25
100.0	205.0	28.75	260.0	25.0	29.75
105.0	480.0	29.25	260.0	310.0	30.50
120.0	415.0	30.25	260.0	500.0	30.50
130.0	200.0	29.25	275.0	430.0	29.25
140.0	105.0	29.50	280.0	255.0	30.75
145.0	245.0	29.75	280.0	300.0	30.25
160.0	430.0	29.50	280.0	395.0	30.25
160.0	435.0	29.00	290.0	270.0	31.50
170.0	360.0	29.25	290.0	335.0	30.00
170.0	450.0	29.50	295.0	160.0	30.25
175.0	35.0	30.00	300.0	330.0	29.75
180.0	80.0	30.00	300.0	365.0	29.75
180.0	240.0	30.00	300.0	425.0	30.25

Appendix D

Datasets for Synthetic Stratigraphy Generation

x	y	z	x	y	z
185.0	465.0	30.25	310.0	230.0	30.50
195.0	275.0	29.50	310.0	275.0	29.75
200.0	205.0	29.75	315.0	275.0	30.50
335.0	310.0	30.75	430.0	320.0	30.75
335.0	315.0	30.75	430.0	385.0	31.00
340.0	475.0	30.25	440.0	375.0	31.00
345.0	150.0	30.25	445.0	65.0	31.50
350.0	290.0	30.75	445.0	85.0	32.00
355.0	145.0	30.75	445.0	120.0	30.75
370.0	245.0	30.25	445.0	290.0	30.75
385.0	355.0	32.00	450.0	0.0	32.00
390.0	435.0	30.75	455.0	265.0	31.50
395.0	135.0	30.75	460.0	130.0	31.50
400.0	220.0	30.50	460.0	310.0	31.00
400.0	435.0	31.75	465.0	35.0	31.75
405.0	95.0	31.50	465.0	460.0	31.00
405.0	500.0	32.25	470.0	85.0	31.50
410.0	405.0	31.25	470.0	265.0	32.50
415.0	255.0	31.50	470.0	270.0	31.25
420.0	195.0	31.50	480.0	250.0	31.25
420.0	410.0	30.50	485.0	160.0	32.00
425.0	40.0	31.50	485.0	335.0	31.50
430.0	140.0	30.75	490.0	430.0	31.50

Dataset for bottom surface of F2 layer (Part II)

x	y	z	x	y	z
0.0	275.0	34.50	155.0	160.0	33.00
20.0	345.0	34.25	155.0	235.0	32.75
30.0	150.0	33.75	165.0	160.0	33.00
30.0	215.0	34.75	175.0	80.0	32.25
40.0	260.0	34.75	175.0	255.0	32.50
60.0	280.0	34.25	180.0	80.0	32.00
65.0	170.0	34.75	180.0	100.0	32.00
70.0	320.0	33.75	180.0	290.0	31.75
75.0	325.0	33.00	185.0	135.0	31.75
90.0	335.0	33.25	190.0	390.0	31.75
90.0	440.0	34.25	200.0	95.0	31.50
95.0	360.0	34.25	205.0	295.0	32.00
115.0	215.0	33.75	205.0	335.0	30.75
125.0	120.0	32.50	210.0	10.0	33.75
125.0	195.0	32.25	210.0	185.0	32.50
130.0	355.0	32.75	215.0	80.0	30.50
135.0	200.0	33.75	215.0	265.0	31.50
135.0	205.0	31.50	215.0	395.0	31.50
140.0	25.0	33.75	220.0	75.0	30.50
140.0	495.0	33.25	220.0	200.0	30.50

Appendix D

Datasets for Synthetic Stratigraphy Generation

x	y	z	x	y	z
150.0	235.0	32.25	225.0	455.0	31.75
150.0	255.0	32.25	240.0	90.0	29.50
150.0	390.0	33.25	250.0	210.0	29.50
155.0	130.0	33.00	250.0	410.0	30.75
265.0	95.0	29.75	400.0	270.0	27.50
270.0	330.0	30.25	405.0	325.0	28.25
275.0	75.0	30.25	405.0	450.0	27.00
280.0	235.0	29.00	410.0	110.0	27.00
285.0	455.0	30.75	415.0	260.0	27.50
290.0	95.0	30.00	425.0	155.0	26.75
295.0	170.0	29.00	425.0	375.0	27.25
295.0	265.0	29.75	430.0	495.0	26.75
300.0	145.0	29.00	435.0	140.0	27.25
300.0	330.0	30.00	435.0	280.0	26.75
300.0	355.0	28.75	435.0	325.0	26.50
300.0	485.0	28.50	440.0	80.0	26.25
305.0	20.0	29.00	440.0	370.0	25.50
320.0	310.0	28.75	445.0	230.0	27.75
330.0	455.0	29.75	450.0	275.0	27.00
350.0	165.0	29.75	450.0	415.0	26.75
355.0	145.0	28.50	460.0	495.0	26.25
355.0	425.0	29.50	465.0	55.0	26.50
365.0	155.0	27.75	470.0	175.0	25.75
365.0	335.0	28.00	470.0	215.0	26.25
375.0	365.0	29.25	470.0	260.0	26.50
385.0	285.0	27.75	470.0	335.0	26.50
390.0	20.0	28.00	475.0	325.0	25.75
395.0	175.0	28.75	480.0	35.0	26.25
395.0	255.0	27.00	495.0	260.0	25.75
400.0	180.0	28.50	500.0	105.0	25.50

Dataset for bottom surface of MC layer

x	y	z	x	y	z
0.0	175.0	8.50	80.0	415.0	9.00
5.0	435.0	11.00	90.0	115.0	11.75
25.0	145.0	8.25	95.0	320.0	10.50
25.0	325.0	11.75	100.0	80.0	10.00
30.0	20.0	10.25	110.0	360.0	9.50
35.0	500.0	11.50	125.0	270.0	9.50
40.0	90.0	8.75	135.0	90.0	9.00
40.0	245.0	9.75	140.0	320.0	8.75
50.0	275.0	7.50	155.0	0.0	10.25
55.0	55.0	9.00	155.0	210.0	10.25
55.0	115.0	9.75	160.0	280.0	8.25
55.0	235.0	9.00	165.0	110.0	11.00
55.0	240.0	10.50	165.0	440.0	9.50

Appendix D

Datasets for Synthetic Stratigraphy Generation

x	y	z	x	y	z
60.0	140.0	10.00	175.0	440.0	8.50
60.0	165.0	9.00	180.0	45.0	10.00
70.0	50.0	10.50	185.0	475.0	11.50
70.0	95.0	10.50	190.0	145.0	10.00
75.0	320.0	10.50	195.0	75.0	8.50
195.0	435.0	9.50	330.0	160.0	10.00
200.0	370.0	8.75	335.0	40.0	10.00
205.0	80.0	9.00	335.0	300.0	10.00
205.0	105.0	9.25	340.0	405.0	9.50
205.0	225.0	11.25	350.0	415.0	10.75
205.0	260.0	11.00	365.0	460.0	8.25
210.0	30.0	10.50	370.0	10.0	12.00
210.0	175.0	10.00	370.0	190.0	8.50
215.0	95.0	10.25	370.0	430.0	9.25
220.0	75.0	10.00	380.0	150.0	10.75
220.0	490.0	10.00	380.0	215.0	8.25
225.0	170.0	11.25	380.0	370.0	11.75
225.0	235.0	10.00	385.0	230.0	9.25
230.0	270.0	11.00	395.0	250.0	9.50
250.0	280.0	9.00	395.0	285.0	9.75
250.0	500.0	10.00	395.0	390.0	8.00
255.0	160.0	10.75	400.0	285.0	9.25
260.0	155.0	8.00	420.0	215.0	9.00
270.0	100.0	9.00	420.0	240.0	10.75
270.0	400.0	8.75	430.0	460.0	9.00
275.0	105.0	11.25	450.0	195.0	9.50
275.0	230.0	8.25	450.0	225.0	9.75
275.0	330.0	9.00	455.0	115.0	10.50
285.0	160.0	11.00	460.0	425.0	8.75
285.0	480.0	9.50	470.0	200.0	9.75
290.0	275.0	8.00	470.0	375.0	9.50
290.0	345.0	11.50	480.0	5.0	7.75
295.0	140.0	10.00	485.0	350.0	9.00
315.0	225.0	7.50	490.0	30.0	9.75
315.0	475.0	10.50	490.0	110.0	9.50
325.0	110.0	9.50	490.0	285.0	9.75
325.0	145.0	9.75	490.0	455.0	9.25

Appendix E Soil Investigation Borehole Logs

This appendix presents borehole log reports for a resident building project. The site investigations were conducted in two stages with different contractors. There are eleven boreholes altogether. Four of them (BH-1, 2, 3, 4) are drilled at the first stage and the other seven (K-1, 2, 3, 4, 5, 6, 7) boreholes are drilled at the second stages.

This project is used as a case study in Chapter 8. The data will be purely used for this study only. The project information and contractor's logos are removed in the following logs.

Appendix E

Soil Investigation Borehole Logs

CLIENT:		PROJECT:		LOCATION:		BOREHOLE NO: BH 1		JCC NO: SIL / 4831 / 01				
DRILLING DATE: 29 March to 02 April 2001		GROUND WATER LEVEL: 1.4 m		BOREHOLE DIA: 100 mm		COREHOLE DIA: - mm		DRILLING METHOD: Rotary				
REDUCED LEVEL: - m		NORTHING: - m		EASTING: - m		LEGEND:						
☒ -Standard Penetration Test (SPT) ☒ -Undisturbed Sample (UD) ☒ -Disturbed Sample (DS)		☐ -Attempt With No Recovery ☒ -Field Vane Test (FVT) ☐ -Piston Sample (PS) ☐ -Core Run (CR)		☐ UNCT ☒ UU ☐ CIU ☐ CID								
Scale	Description	Graphic Log	SPT (N-Value)	Blows/cm	Samples & Tests	Rock	Bulk Density	Water Content	Shear Strength	Atterberg Limit (%)	Grain Size Analysis	
					Type & No.	Depth (m)	TCR (%)	ROD (%)	C (%)	ϕ (°)	LL	PL
1	0.00-1.30m FILL - Soft Sandy Clay. Dark Brown.											
2	1.30-2.00m FILL - Firm Silty Clay. Reddish Pink/Light Grey.											
3	2.00-6.30m Soft Organic Peaty Clay With Wood. Black.				UD1	2.00-2.50		1.43	81.8	12	-	-
4					SPT1	2.50-2.95						
5					UD2	3.00-3.50		1.22	153.4	15	0	135
6					SPT2	3.50-3.95						
7					UD3	4.50-5.00			130.6			
8					SPT3	5.00-5.45						
9	6.30-7.90m Soft Marine Clay. Greenish Grey.				UD4	6.00-6.50		1.47	100.7	5.5	0	-
10					FVT1	7.00		1.52	82.2	13		-
11	7.90-10.40m Stiff Silty Clay. Reddish Brown/Light Grey.				UD5	9.00-9.50		1.96	27.0	22	5	36
12					SPT4	9.50-9.95						
13	10.40-14.00m Stiff Sandy Silt. Light Grey.				UD6	12.00-12.50		1.94	24.0	25	8.5	-
14					SPT5	12.50-12.95						
15	14.00-16.70m Stiff Sandy Clay. Yellowish Brown.				SPT6	15.00-15.45						
16												
17	16.70-19.40m Very Stiff Clayey Silt With Sand. Greyish Yellow.				SPT7	18.00-18.45						
18												
19	19.40-22.50m Hard Clayey Silt. Greyish Yellow.											
20												
TO BE CONTINUED												
REMARKS:												

Page 1 of 2

BH-1 (1)

Appendix E

Soil Investigation Borehole Logs

CLIENT: City of Vancouver		PROJECT: City of Vancouver Housing Development		LOCATION: City of Vancouver		JOB NO: 001 / 01	
BOREHOLE NO: BH 2		GROUND WATER LEVEL: 1.2 m		BOREHOLE DIA: 100 mm		COREHOLE DIA: - mm	
DRILLING DATE: 23 March to 27 March 2001		DRILLING METHOD: Rotary		REDUCED LEVEL: - m		NORTHING: - m	
EASTING: - m		LEGEND:		☐ -Attempt With No Recovery ☐ UNCT ☒ -Standard Penetration Test (SPT) ☒ -Field Vane Test (FVT) ☒ -Undisturbed Sample (UD) ☒ -Piston Sample (PS) ☒ -Disturbed Sample (DS) ☒ -Core Run (CR)		☐ UU ☐ CIU ☐ CID	

Scale	Description	Graphic Log	SPT (N-Value)		Samples & Tests		Rock		Bulk Density (kg/m³)	Water Content (%)	Shear Strength		Atterberg Limit (%)		Grain Size Analysis G/SA/S/C
			Blows/30 cm	Depth (m)	Type & No.	Type (%)	Type (%)	C _u (kN/m²)			φ (°)	LL	PL		
21	16.70-21.00m Hard Sandy Silt. Greyish Yellow.	X X X X X X X X X	42/30	21.00-21.45	SPT9										
22	21.00-31.10m Hard Silty Clay. Greyish Yellow.	X X X X X X X X X	100/30	24.00-24.45	SPT10										
23		X X X X X X X X X													
24		X X X X X X X X X													
25		X X X X X X X X X													
26		X X X X X X X X X													
27		X X X X X X X X X	100/25	27.00-27.40	SPT11										
28		X X X X X X X X X													
29		X X X X X X X X X													
30		X X X X X X X X X	100/22	30.00-30.37	SPT12										
31	31.10-33.10m Hard Sandy Silt. Greyish Yellow.	X X X X X X X X X													
32		X X X X X X X X X													
33	Borehole terminated at 33.1m	X X X X X X X X X	100/10	33.00-33.10	SPT13										
34															
35															
36															
37															
38															
39															
40															

REMARKS:

Page 2 of 2

Appendix E

Soil Investigation Borehole Logs

CLIENT:		PROJECT: Proposed Erection Of A Condominium Building									
LOCATION:		BOREHOLE NO: BH 3									
BOREHOLE NO: BH 3		JOB NO: SIL / 4831 / 01									
DRILLING DATE: 28 March to 29 March 2001		GROUND WATER LEVEL: 1.5 m									
BOREHOLE DIA: 100 mm		COREHOLE DIA: - mm									
DRILLING METHOD: Rotary		REDUCED LEVEL: - m									
NORTHING: - m		EASTING: - m									

LEGEND:		☐ - Attempt With No Recovery		☐ UNCT	
☒ - Standard Penetration Test (SPT)	☒ - Field Vane Test (FVT)	☒ UU			
☒ - Undisturbed Sample (UD)	☒ - Piston Sample (PS)	☒ CIU			
☒ - Disturbed Sample (DS)	☒ - Core Run (CR)	☒ CID			

Scale	Description	Graphic Log	SPT (N-Value)		Samples & Tests		Rock		Bulk Density (Mg/m³)	Water Content (%)	Shear Strength		Atterberg Limit (%)		Grain Size Analysis G/S/A/S/C
			Blows/cm	Depth (m)	Type & No.	Depth (m)	TCR (%)	RCO (%)			Cu (kN/m²)	φ (°)	LL	PL	
1	0.00-1.50m FILL - Firm Silty Clay. Dark Brown.	X X X X													
2	1.50-2.80m FILL - Soft Sandy Clay. Greyish Brown.	X X X X	2/30		UD1 1.50-1.95 SPT1 1.95-2.40			1.94	26.6	13	-	47	36	-	
3	2.80-4.50m Soft Organic Peaty Clay. Black.	X X X X	2/30		UD2 3.00-3.45 SPT2 3.45-3.90			1.42	88.0	14	0	-	-	-	
4	4.50-7.60m Firm Silty Clay. Light Grey.	X X X X	2/30		UD3 4.50-4.95 SPT3 4.95-5.40			1.89	31.6	11	-	-	-	-	
5		X X X X	2/30		UD4 6.00-6.45 SPT4 6.45-6.90			2.03	23.1	-	-	36	21	-	
6		X X X X													
7		X X X X													
8	7.60-12.50m Loose Clayey Sand. Brownish Yellow.	X X X X	3/30		UD5 9.00-9.45 SPT5 9.45-9.90			1.99	23.1	10.5	-	-	-	0/78/22	
9		X X X X													
10		X X X X													
11		X X X X													
12		X X X X													
13	12.50-15.00m Stiff Silty Clay. Light Brownish Yellow.	X X X X	10/30		UD6 12.00-12.45 SPT6 12.45-12.90			1.98	23.4	7.5	27	-	-	2/78/20	
14		X X X X													
15	15.00-17.90m Very Stiff Sandy Silt. Greyish Yellow.	X X X X	26/30		UD7 15.00-15.45 SPT7 15.45-15.90			2.01	23.9	47.5	14	41	22	-	
16		X X X X													
17		X X X X													
18	17.90-36.15m Hard Sandy Silt. Greyish Yellow.	X X X X	66/30		SPT8 18.00-18.45										
19		X X X X													
20		X X X X													

TO BE CONTINUED

REMARKS:

Page 1 of 2

BH-3 (1)

Appendix E

Soil Investigation Borehole Logs

CLIENT:		PROJECT:		LOCATION:		BOREHOLE NO: BH. 4		JOB NO: 71	
DRILLING DATE: 23 March to 26 March 2001		GROUND WATER LEVEL: 1.5 m		BOREHOLE DIA: 100 mm		COREHOLE DIA: - mm		DRILLING METHOD: Rotary	
REDUCED LEVEL: - m		NORTHING: - m		EASTING: - m		UNCT		UU	
CIU		CID		Legend:		-Attempt With No Recovery		-Field Vane Test (FVT)	
-Standard Penetration Test (SPT)		-Undisturbed Sample (UD)		-Disturbed Sample (DS)		-Piston Sample (PS)		-Core Run (CR)	
Scale		Description		Graphic Log		SPT (N-Value)		Samples & Tests	
						Blows/cm		Depth (m)	
						Legend		Type & No.	
						TCR (%)		Rock	
						Bulk Density (kg/m ³)		Water Content (%)	
						Shear Strength		Atterberg Limit (%)	
						C _u (kN/m ²)		φ (°)	
						LL		PL	
						Grain Size Analysis		G/S _A /S _S /C	
0.00-2.00m		FILL - Medium Stiff Sandy Clay. Yellowish Brown.							
2.00-5.50m		Medium Stiff To Stiff Silty Clay. Reddish Brown/Light Grey.							
5.50-8.50m		Medium Stiff Silty Clay. Light Greyish Pink.							
8.50-12.00m		Loose Clayey Sand. Reddish Pink.							
12.00-17.60m		Very Stiff Silty Clay. Greenish Grey.							
17.60-30.30m		Hard Sandy Silt. Greenish Yellow.							
TO BE CONTINUED									
REMARKS:									

BH-4 (1)

Appendix E

Soil Investigation Borehole Logs

CLIENT:		Proposed Fraction Of A Condominium Housing Development Comprising 2 Phases									
PROJECT:		DT 12 2001									
LOCATION:											
BOREHOLE NO: BH 4		JOB NO:		GROUND WATER LEVEL: 1.5 m							
DRILLING DATE: 23 March to 26 March 2001		BOREHOLE DIA: 100 mm									

LEGEND:		☐ - Attempt With No Recovery ☒ - Field Vane Test (FVT) ☐ - Piston Sample (PS) ☒ - Core Run (CR)		☐ UNCT ☒ UU ☐ CIU ☒ CID		COREHOLE DIA: - mm DRILLING METHOD: Rotary REDUCED LEVEL: - m NORTHING: - m EASTING: - m				
☒ - Standard Penetration Test (SPT) ☐ - Undisturbed Sample (UD) ☒ - Disturbed Sample (DS)										

Scale	Description	Graphic Log	SPT (N-Value)		Samples & Tests		Rock		S _u Density (kg/m ³)	Water Content (%)	Shear Strength		Atterberg Limit (%)		Grain Size Analysis G/S _u /S _c
			Blows/cm	Legend	Type & No.	Depth (m)	TCR (%)	ROD (%)			C _u (kN/m ²)	φ (°)	LL	PL	
21	17.60-30.30m Hard Sandy Silt Greenish Yellow.		44/30	SP19	21.00-21.45										
22															
23															
24			100/15	SP110	24.00-24.15										
25															
26															
27			100/30	SP111	27.00-27.30										
28															
29															
30			100/30	SP112	30.00-30.30										
31	Borehole terminated at 30.3m														
32															
33															
34															
35															
36															
37															
38															
39															
40															

REMARKS:	
----------	--

Appendix E

Soil Investigation Borehole Logs

FIG DRILLING LOG

Project No. _____ Project _____ INC Type of Drilling: ROTARY
Hole Number K-1 (PAGE 1 of 2) _____ Date 07/01/03-11/01/03
Water Table 0.150 m Elevation 101.535 m Driller _____

Remarks
P: Standard Penetration Test
UDP: Fixed-Piston Undisturbed Sampling
UDS: Open-Drive Undisturbed Sampling
VST: Vane Shear Test

Scale in m	Elevation in m	Depth in m	Thickness in m	Legend	Type of Soil	Colour	Relative Density or Consistency	General Remarks	Sampling		Standard Penetration Test					
									Depth in m	Sample No.	N-Value Blows/30cm	Blows Per Each 10 cm	N - Value 10 20 30 40 50			
1	102.04	1.50	1.50		Sand	Brown	Very Loose	Sand is fine to coarse								
2					Silty Clay	Yellowish Brown	Medium Stiff	Mixed with gravel, Hardcore and broken pieces of concrete etc.	1.85	P-1R	6	2	2	2		
3									1.95							
4	99.04	4.50	3.00						1.00	UDP-1R	Recovery = 20/20 cm					
5					Sand	Grey	Very Loose	Sand is coarse	1.20	P-2R	13	7	4	2		
6	97.54	6.00	1.50						1.85	P-3R	1	1	0	0		
7					Silty Clay	Grey	Very Soft	Marine clay. On top portion mixed with coarse sand and significantly and shell fragments. Similarly mixed with fine gravel and coarse sand at 11.5-11.7m.	1.95							
8									6.15	P-4R	0	0	0	0		
9									6.45							
10									7.00							
11									7.85	UDP-1R	Recovery = 25/25 cm					
12	91.54	12.00	6.00						8.50	VST-1	Sw = 12.8 %					
13					Silty Sand	Light Greenish Grey	Medium Stiff	Old Alluvium. Sand is fine to medium	10.00							
14	89.54	14.00	2.00						10.85	UDP-2R	Recovery = 50/25 cm					
15					Silty Sand	Light Greenish Grey	Medium Dense	Sand is fine to medium	11.50	VST-2	Sw = 34.5 %					
16									13.10	P-5R	6	1	2	3		
17	86.04	17.50	3.50						13.45							
18					Silty Clay	Brownish Grey	Hard	Mottled with greenish grey.	14.00	UDP-3R	Recovery = 50/25 cm					
19									15.65	P-6R	21	6	7	8		
20	83.54	20.00	2.50						17.55	P-7R	40	10	13	17		
21	82.04	21.50	1.50		Clayey Sil	Brownish Grey	Hard	Mottled with greenish brown.	18.55	P-8R	36	11	13	14		
22	80.04	23.50	2.00		Silty Sand	Greenish Grey	Very Dense	Sand is fine to coarse	21.65	P-9R	50	15	15	20		
23	80.04	23.50	2.00						21.85							
24	78.04	25.50	2.00		Silty Clay	Greenish Grey	Hard		23.65	P-10R	42	10	15	17		
25	76.04	27.50	2.00		Clayey Sil	Greenish Grey	Hard	Mottled with brown	25.65	P-11R	43	14	14	15		
26	76.04	27.50	2.00						25.95							
27									27.60	P-12R	100/21	31	48	21/1		
28	68.59	33.95	6.45		Silty Sand	Greenish Grey	Very Dense	Sand is fine to coarse	27.80							
29									28.65	P-13R	56	16	17	23		
30									29.95							
31																

Prepared by: _____ Checked by: _____ Approved by: _____

K-1 (1)

Appendix E

Soil Investigation Borehole Logs

FIG

DRILLING

LOG

Project No. SH-31

Project S

Type of Drilling ROTARY

Hole Number K-1 (PAGE 2 of 2)

AT

Date 07/01/03-11/01/03

Water Table 0.150 m

Elevation RL+101.535 m

Driller

Remarks

P : Standard Penetration Test

UOP : Fixed-Piston Undisturbed Sampling

UOS : Open-Drive Undisturbed Sampling

YST : Vane Shear Test

Scale in m	Elevation in m	Depth in m	Thickness in m	Legend	Type of Soil	Colour	Relative Density or Consistency	General Remarks	Sampling		Standard Penetration Test					
									Depth in m	Sample No.	N-Value Blows/30cm	Blows Per Each 10 cm	N - Value			
											10	20	30	40	50	
31					Silly Sand	Greenish Grey	Very Dense	Sand is fine to coarse								
32									31.60	B-14	100/8	100/8				100 B.L.S/5cm
33									31.68							
34	69.59	31.95	6.45						31.65	P-15	100	18	36	46		100 B.L.S/10cm
35								-END OF DRILLING-	31.95							
36																
37																
38																
39																
40																
41																
42																
43																
44																
45																
46																
47																
48																
49																
50																
51																
52																
53																
54																
55																
56																
57																
58																
59																
60																
61																

Prepared by:

Checked by:

Approved by:

Appendix E

Soil Investigation Borehole Logs

FIG DRILLING LOG

Project No. _____ Project S. _____ 46 Type of Drilling ROTARY

Hole Number K-2 (PAGE 1 of 1) Date 07/03/03-11/01/03

Water Table 0.0-0.50 m. Elevation RL+102.980 m. Driller _____

Remarks

P : Standard Penetration Test
 UQP : Fixed-Piston Undisturbed Sampling
 UDS : Open-Drive Undisturbed Sampling
 VST : Vane Shear Test

Scale in m	Elevation in m	Depth in m	Thickness in m	Legend	Type of Soil	Colour	Relative Density or Consistency	General Remarks	Sampling		Standard Penetration Test								
									Depth in m	Sample No.	N-Value Blows/30cm	Blows Per Each 10 cm	N - Value						
											10cm	10cm	10cm	10	20	30	40	50	
1					Sand	Brown	Very Loose	Sand is fine to coarse. (FW)											
2									1.85										
3									1.95										
4	96.48	3.50	3.50		Peaty Clay	Dark Brown	Very Soft	With decomposed woods, traces of organic matter	3.15										
5	96.48	4.50	1.00		Silty Clay	Dark Grey	Very Soft	Marine clay. With shell fragments.	3.15										
6									3.50										
7									4.30										
8									5.00										
9									5.20										
10									6.00										
11									6.50										
12	90.98	12.00	7.50		Clayey Sil	Light and Dark Grey	Soft	Mixed with fine to coarse sand. With shell fragments on top.	7.00										
13	88.73	13.25	1.25		Clayey Sil	Greenish Grey	Stiff	Old Alluvium. With fine sand.	7.85										
14									8.50										
15									9.00										
16									9.85										
17	85.48	17.50	4.25		Silty Clay	Yellowish Brown	Hard	Weathered with greenish. With pockets of sand.	11.50										
18									13.00										
19	81.46	19.50	2.00		Silty Sand	Greenish Grey	Very Dense to Dense	Sand is fine to coarse.	13.70										
20									14.65										
21	81.46	21.50	2.00		Silty Clay	Greenish Grey to Yellowish Brown	Very Stiff	With some fine sand	14.95										
22									16.15										
23									16.45										
24	77.43	25.50	4.00		Silty Sand	Greenish Grey	Very Dense	Sand is fine to coarse.	17.65										
25									17.95										
26	75.11	27.84	2.34						18.65										
27									18.95										
28									21.65										
29									21.95										
30									23.65										
31									23.77										
									25.65										
									27.65										
									27.84										

Prepared by: _____ Checked by: _____ Approved by: _____

-END OF DRILLING-

K-2

Appendix E

Soil Investigation Borehole Logs

FIG DRILLING LOG

Project No. _____

Hole Number K-3 (PAGE 1 of 1)

Water Table 0.2-1.20 m.

Project _____

Elevation R+101.951 m.

Type of Drilling ROTARY

Date 08/01/03-09/01/03

Driller _____

Remarks

P : Standard Penetration Test
UOP : Fixed-Piston Undisturbed Sampling
UOS : Open-Drive Undisturbed Sampling
VSI : Vane Shear Test

Scale in m	Elevation in m	Depth in m	Thickness in m	Legend	Type of Soil	Colour	Relative Density or Consistency	General Remarks	Sampling		Standard Penetration Test					
									Depth in m	Sample No.	N-Value Blows/30cm	Blows Per Each 10 cm			N - Value	
											10	20	30	40	50	
1					Silty Clay	Yellowish Brown	Soft	enlited with reddish brown of some portion and occurred broken pieces of stone brick significantly								
2									1.85	P-1	4	1	2			
3	100.95	3.00	3.00		Organic Clay	Dark Grey	Soft	With decomposed wood and some organic matters	3.15	P-2	2	1	0	1		
4	99.45	4.50	1.50						3.45							
5					Silty Clay	Bluish Grey	Very Soft	Moist clay. Traces of some decomposed materials and shell fragments. With some fine to coarse sand at bottom.	4.50							
6									5.30	UOP-1						
7									6.00	VSI-1						
8									6.50							
9									7.30	UOP-2						
10									8.00	VSI-2						
11									9.50							
12									10.30	UOP-3						
13									11.00	VSI-3						
14	89.95	14.00	9.50						12.50							
15					Silty Sand	Greenish Grey	Medium Dense	Old Alluvium. Sand is fine to medium	13.30	UOP-4						
16	88.01	15.95	1.95						14.00	VSI-4						
17									15.65	P-3	12	3	4	5		
18									15.95							
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																

Prepared by: _____

Checked by: _____

Approved by: _____

Appendix E

Soil Investigation Borehole Logs

FIG

DRILLING

LOG

Project No. 3

Hole Number K-4 (PAGE 1 of 1)

Water Table Q-00 m

Project KUNING

Elevation R+102.570 m

Type of Drilling ROTARY

Date 10/01/03-11/01/03

Driller

Remarks

P : Standard Penetration Test

UOP : Fixed-Piston Undisturbed Sampling

UOS : Open-Drive Undisturbed Sampling

VST : Vane Shear Test

Scale in m	Elevation in m	Depth in m	Thickness in m	Legend	Type of Soil	Colour	Relative Density or Consistency	General Remarks	Sampling		Standard Penetration Test					
									Depth in m	Sample No.	Nc Value Blows/30cm	Blows Per Each 10 cm	N - Value			
											10	20	30	40	50	
1	101.77	0.80	0.80		Sand	Grey	Very Loose	With some hardcore and debris								
2					Silty Clay	Yellowish Brown to Grey	Soft	With lots of sand and gravel and broken pieces of stone concrete chip	1.65	P-1	4	1	1	2		
3	99.47	3.15	2.35		Organic Clay	Dark Grey	Soft	with decomposed woods	3.15	P-2	2	1	0	1		
4									4.50							
5	97.27	5.30	2.15		Silty Clay	Grey	Very Soft	Marine clay. With shell fragments and fine sand at bottom.	5.30	UOS-1		Recovery = 66.7% cm				
6									6.00							
7									6.80	UOP-1		Recovery = 40.7% cm				
8	94.27	8.30	3.00		Silty Clay	Grey to Light Grey	Soft	With fine sand.	7.50	VST-1		Sw = 17.2 %				
9									9.00							
10	92.07	10.50	2.20		Clayey Sand	Light Grey	Loose to Medium Dense	Old Alluvium. Sand is fine to coarse.	9.80	UOS-2		Recovery = 80.8% cm				
11									10.65	P-3	14	3	5	6		
12									12.15	P-4	12	2	4	6		
13									13.65	P-5	7	2	2	3		
14									15.15	P-6	24	6	9	9		
15									17.15	P-7	35	10	12	13		
16	85.57	17.00	6.50		Silty Clay	Yellowish Brown	Hard		17.45							
17									19.15	P-8	22	6	7	9		
18	83.57	19.00	2.00		Clayey Sil	Greenish Grey	Very Stiff	With sand. Sand is fine to coarse.	19.45							
19									21.15	P-9	39	12	14	13		
20	81.57	21.00	2.00		Silty Clay	Greenish Grey	Hard	With pockets of sand.	21.45							
21									23.15	P-10	25	6	8	11		
22	79.57	23.00	2.00		Silty Clay	Light Grey	Very Stiff	Weathered with reddish brown and some small silt pockets.	23.45							
23									25.10	P-11	100/15	43	57/5			100 BLOWS/15cm
24	77.57	25.00	2.00		Silty Sand	Greenish Grey	Very Dense	Sand is fine to coarse.	25.25							
25									27.15	P-12	100/28	25	35	40/8		100 BLOWS/28cm
26	75.14	27.43	2.43						27.43							
27																
28																
29																
30																
31																

Prepared by: [Signature]

Checked by: [Signature]

Approved by: [Signature]

K-4

Appendix E

Soil Investigation Borehole Logs

FIG DRILLING LOG

Project No. 516-31 Project CDM Type of Drilling ROTARY

Hole Number K-5 (PAGE 1 of 2) Date 30/12/02-06/01/03

Water Table 0.180 m. Elevation R+104.854 m. Driller L. ...

Remarks
 P: Standard Penetration Test
 UDF: Fired-Piston Undisturbed Sampling
 UDS: Open-Drive Undisturbed Sampling
 VST: Vane Shear Test

Scale in m	Elevation in m	Depth in m	Thickness in m	Legend	Type of Soil	Colour	Relative Density or Consistency	General Remarks	Sampling		Standard Penetration Test							
									Depth in m	Sample No.	N-Value Blows/30cm	Blows Per Each 10 cm	N - Value 10 20 30 40 50					
1	104.35	0.30	0.50		Sand	Grey	(Loose)	Sand is fine to coarse										
2					Silty Clay	Yellowish Brown	Medium Stiff to Stiff	Mixed with stones, gravel, concrete chips and lean concrete in this layer (7%)	1.00	UDS-1	Recovery = 10/37 cm							
3	101.65	3.00	2.50		Peaty Clay	Grey and Dark Brown	Very Soft	With lots of decomposed woods and fine sand. With organic matters	1.15	P-13	0 0 0 0	0	0	0	0	0	0	SELF PENETRATION BY HAMMER
4									1.30	UDS-2	Recovery = 50/80 cm							
5	99.55	5.30	2.30						1.50	UDS-3	Recovery = 30/85 cm							
6					Silty Clay	Yellowish Brown	Very Soft to Soft	With lots of fine sand	1.65	UDP-1	Recovery = 85/84 cm							
7	97.35	7.30	2.20						1.75	UDP-2	Recovery = 80/85 cm							
8	96.35	8.50	1.00		Silty Clay	Reddish Brown and Light Grey	Medium Stiff	Mottled with yellowish	1.70	UDP-3	Recovery = 40/10 cm							
9					Clayey Sand	Grey	Very Loose	Sand is fine	1.85	P-22	5 1 2 2							
10	94.65	10.20	1.70						1.85	P-3	1 1 0 0							
11					Silty Clay	Yellowish Brown	Soft to Medium Stiff	Mottled with grey, significantly mixed with fine sand	1.95	P-4	1 1 0 0							
12	92.85	12.00	1.80						10.35	P-5	5 1 2 2							
13	91.85	13.00	1.00		Clayey Sand	Reddish Brown	Loose	Sand is fine	10.65									
14	90.85	14.00	1.00		Silty Clay	Yellowish Brown	Medium Stiff	With lots of fine sand	11.50	UDS-4	Recovery = 50/80 cm							
15					Silty Sand	Yellowish Grey to Greenish Grey	Loose to Medium Dense	Old Alluvium, Sand is fine to coarse and occurred some low clay inclusion	12.00	P-6	7 2 2 3							
16									12.45	UDS-5	Recovery = 50/80 cm							
17									13.00									
18	86.35	18.50	4.50		Silty Clay	Greenish Grey	Hard	With fine sand	13.60	P-7	8 2 3 3							
19									14.15	P-8	7 2 2 3							
20	84.35	20.50	7.00		Clayey Sand	Greenish Grey	Medium Dense to Dense	Sand is fine at top portion and coarse at bottom	15.15	P-9	7 2 2 3							
21									16.85	P-10	18 5 6 7							
22									18.15	P-11	33 10 12 11							
23	81.05	23.80	3.30		Silty Clay	Greenish Grey	Hard	With some fine sand	18.65	P-12	12 4 4 4							
24									21.15	P-13	30 8 10 12							
25									21.45	P-14	38 10 13 15							
26									22.65	P-15	150/24 78 48 21/4							150 B.C.MS/24cm
27									22.95									
28									24.15	P-16	51 15 15 21							51 B.C.MS/30cm
29	75.85	29.00	5.20		Silty Sand	Greenish Grey	Very Dense	Sand is fine to medium	27.15	P-17	100 25 32 43							100 B.C.MS/30cm
30									29.15	P-18	100/3 100/2							100 B.C.MS/3cm
31	73.77	31.08	2.08						30.05									

Prepared by: AP Checked by: AP Approved by: AP

K-5 (1)

Appendix E

Soil Investigation Borehole Logs

FIG DRILLING LOG

Project No. _____

Hole Number K-5 (PAGE 2 of 2)

Water Table Q. -1.80 m

Project _____

Elevation RL+104.854 m

Type of Drilling ROTARY

Date 30/12/01-06/01/03

Driller _____

Remarks

P : Standard Penetration Test
UUP : Fixed-Piston Undisturbed Sampling
UOS : Open-Drive Undisturbed Sampling
VSI : Vane Shear Test

Scale in m	Elevation in m	Depth in m	Thickness in m	Legend	Type of Soil	Colour	Relative Density or Consistency	General Remarks	Sampling		Standard Penetration Test				
									Depth in m	Sample No.	N-Value Blows/30cm	Blows Per Each 10 cm	N - Value		
											10	20	30	40	50
					Silty Sand	Greenish Grey	Very Dense	Sand is fine to medium	30.65 30.08	P-18	100/1	100/1			100 Blows/30cm
31	21.77	31.08	2.08	x x											
32					-END OF DRILLING-										
33															
34															
35															
36															
37															
38															
39															
40															
41															
42															
43															
44															
45															
46															
47															
48															
49															
50															
51															
52															
53															
54															
55															
56															
57															
58															
59															
60															
61															

Prepared by: _____

Checked by: _____

Approved by: _____

K-5 (2)

Appendix E

Soil Investigation Borehole Logs

FIG DRILLING LOG

Project No. _____		Project _____		Type of Drilling <u>ROTARY</u>		Remarks P : Standard Penetration Test UOP : Fixed-Piston Undisturbed Sampling UOS : Open-Drive Undisturbed Sampling YST : Vane Shear Test
Hole Number <u>K-6 (PAGE 1 of 1)</u>		C _____		Date <u>04/01/03 - 07/01/03</u>		
Water Table <u>0.210</u> m.		Elevation <u>R+104.20</u> m.		Driller _____		
Scale in m _____		_____		_____		

Scale in m	Elevation in m	Depth in m	Thickness in m	Legend	Type of Soil	Colour	Relative Density or Consistency	General Remarks	Sampling		Standard Penetration Test					
									Depth in m	Sample No.	N-Value Blows/30cm	Blows Per Each 10 cm	N - Value			
											10	20	30	40	50	
	103.75	0.50	0.50		Sand	Grey	(Loose)	Sand is fine to coarse.								
1					Silty Clay	Yellowish Brown	Soft	Mixed with stone concrete chips and debris and organic molluscs. (72)								
2									1.85	P-0.8	2	0	1	1		
3	101.25	3.00	2.50		Peaty Clay	Dark Grey and Brown	Very Soft to Soft	With bits of decomposed woods fine sand and some organic molluscs.	3.00							
4									3.80	UOS-1	Recovery = 54/76 cm					
5									4.80	UOS-2	Recovery = 50/76 cm					
6									5.00	UOS-3	Recovery = 46/76 cm					
7	97.55	6.70	3.70		Silty Clay	Light Yellowish Brown	Very Soft to Soft	With lots of fine to coarse sand and fine gravel.	6.85	UOP-1	Recovery = 65/76 cm					
8	96.25	8.00	1.30		Silty Clay	Reddish Brown	Very Soft	Mottled with grey.	7.85	UOP-2	Recovery = 40/76 cm					
9	94.75	9.50	1.50						8.15	P-1.8	0	0	0	0		
10					Clayey Sand	Whitish Grey to Yellowish Brown	Very Loose	Sand is fine to coarse and occurred colour changes.	8.50	UOS-4	Recovery = 50/76 cm					
11	92.75	11.50	2.00						10.00	P-2.8	1	0	0	1		
12					Clayey Sand	Whitish Grey to Yellowish Brown	Loose	Old Alluvium. Sand is fine to coarse and occurred colour changes.	11.65	P-3.8	9	2	3	4		
13	91.25	13.00	1.50		Silty Clay	Grey	Very Still	Mottled with reddish brown.	13.15	P-4.8	16	4	5	7		
14	89.75	14.50	1.50		Clayey Silt	Greenish Grey	Still	With fine sand.	14.65	P-5.8	10	3	3	4		
15	88.25	16.00	1.50		Silty Sand	Whitish Grey	Medium Dense	Sand is coarse to medium.	16.15	P-6.8	11	4	3	4		
17	86.75	17.50	1.50		Clayey Sand	Whitish Grey	Medium Dense to Dense	Sand is coarse to medium.	17.65	P-7.8	10	1	4	5		
18									19.15	P-8.8	22	6	7	9		
19									20.65	P-9.8	28	9	9	10		
20									22.65	P-10.8	38	10	12	16		
21									24.65	P-11.8	34	8	11	15		
22									26.65	P-12.8	100/16	41	59/8			
23	77.75	26.50	8.00		Clayey Silt	Whitish Grey	Very Dense		28.65	P-13.8	72	26/3				
24																
25																
26																
27	75.47	28.78	2.28													
28																
29																
30																
31																

-END OF DRILLING-

Prepared by: _____

Checked by: _____

Approved by: _____