DATA EXCHANGE IN GEOTECHNICAL ENGINEERING

by

Nazila Mokarram

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(CIVIL ENGINEERING)

December 2010

Dedication

to

my parents

Nourintaj Pashaeirad, Ali Mokarram

my sister

Nazanin Mokarram

and my husband

Soheil Seiqali

Acknowledgments

I would like to express my gratitude to everyone who has helped me with this study.

Particularly, my sincere gratitude goes to my advisor, Prof. Jean-Pierre Bardet, for his guidance and support during the entire course of my doctoral program. Special thanks are extended to my committee members, Prof. Geoffrey R. Martin, Prof. John Wilson, and Prof. L. Carter Wellford, for their constructive criticism and advices. Special appreciation also goes to Dr. Behnam Hushmand, for his encouragement and support.

I would like to recognize the special contribution and many helpful suggestions from my colleagues in the geotechnical engineering group Mr. Amir Zand and Dr. Fang Liu, who generously shared with me their knowledge and experiences. My gratitude also extends to my other colleagues Dr. Jinaping Hu, Dr. Rana Alfares, and Dr. Yangsoo Lee for building a friendly atmosphere at University of Southern California.

Finally, I appreciate my family's continuing understanding and support, especially, my best friend and husband, Soheil Seiqali, who has always been present to help me with my biggest challenges.

Table of Contents

Appendices

List of Tables

List of Figures

Abstract

Although geotechnical information is obtained from rather costly drilling and laboratory operations, they are poorly documented and curated, due to lack of adoptable standards for data handling.

In this PhD dissertation proposal, first geotechnical community and data will be reviewed. Then, evolution of geotechnical data release within the community is studied. Based on the advantages and shortcomings of past efforts and the community specific needs, a requirement list for data exchange format is created. After detail analysis of AGS format for geotechnical data, an eXtensible Markup Language (XML)-based data organization is proposed. The eXtensible AGS (XAGS) data format is discussed in detail. XAGS is validated by examples of data generation and modification, data validation, data exchange, and archive and distribution via World Wide Web. To show the improvements of the new data exchange format over the previous formats, the proposed data format is evaluated by comparing its capabilities with the requirement list developed early in the study. At the end, a metadata model has been developed for documenting the data sets generated by experiment and simulation processes. The metadata model has an object-oriented structure developed using web ontology tools and expressible in XML schemas. The usefulness of the metadata model is demonstrated by generating automatic data reports and exchanging data sets with complete documentations. The metadata model for experimental research can be used as a guideline to develop metadata model for geotechnical information that are not well-documented.

Chapter 1. Introduction

1.1. Exchange of Geotechnical Information

Adequate site investigation is essential for properly executing construction of any infrastructure, commercial, residential and industrial project. The site investigation procedure whtever the type of project and nature of the site normally includes reports on boreholes, in situ and laboratory tests, and site stratigraphy. These reports reveal critically important information such as the geometries and properties of soil profiles. Site investigation usually costs within the range of 0.1% to 2% of the total project budget (Craig, 2001). An example of very large projects, the Central Artery/Tunnel project in Boston (massDOT, 2010), generated an excess of 2963 geotechnical boreholes, which were compiled with great efforts by Baise and Brankman (2004) for characterizing the liquefaction susceptibility of soil deposits in the Boston area. The total cost of the project turned out to be about $14.625 billion (Road Traffic, 2010). Assuming a modest 1% cost for geotechnical investigation for such a geotechnically challenging project, the cost of geotechnical site investigation alone amounted to about $146 million.

Presently, these valuable geotechnical data are stored in various forms and formats. Geotechnical experts face problems in comprehending, and extracting the needed information from data files on a daily basis, let alone manipulating and passing data to other team members and entities. This data incompatibility impedes project optimization and makes it difficult to reuse data already available from previous projects (Zimmermann et. al., 2006). Even though, geotechnical site investigations are costly and time consuming, their data are poorly archived and data retrieval and exchange is very difficult and sometimes even impossible. Therefore, there is an urgent need to develop a well-structured and practical data format for efficient exchange of data.

1.2. Research Objectives

The objectives of this research are to propose a versatile exchange format for geotechnical information, which: (1) provides data exchange within the geotechnical community using the Internet, (2) can be viewed by widely accessible free or commercial programs, such as Internet browsers and interactive maps, (3) can be easily generated and modified in free software programs, (4) utilizes a terminology commonly accepted in the geotechnical community but (5) can easily be implemented. This dissertation is also organized to be used as a guideline for the geotechnical community to develop exchange formats. However, it should be noted that this research is not intended to define standards; such standards have to be defined by a collective and collaborative effort in the geotechnical community.

1.3. Organization of Dissertation

Following the introduction, chapter two develops requirements for exchanging geotechnical information. First, it defines the geotechnical community in terms of commonly involved parties and their roles in geotechnical projects. Then, it defines geotechnical information and identifies boreholes with in situ tests and laboratory tests as the most common methods of data acquisition in geotechnical projects. In this chapter, common in situ and laboratory tests are selected as the scope of study. Additionally, it defines and distinguishes exchange data from other types of data. Later, it presents short reviews of recent efforts in geotechnical information handling. Two of these efforts (AGS and DIGGS) are scrutinized in more detail. Based on the conducted reviews, a set of requirements for the geotechnical data which should be satisfied by an exchange format is presented.

Chapter three illustrates a methodology to utilize advanced information technologies and domain knowledge expertise to effectively define an XML-based data organization. This chapter depicts transformation of relational information to XML format, principles used and the path traveled to reach the proposed format (XAGS 2.0). The study then implements XAGS 2.0 by detail discussion of project, borehole, in situ tests, sample, and laboratory tests. Chapter three also addresses the domain-specific decisions made on data structure for some of the geotechnical information components considering practicality. It also describes why the proposed format seems to be the most suitable one.

Chapter four illustrates how to use and exchange XAGS data. The proposed format is used to display borehole data in spatial and non-spatial formats. Conversion from and to other formats is addressed as well. The chapter demonstrates XAGS data generation and modification with XML editors and spreadsheets. Then it depicts scenarios of data exchange within the geotechnical community and proposes an information system for archive and exchange of XAGS data via World Wide Web. At the end, XAGS is assessed and evaluated against the list of data exchange requirements introduced in chapter two.

Chapter five presents one possible future direction for exchanging data. As data exchange gets more and more comprehensive, not all data will be defined with standard procedure and routines. Therefore, additional data needs to be exchanged for understanding of how the data is obtained. This is the realm of metadata. Metadata modeling helps understanding of the data generation process. In this case, metadata modeling is an additional layer over exchange data for documenting the process of data generation in geotechnical projects.

Finally, Chapter six summarizes the entire research, highlights the original contributions and lists possible future considerations for XAGS.

Chapter 2. Exchange of Geotechnical Information

As many geotechnical projects are becoming larger and more complex, they produce an increasingly large volume of heterogeneous geotechnical information. This requires users to spend more time and resources to locate, properly utilize, and process data for engineering analysis, design, and simulation.

This chapter reviews the geotechnical community and its involved entities. It also discusses different geotechnical data and their formats. Data exchange will be defined and data exchange in the geotechnical community will be reviewed. As will be shown, the data generated, processed and stored in the geotechnical community are broad and diverse. However, this research will focus on the most commonly used geotechnical data. These data will be defined in more detail later in this chapter.

Past efforts to improve geotechnical information release are also documented in this chapter. Based on the advantages and shortcomings of them, a comprehensive list of requirements for a successful geotechnical information exchange format is introduced.

2.1. Geotechnical Community and Information

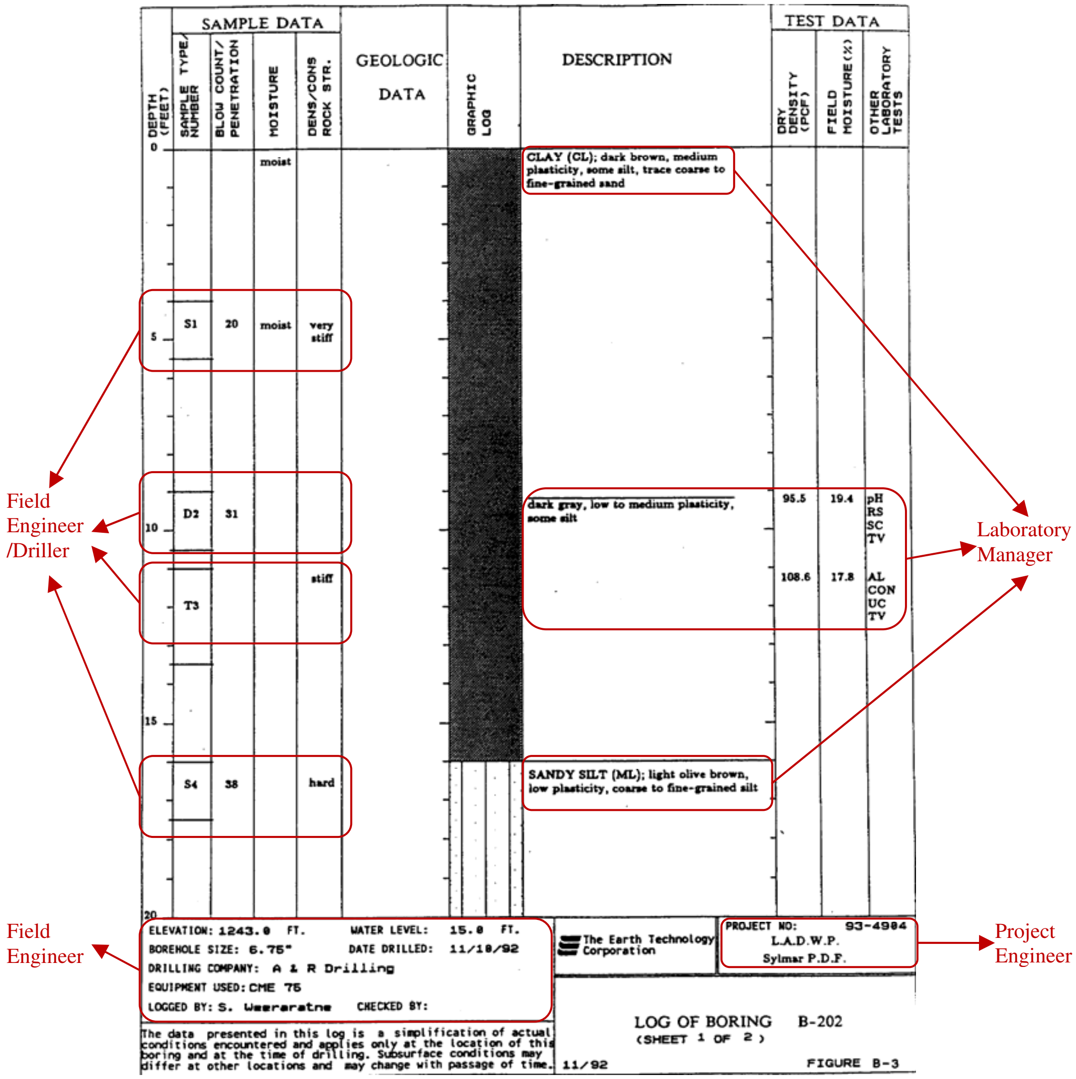
It is of great importance to first define the involved entities in the geotechnical community, different forms of geotechnical information and the current state of information exchange and its problems, to understand the necessity of this study.

2.1.1. Geotechnical Community

Geotechnical community can be generally divided into two groups:

Engineering practice-oriented community

Engineering practice-oriented community, usually concentrated in small to large firms, mainly uses standard and well defined test procedures. Field engineers, drillers, laboratory technicians, geologists, project engineers and project managers can all be members of a geotechnical project team. Some operations of these team members are standardized by their respective professional organizations such as ASCE (American Society of Civil Engineers) and ASTM (American Society for Testing and Materials). Definition of each involved team member can be found in main geotechnical textbooks such as Craig (2001). Figure 2.1 shows an example of a borehole log as an instance of final result of a practice-oriented project and shows how different members of the project team contribute to the production of the final borehole log.



moist

S1 20 moist very stiff

D2 31

T3 stiff

S4 38 hard

CLAY (CL); dark brown, medium plasticity, some silt, trace coarse to fine-grained sand

dark gray, low to medium plasticity, some silt

Dry density (PCF)	Field moisture (%)	Other laboratory tests
95.5	19.4	pH RS SC TV
108.6	17.8	AL CON UC TV

SANDY SILT (ML); light olive brown, low plasticity, coarse to fine-grained silt

ELEVATION: 1243.0 FT. WATER LEVEL: 15.0 FT.
BOREHOLE SIZE: 6.75" DATE DRILLED: 11/10/92
DRILLING COMPANY: A & R Drilling
EQUIPMENT USED: CME 75
LOGGED BY: S. Weeraratne CHECKED BY:

The Earth Technology Corporation

PROJECT NO: 93-4984
L.A.D.W.P.
Sylmar P.D.F.

The data presented in this log is a simplification of actual conditions encountered and applies only at the location of this boring and at the time of drilling. Subsurface conditions may differ at other locations and may change with passage of time.

LOG OF BORING B-202
(SHEET 1 OF 2)
11/92 FIGURE B-3

Figure 2.1 Geotechnical community members involved in compiling a borehole log (Borehole log image from Bardet et al., 1999)

Engineering research-oriented community

Engineering research community, generally concentrated in universities and research centers, develops new procedures and tests. As a result, the procedures used are not well defined and can evolve as the project develops over time. Research is usually very detail oriented and documents data as well as how the data is obtained, or in other words data about data. This is commonly known as metadata. It is possible that, once a research procedure is successfully defined, it becomes part of the general practice in the relevant field. NEES (George E. Brown, Jr. Network for Earthquake Engineering Simulation) is an example of a research-oriented organization conducting collaborative experimental research. Their tests rarely follow established procedures.

It is evident that the boundaries between these two groups are not rigid and they overlap. The public sector, such as state, federal and municipal agencies, and some private companies such as larger size firms can be involved in both practice-oriented and research-oriented projects. Additionally, other disciplines like geophysicists and geologists can also get involved in a geotechnical project. This mostly happens in large projects overlapping with other specialties such as geoenvironmental testing.

2.1.2. Geotechnical Data

Geotechnical data is extensive. It morphs, mutates, aggregates and expands and is presented in many different types and formats. To display this fact, data types used in a finite element analysis of a tunnel and their formats are shown in Figure 2.2. In situ and laboratory tests will provide the site material properties. These data are usually compiled in different formats such as spreadsheet files, geotechnical reports, Access databases and other data repositories. By combining test results and spatial locations layer stratigraphy is defined and displayed in form of borehole logs or cross sections. When the geometry of problem is defined, finite element types and properties are selected based on this layer stratigraphy. The finite element analysis will produce charts, tables, and graphs of strain and stress contours.

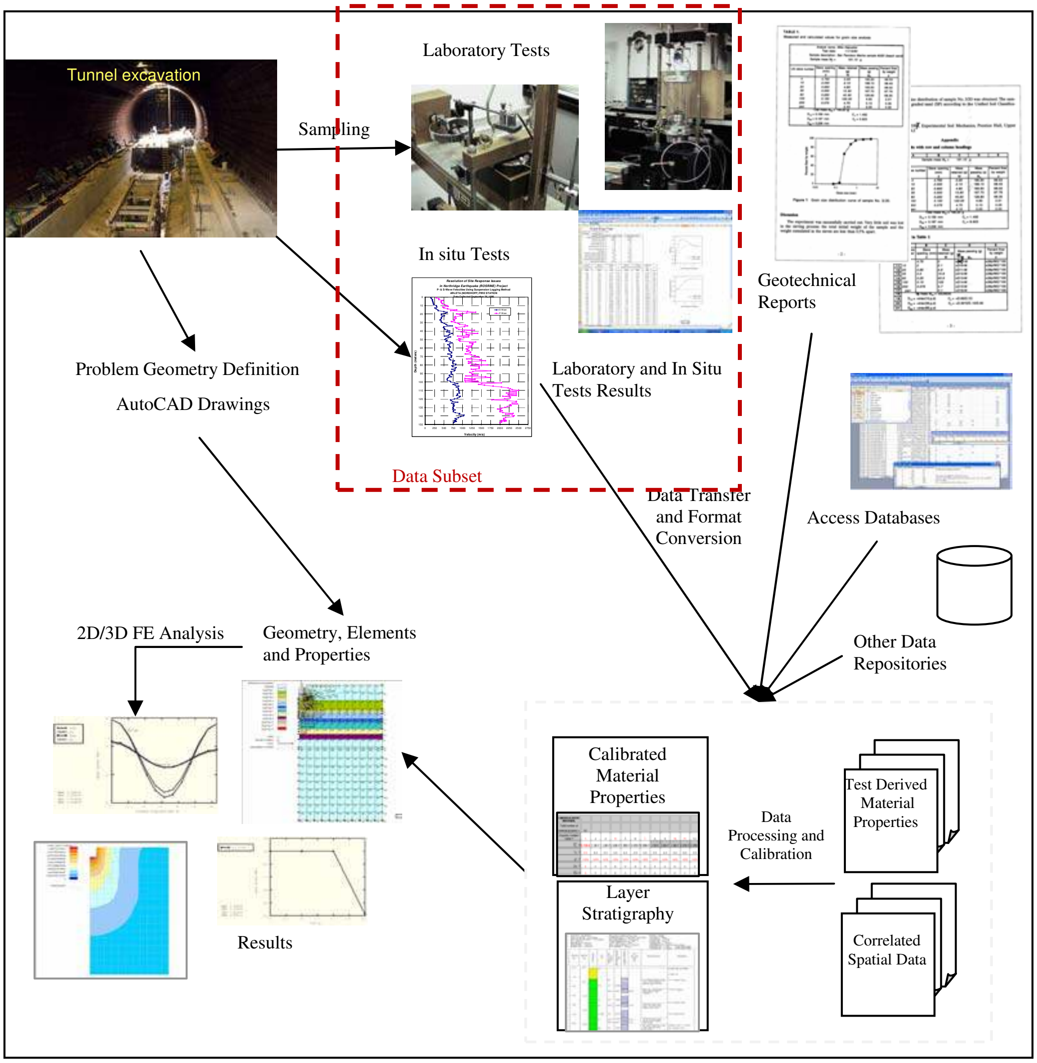

Tunnel excavation
Sampling
Laboratory Tests
In situ Tests
Laboratory and In Situ Tests Results
Geotechnical Reports
Problem Geometry Definition
AutoCAD Drawings
Data Subset
Data Transfer and Format Conversion
Access Databases
Other Data Repositories
2D/3D FE Analysis
Geometry, Elements and Properties
Calibrated Material Properties
Data Processing and Calibration
Test Derived Material Properties
Layer Stratigraphy
Correlated Spatial Data
Results

Figure 2.2 Different types of geotechnical data produced and transformed within a sample geotechnical project. Not all data types can be easily exchanged.

This proposed scenario is just one sample scenario out of many possible ones. Other projects might be analyzed with foundation design, retaining walls, liquefaction analysis, road design, water proofing of buried structures or many other objectives instead of the finite element analysis used as an example here. However, the common underlying fact is that the data forms could totally vary from project to project based on the needs of each specific project. Land surveying, remote sensing, cartography, Geographic Information Systems (GIS), Global Navigation Satellite Systems such as GPS, photogrammetry, and topography are some of the other possible data types in geotechnical engineering projects. However, it can be seen that for almost all the civil engineering projects, the effort to complete the geotechnical information is mainly concentrated on obtaining the needed data from boreholes. The borehole logs commonly include the information gathered during the site investigation, such as in situ test results and stratigraphy as well as the laboratory tests performed on the collected samples. Additionally, this example demonstrates that exchange of data is almost inevitable for any project. Different data will be produced in different stages by different entities and need to be transferred to the next entity for some additional processing or use.

Members of geotechnical community might use geotechnical information in different ways, as shown in Table 2.1. Some members only contribute to generating a specific portion of data. For instance, a driller reports number of SPT blow counts at each depth. Others, such as a field engineer, might assemble the data generated in the field and plot the data location on maps, as well as archive the field data. On the other hand, others such as lab technician might be assembling and archiving data as well as generating them by performing the lab tests. Then, the project engineer might assemble data received from different team members and utilize the data set for engineering calculations. The project engineer usually plots, tabulates and generates graphs for the final report as well. Usually, after completion of project the complete data set will become available to the client (data owner) and other data users (e.g. researchers) for archival and utilization of complete data set. Data users usually plot, tabulate and graph data in their own preferred format during data utilization. The type of ownership might also impact the use of data as well; while public agencies main responsibility is generally overseeing the projects conducted in their jurisdiction with reviewing the data set, approval and archival in their database, private agencies might additionally be involved in the assembly of data sets received from their sub contractors or other collaborators as well.

Table 2.1 A sample matrix of practice-oriented geotechnical entities involved in a site investigation project and their most probable data usage types

Data Usage Types	Project Engineer	Field Engineer	Driller	Lab Technician	Data Owner (Client)	Data User (researcher analyst)	Public Agencies	Private Agencies
Generating Data			▨	▨				
Assembling Data	▨	▨		▨				▨
Utilizing Data for Design/ Calculation	▨				▨	▨	▨	▨
Archiving Data	▨	▨		▨	▨	▨	▨	▨
Reporting Data	▨	▨				▨		

2.2. Data Exchange in Geotechnical Community

A data format is generally used for data storage and archiving on a storage medium such as DataBase Management System (DBMS). A data exchange format, however, needs to satisfy some more criteria to be proper for exchange of information between different entities and might or might not be used for data storage.

Teams and team members interact with each other, produce, modify, and exchange data with one another constantly. Geotechnical community members use data in their own space. For instance, a CPT driller (driller A) uses A to D convertor to transform analog data measured from CPT cone sensors into voltage and then convert voltage to tip resistance, skin friction and pore water pressure measurements at each depth. These data are tabulated in a specific format used only by the specific drilling company and be sent to the geotechnical engineer as demonstrated in Figure 2.3. Geotechnical engineer (engineer A) then might import this data in a spreadsheet program for settlement, liquefaction and other calculations. A new data set is made by data analysis that has a different form from the CPT results. This new data set then might be exchanged with another geotechnical engineer (engineer B) for further processing. However, if the original CPT data set is passed to the engineer B without the personal communications with driller A for data set explanation, there is a high chance of loss of information in the process of transferring data. Therefore, as shown in Figure 2.3, problems in data comprehension are possible without personal communications between team members when there is a lack of exchange format. However, there is still no common standard used for geotechnical data exchange in practice, and data is exchanged in different digital and sometimes even hardcopy formats. A data exchange format can improve efficiency and data quality and eventually reduce costs.

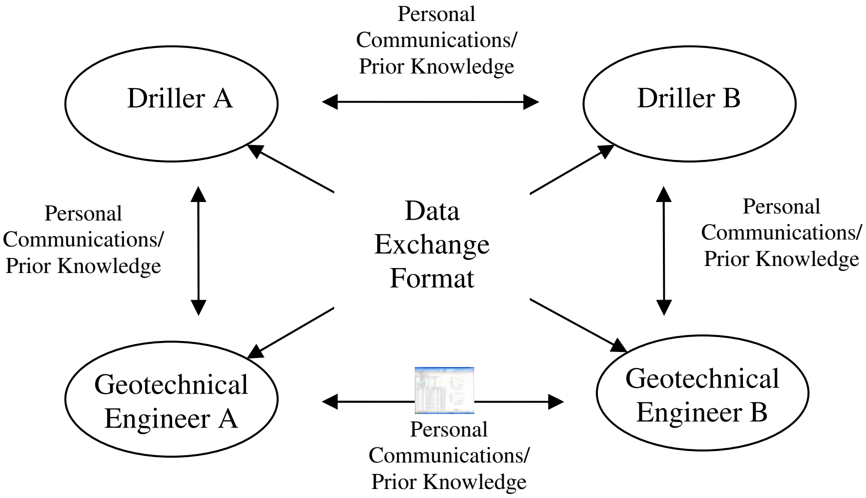

Personal Communications/ Prior Knowledge
Driller A
Driller B
Personal Communications/ Prior Knowledge
Data Exchange Format
Personal Communications/ Prior Knowledge
Geotechnical Engineer A
Geotechnical Engineer B
Personal Communications/ Prior Knowledge

Figure 2.3 Possible problem in data comprehension, without personal communications or prior knowledge between team members by lack of uniform geotechnical exchange format

Figure 2.4a represents the data flow diagram (Yourdon, 2006) for geotechnical project demonstrated in Figure 2.2, which represent the main steps in utilization of information from production to conclusion. For a particular engineering problem (e.g., finite element analysis of a tunnel excavation), (1) the site is first characterized by a series of in situ and laboratory tests (processes 1, 2 and 3); (2) the test results are used to define the geometry of soil deposits and their material properties (process 4); and (3) defined geometry and material properties are then assembled into a comprehensive computer model (process 5), which assists engineers in calculating various forces and displacements that affect the engineering problem (process 6).

The current data flow in geotechnical engineering in Figure 2.4b illustrates how complicated it is to relate data from laboratory tests to data input in numerical analysis. The reprocessing of geotechnical information requires large person-hour resources, and impedes the utilization of geotechnical information in analysis, design and decision making processes. This data flow has to be simplified for efficient uses of data in realistic computer simulations. Figure 2.4c depicts how a traditional data flow can be improved by handling of information by a proper exchange format; data reprocessing can be accelerated using machine-to-machine communication, and data from original tests can be closely linked to simulation data and analysis results. This will eliminate extra layers of data processing; and reduce the chances of different data interpretations. The benefits of this modern data flow, which may be not apparent for small projects, become obvious for large projects whose data originate from various heterogeneous sources and require time-consuming reprocessing.

Although, it is not simple to identify what a proper exchange format is, after studying the past efforts for geotechnical information release, a list of requirements for the proper geotechnical data exchange will be developed and presented at the end of this chapter.

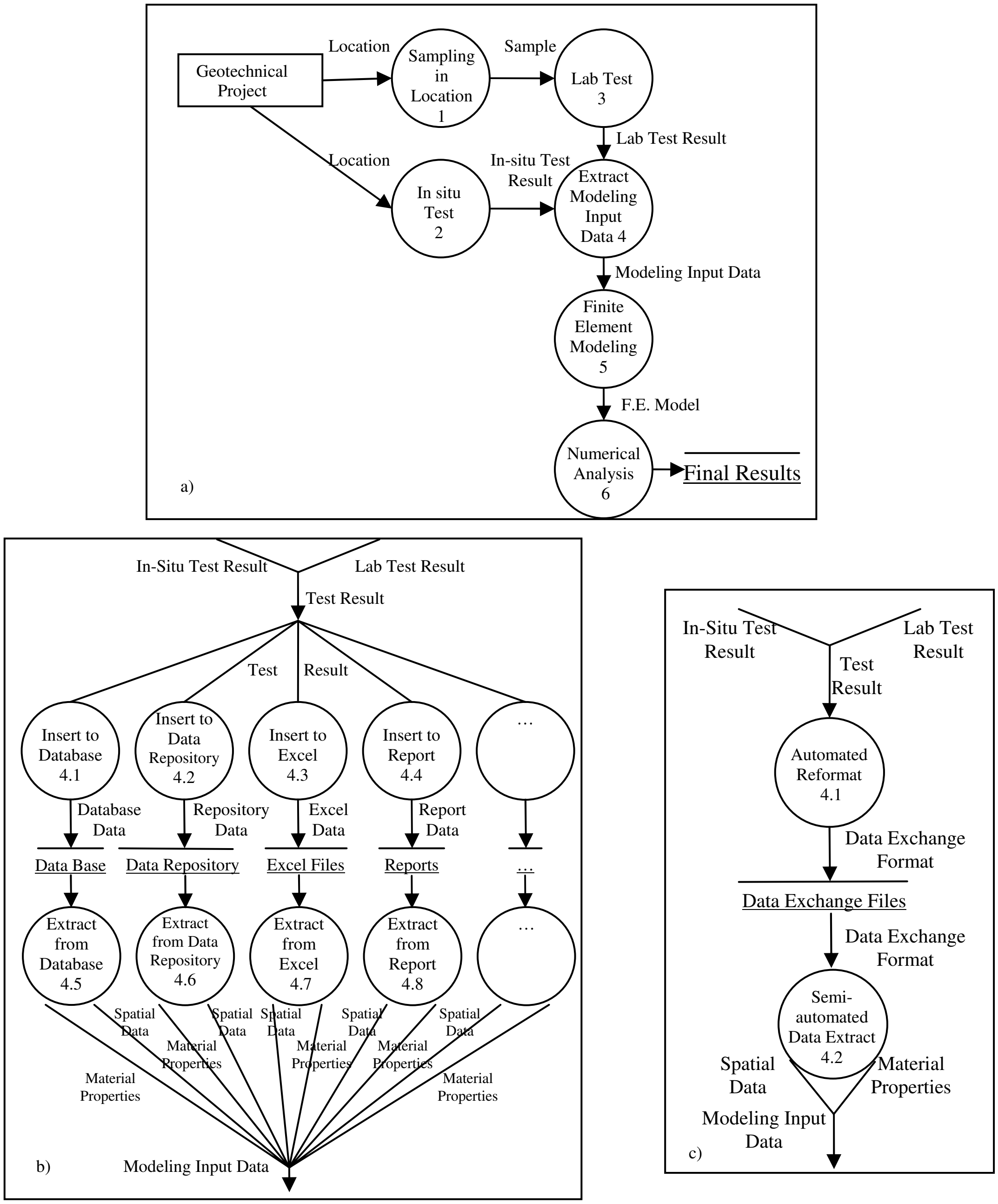


Figure 2.4 Data flow diagrams (DFD) of current state and future usage of data in a sample geotechnical engineering project. a) A sample data flow diagram of geotechnical engineering projects. b) Second level DFD for current state of extracting input data from test results. c) Second level DFD for future vision of extracting input data from test results.

2.3. Scope of This Study

The scope of this study is restricted to the most commonly used and principle types of data in geotechnical engineering that almost all geotechnical tasks rest upon. Only results of field and laboratory investigation will be discussed as a subset of geotechnical data. Study of information retained from boreholes will be a good representative of the most common data types and information acquired, exchanged, compiled, presented and archived for further use in practice-oriented geotechnical projects. Moreover, its incorporation into a more efficient system will have a high rate of return for the effort. Research data sets will be out of our scope, but will be addressed in the chapter on future direction.

Boreholes are currently the most popular and economical means to obtain subsurface information. Borings are drilled as vertical, inclined or horizontal holes into earth materials. The primary purpose is measuring the overburden or rock material properties present and thereby permitting the determination of the stratigraphy and/or engineering properties of the soil and groundwater conditions (Craig, 2001). Boreholes may include: cone penetration tests (CPT), standard penetration tests (SPT), pressure meter, shear vane, borehole permeability tests, or seismic borehole tests, such as geophysical suspension logging. For this study SPT and CPT boreholes will be considered as the most common types and are defined here. Definition of the remainder subsurface field investigations can be found at Das (1994).

CPT is an in situ geotechnical analysis that consists of measuring different soil parameters while driving a cone and friction sleeve vertically into the ground (D3441-98 ASTM, 2005a). The CPT directly measures the resistance of a cone during penetration to the ground. It also determines the lateral friction on a given length of the friction sleeve and pore water pressure in the ground, around the cone, during driving. Using the measured parameters, the CPT permits the appraisal of (1) the succession of different stratigraphic layers and their geophysical characteristics, (2) the homogeneity of a layer or the presence of anomalies, (3) geotechnical characteristics of the soil. Figure 2.5 shows two different ways to present CPT results.

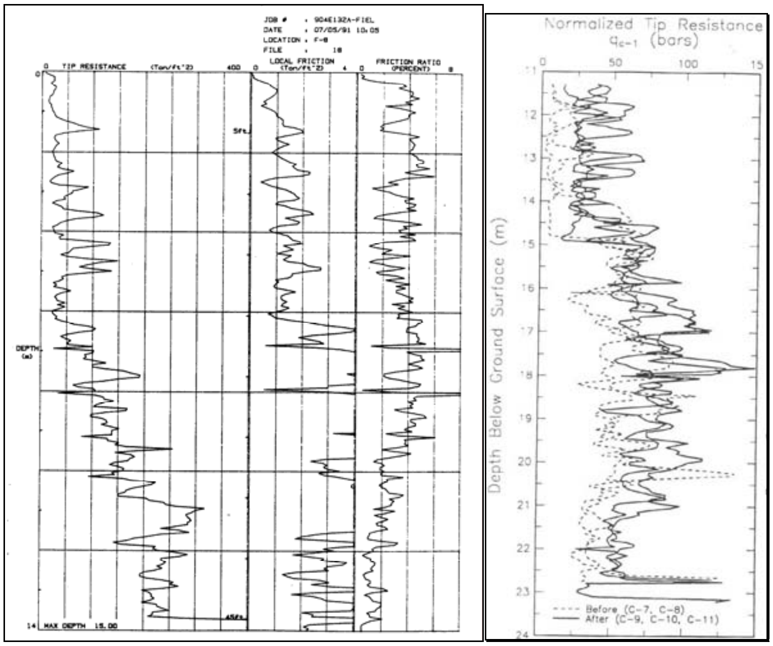


Figure 2.5 Two different graphical presentations of CPT data. Left: graphical presentations of CPT results in three graphs, right: graphical presentations of CPT results in one combined graphs (Bardet et al., 1999).

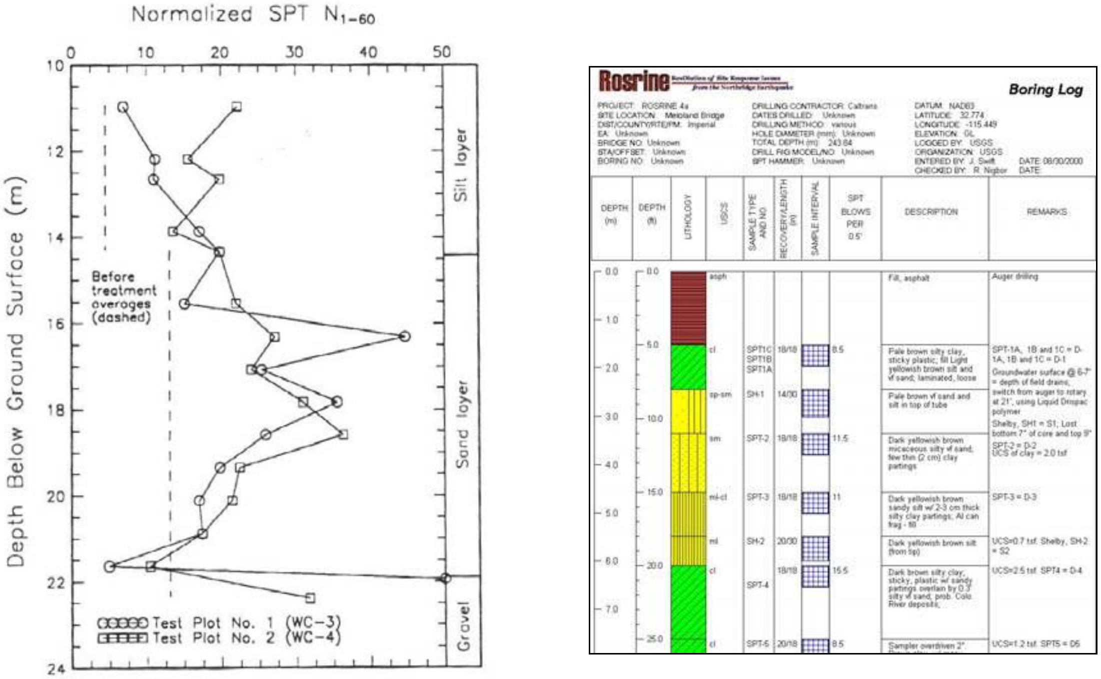
SPT is also an in situ geotechnical test that provides conventional soil characteristics and physical soil samples (D1586-99, ASTM, 2005b). The resistance at dynamic penetration of hardened steel, split spoon sampler is determined by driving into the soil using a 140-lb hammer falling 30 inches, and counting the number of blows required to penetrate from a depth interval of 6 to 18 inches. CPT allows continuous recording of soil changes with depth, whereas SPT only records major changes at discrete steps. However, SPT allows soil sampling for laboratory testing (Das, 1994). The results of this test are used to establish the relationship between (1) the resistance of the soil at penetration and (2) the geotechnical and geophysical characteristics and variability of the soil. The SPT is only applicable to fine or gravelly soils with a grain size less than 20 mm, and the maximum depth to which the test can be carried out is about 50 m (Afnor, 1991). Figure 2.6 illustrates two different ways to present SPT results.

A wide variety of laboratory tests can be performed on soil samples collected at different depths to measure soil properties. Some soil properties are intrinsic to the composition of the soil matrix and are not affected by sample disturbance, while other properties depend on the structure of the soil as well as its composition, and can only be effectively tested on relatively undisturbed samples (Bardet, 1997).

In this research, CPT and SPT, and some of the more commonly performed laboratory tests listed later are used for developing an exchange format. The principles used for exchange of these data, can be easily extended to include other geotechnical information.

2.4. Evolution of Geotechnical Information Release

At this moment, there is no standard or commonly accepted information release format for exchange, storage or display of geotechnical boreholes, or established methods for archiving and distributing borehole data to other researchers and practitioners. Results of geotechnical studies are usually stored in a database by the investigators, in tabular (text, numbers) or graphic (plot) formats. Here the past efforts in improving geotechnical information release will be discussed briefly some of the past efforts in improving geotechnical information release. The most recent effort and the most accepted format by the community will be analyzed in more detail in the next sections.



Boring Log

PROJECT: ROSRINE 4a
SITE LOCATION: Meloland Bridge
DIST/COUNTY/RTE/PM: Imperial
EA: Unknown
BRIDGE NO: Unknown
STA/OFFSET: Unknown
BORING NO: Unknown

DRILLING CONTRACTOR: Caltrans
DATES DRILLED: Unknown
DRILLING METHOD: various
HOLE DIAMETER (mm): Unknown
TOTAL DEPTH (m): 243.84
DRILL RIG MODEL/NO: Unknown
SPT HAMMER: Unknown

DATUM: NAD83
LATITUDE: 32.774
LONGITUDE: -115.449
ELEVATION: GL
LOGGED BY: USGS
ORGANIZATION: USGS
ENTERED BY: J. Swift DATE: 08/30/2000
CHECKED BY: R. Nigbor DATE:

DEPTH (m)	DEPTH (ft)	LITHOLOGY	USCS	SAMPLE TYPE AND NO	RECOVERY/LENGTH (in)	SAMPLE INTERVAL	SPT BLOWS PER 0.5'	DESCRIPTION	REMARKS
0.0 1.0	0.0		asph					Fill, asphalt	Auger drilling
2.0	5.0		cl	SPT1C SPT1B SPT1A	18/18	▦	8.5	Pale brown silty clay, sticky plastic; fill Light yellowish brown silt and vf sand; laminated, loose	SPT-1A, 1B and 1C = D-1A, 1B and 1C = D-1 Groundwater surface @ 6-7' = depth of field drains; switch from auger to rotary at 21', using Liquid Drispac polymer
3.0	10.0		sp-sm	SH-1	14/30	▦		Pale brown vf sand and silt in top of tube	Shelby, SH1 = S1; Lost bottom 7" of core and top 9"
4.0			sm	SPT-2	18/18	▦	11.5	Dark yellowish brown micaceous silty vf sand, few thin (2 cm) clay partings	SPT-2 = D-2 UCS of clay = 2.0 tsf
5.0	15.0		ml-cl	SPT-3	18/18	▦	11	Dark yellowish brown sandy silt w/ 2-3 cm thick silty clay partings; Al can frag - fill	SPT-3 = D-3
6.0			ml	SH-2	20/30	▦		Dark yellowish brown silt (from tip)	UCS=0.7 tsf. Shelby, SH-2 = S2
7.0	20.0		cl	SPT-4	18/18	▦	15.5	Dark brown silty clay, sticky, plastic w/ sandy partings overlain by 0.3' silty vf sand; prob. Colo. River deposits;	UCS=2.5 tsf. SPT4 = D-4
	25.0		cl	SPT-5	20/18	▦	8.5	Sampler overdriven 2".	UCS=1.2 tsf. SPT5 = D5

Figure 2.6 Two different graphical presentations of SPT data. Left: graphical presentations of SPT results, right: Sample boring log, showing stratigraphy, physical sampling record and SPT blow counts (ROSRINE, 2008)

2.4.1. AGS and AGSML

One of the first noticeable data representations for geotechnical information was proposed by the Association of Geotechnical and Geoenvironmental Specialists (AGS, 2004), for exchanging geotechnical data in a digital format instead of hardcopy reports. Some of the main concepts are: base data, ASCII file format, data dictionary, data groups and identifiers and units. AGS has been the most successful information release format accepted by the geotechnical community. However, AGS has difficulties in integrating data from different resources and does not provide a systematic way to check the data structure and integrity of a file.

In 2003 the AGS commissioned a working party to review the capabilities of eXtensible Markup Languages (XML) and proposed an XML version of AGS, named AGSML to give members some additional benefits (Chandler et. al. 2006). A report (AGS, 2005) and a set of schemas and example files were produced (AGSML, 2005). Separate schemas for field testing, ground information, hole information, monitoring, laboratory testing, in addition to a top level schema (Site investigation, project and hole) and a base schema (elements and type definitions) were presented. Since these primary publications of AGSML, the group has joined forces with other agencies to prepare an international format, DIGGS (Data Interchange for Geotechnical and GeoEnvironmental Specialists).

2.4.2. NGES

The AGS data dictionary, which encompasses the most common geotechnical tests in engineering practice, was extended in late 1990s to research tests by researchers of the

National Geotechnical Experimentation Sites (NGES, 2000). NGES aimed at facilitating the development of new techniques of soil characterization and earthwork construction (Benoit and Lutenegger, 2000). NGES introduced new types of research tests and added a large number of attributes to existing tests, providing much more details than AGS. Even though, these extra details might be of interest to researchers, they make usage of NGES cumbersome for practice-oriented projects.

2.4.3. NEES

Another substantial effort in the modeling of geotechnical data resulted from the George E. Brown Jr. Network for Earthquake Engineering Simulation (NEES). NEES is a research collaboratory that connects fifteen earthquake engineering testing sites located at different places in the United States through a high performance network. The model was revised numerous times to accommodate various requirements, and became unfortunately too complicated for practical use. It was burdened with too much detail and too many object types that could not be simply related. The reference metadata model was, however, a pioneering and instructive project; it was the first attempt to apply the metadata framework for documenting earthquake engineering information. NEES prompted the development of several data and metadata models including the one presented in chapter five of this study and those introduced in Bardet et al., 2004a, 2005, Peng and Law 2004 and Swift et al. 2004.

2.4.4. GeotechML

GeotechML was introduced as a Geotechnical Mark-up Language Using XML data format (Toll and Shields, 2003). A structure for ground investigation data is presented in GeotechML and concept of hierarchy structure is introduced. Document Type Definition (DTD) files are used to avoid misspelling, incorrect definition of tags and file structure. Stylesheets are used to display GeotechML data and some Java programming is used for producing graphical output. AGS data dictionary is used as the base for defining nomenclature. However, AGS scope is extended to include more complex geotechnical tests based on work of Toll and Oliver (1995). This had added complexity and reduced the consistency of naming conventions. The authors joined AGS to develop AGSML in 2005.

2.4.5. DIGGS

DIGGS is a coalition of government agencies, universities and industry partners whose focus is on the creation and maintenance of an international data exchange standard. The coalition came into existence through coordination from the US Federal Highway Administration sponsoring meetings and eventually forming the pooled fund study project. The standard includes an XML schema and associated data dictionary that are Geography Markup Language (GML, 2004) compliant. Version 1.0 of the DIGGS standard has been constructed by combining existing data dictionaries developed by AGS, and the University of Florida, Department of Civil Engineering, and COSMOS (The Consortium of Organizations for Strong-Motion Observation Systems, a pilot XML-based exchange standard for Geotechnical Virtual Data Center). The committee tasked with the development of the new DIGGS dictionary and schema consists of representatives of these three groups, along with representatives from the geotechnical software industry. DIGGS is studied in more detail later in this chapter.

2.4.6. GML-Conformant Spatial Modeling

The basic geometric objects in AGS geotechnical data are identified by Bardet and Zand, 2009. These geometric objects are mapped to basic geometric features of the Geography Markup Language (GML): Hole, single Point (for SPTs), double Point (for laboratory tests), lineString (for geology), and multiPoint (for CPTs). AGS data is rendered using GML-conformant schemas, which make geotechnical data readily importable into GML-aware applications. The data can be also imported to mainstream GIS applications using XML transformations. This approach demonstrates the rendition of AGS data format to a GML-conformant schema and illustrates the implementation of the new format through a few geotechnical examples. The acceptance and use of this GML approach will largely depend on the increase of GIS packages supporting GML-compliant models, and more familiarity of the geotechnical community with GML.

2.5. Review of AGS Format

In 1991, the Association of Geotechnical and Geoenvironmental Specialists (AGS 2004) proposed a data organization and format for exchanging geotechnical data. The latest documented and available version, 3.1 released in 2004, is used for this study. At the time of writing (July 2010) this document, AGS web site, http://www.ags.org.uk, has announced that version 4.0 is ready for distribution but the documentation is not available for review yet. AGS laid out a few basic ground rules in their development, including data dictionary and base data, identifiers, file format, two level tests, and units. AGS has been the most widely accepted information release method for geotechnical community. Hereafter, AGS will be studied in more detail to recognize its main strong points and shortcomings, to help us understand what needs improving in the process of geotechnical information release.

2.5.1. Data Dictionary and Base Data

AGS developed a data dictionary that identifies and comprehensively defines the most common geotechnical and geo-environmental data. The AGS data dictionary is well accepted and is used by a large number of geotechnical and environmental consultants especially in Europe and Asia where British Standards (BS) are used. This acceptance is clearly demonstrated by the number of geotechnical software programs that are currently supporting AGS format, listed in Table 2.2. As demonstrated by NGES (2000), the AGS data model can be adapted to other standards such as American Society for Testing and Materials (ASTM 2005a, 2005b).

Table 2.2 AGS compatible geotechnical software programs, their categories and acceptable operating systems (GGSD, http://www.ggsd.com/)

Program	Category	Operating system
AGS File Manager	Data validation	Win95/98, WinNT
AutoCAD Civil 3D 2011	Geographical information systems	WinXP, Windows Vista; Windows 7 32-bit & 64-bit
Contam Data System	Geoenvironmental database systems	Win95/98, WinNT, Win2000
Contester	Geoenvironmental database systems	WinNT, Win2000, WinXP
CPT-pro	Insitu testing	Win95/98, WinNT, Win2000, WinXP
DS7 Geotechnical Software	Laboratory testing (soil)	Win95/98, WinNT, Win2000, WinXP
GEODASY	Database systems (with log production)	Win95/98, WinNT
GEODASY CE	Field data collection	WinCE
GeoSmart II	Borehole log production	Win95/98, WinNT, Win2000, WinXP, MS Excel
GeoSmart II Lab Tool	Laboratory testing (soil)	Win95/98, WinNT, Win2000, WinXP
GEOVIEW	Geographical information systems	Win3x, Win95/98, WinNT
gINT AGS Checker	Data validation	Win2000, WinXP
gINT Logs Plus	Database systems (with log production)	Win2000, WinXP
gINT Professional	Database systems (with log production)	Win2000, WinXP
HoleBASE III	Database systems (with log production)	Win95/98, WinNT, Win2000, WinXP
INCLI-pro	Instrumentation	WinNT, Win2000, WinXP
KeyAGS	Data validation	Win95/98, Win2000, WinXP, Excel 95 or 97
KeyGeoView	Geographical information systems	Win95/98, WinNT, Web/Java
KeyHOLE	Database systems (general)	AutoCAD 2000, 2002, 2004
KeyLAB	Laboratory testing (soil)	Win95/98, WinNT, Win2000, WinXP
LA Contaminated Land Tool	Geoenvironmental database systems	Win95/98, WinNT, Win2000, WinXP
MAP-pro	Mapping	Win2000, WinXP
MonitoringPoint	Instrumentation	Web/Java
PLog	Field data collection	Win95/98, WinNT, Win2000, WinXP, Palm OS
PocketSI	Field data collection	Pocket PC 2002, 2003
SID	Database systems (with log production)	DOS, Win3x, Win95/98, WinNT
TECHBASE	Database systems (with log production)	DOS, Win95/98, WinNT, UNIX, LINUX, Open VMS
WinLoG	Borehole log production	Win95/98, WinNT, Win2000, WinXP
WINSITU	Insitu testing	Win95/98, WinNT, Win2000, WinXP

Table 2.3 lists 16 geotechnical laboratory tests documented in AGS, and indicates the close correspondence between British Standards and ASTM. The number of attributes for tests varies from 8 to 48. In general, AGS data dictionary is defined in a way that AGS files only contain basic data such as exploratory hole records and the test data required to be reported by the relevant British Standards and other recognized documents. These data would normally be contained within the final report. Any interpretation and calculation needs to be done by the user, rather than being transferred within the data files.

2.5.2. Data Groups and Unique Identifiers

AGS (2004) uses data groups "to structure data in consistent and logical manner within which a series of fields are defined. They have been chosen to relate to specific elements of data which are obtained" such as, project information, and borehole details. It also uses "fields within each data group to identify specific items," such as, borehole date and sample depth.

Figure 2.7 shows a typical geometry for the main AGS data groups, which are project, hole, sample and specimen. Project is a collection of holes belonging to the same engineering project. Hole is a single sampling station, from which soil materials are collected or described, or soil material properties are measured. Sample is a segment from a hole, which is located using the depth relative to the top of borehole. Specimen is a part of sample collected for the purpose of description or testing, which is also located using depth. All laboratory tests are performed on specimens that originate from boreholes. AGS identifies data group instances using unique identifiers, which avoid repetition of data and distinguish data group instances from one another.

Table 2.3 Geotechnical laboratory tests defined by AGS data dictionary, their number of attributes and their corresponding British Standard and ASTM standard tests

Laboratory Test	AGS Description	Attributes	British Standard	ASTM Test	ASTM standard
CBR Test	CBR Test - General CBR Test	14 15	BS 1377- Part 4	California Bearing Ratio	D1883
Chalk Tests	Chalk Tests	14	BS 1377- Part 4		
Classification Tests	Classification Tests	23	BS 1377- Part 2	Atterberg Limits, Vane Test, Shrinkage Limit Analysis, Specific Gravity, Water Content, Soil Classification	D4318, D4648, D427, D4943, D854, D2216, D2487
Compaction Tests	Compaction Tests General Compaction Tests	15 9	BS 1377- Part 4	Compaction Test	D1140, D1557
Contaminant and Chemical Testing	Contaminant and Chemical Testing	21	BS 1377- Part 3		
Consolidation Test	Consolidation Test - General Consolidation Test	23 16	BS 1377- Part 5	Consolidation Test, Swell Test	D2435, D4186, D4546
				Expansion Index	D4829
Frost Susceptibility	Frost Susceptibility	15	BS 812- Part 124		D5918
Laboratory Permeability Tests	Laboratory Permeability Tests	23	BS 1377- Part 5	Permeability Test	D2434
MCV Test	MCV Test - General MCV Test	11 10	BS 1377- Part 4	Determination of Moisture Content of Soil	D2216, D4959
Particle Size Distribution Analysis Data	Particle Size Distribution Analysis Data	9	BS 1377- Part 2	Particle Size Analysis, Hydrometer Analysis	D422, E100
Relative Density Test	Relative Density Test	12	BS 1377- Part 4	Density Tests	D4253, D4254
				Resonant Column Test	D4015
Rock Testing	Rock Testing	48	BS 812		D2845, D2938, D2936, D3967, D5731
Shear Box Testing	Shear Box Testing - General Shear Box Testing	13 20	BS 1377- Part 7	Direct Shear Test	D3080
Suction Tests	Suction Tests	8		Soil Suction Test	D5298
Ten Per Cent Fines	Ten Per Cent Fines	11	BS 812		D422, D421, D2217
				Torsional Shear Test	D6467
Triaxial Test	Triaxial Test - General Triaxial Test	13 19	BS 1377- Part 9	Triaxial Test	D4767, D2850
				Unconfined Compression Test	D2166

The unique identifiers are aggregation of several identifiers. AGS identifies each laboratory test instance by a unique identifier through aggregating the identifiers of Project, Hole, Sample and Specimen.

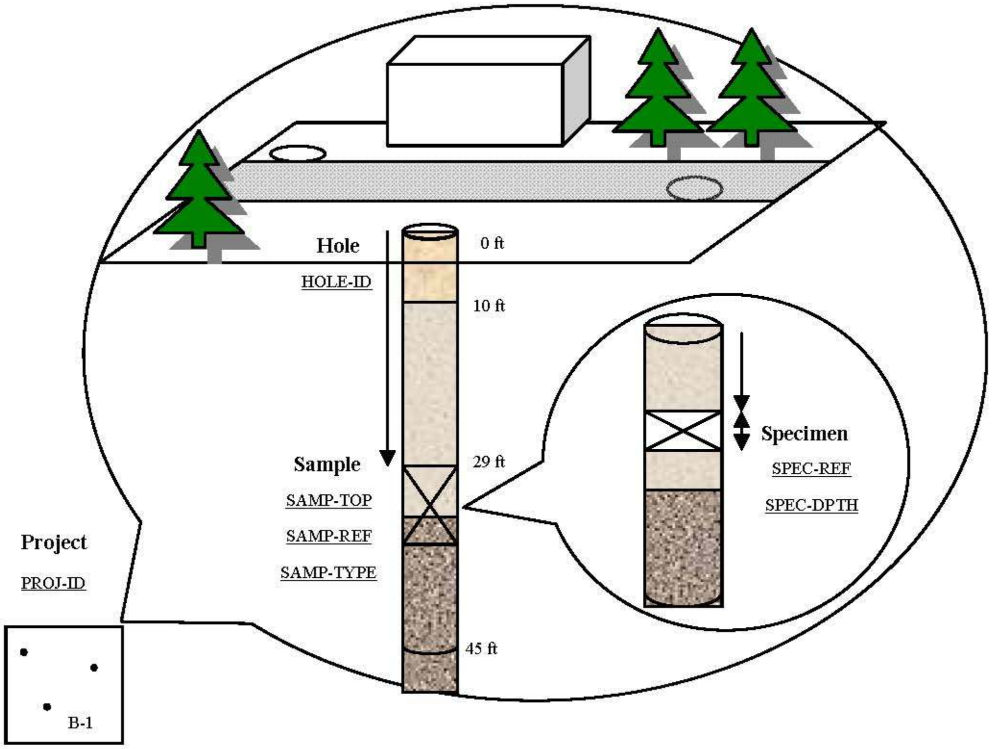


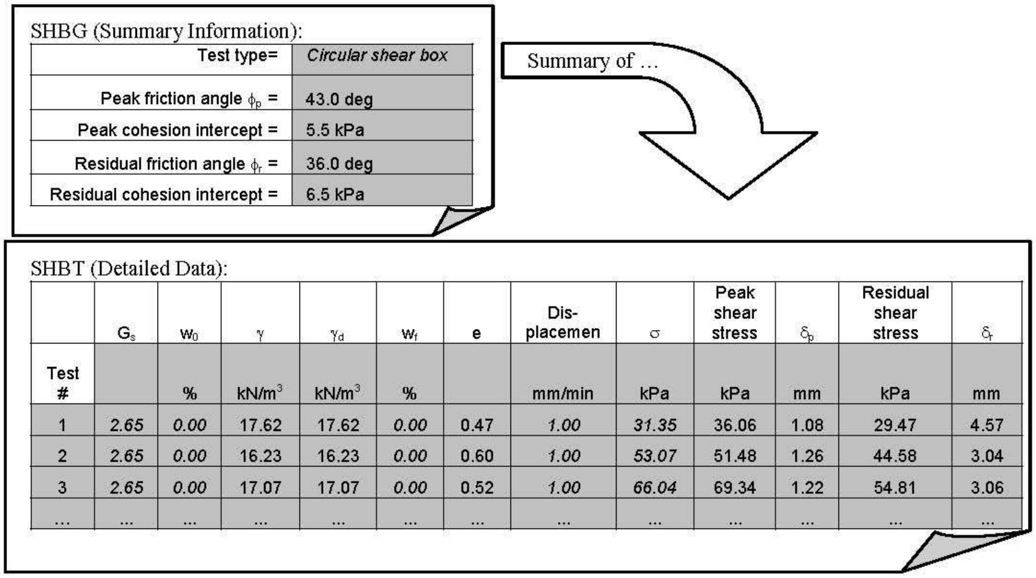
Figure 2.7 Relationships between Project, Hole, Sample and Specimen in AGS

2.5.3. File Format

AGS files have been expressed in the American Standard Code for Information Interchange (ASCII) format since AGS inception in 1991. AGS file format is supported by some of the leading geotechnical software, such as gINT and TECHBASE for basic applications such as graphing and presenting lab tests or borehole data, report generation and layer stratigraphy (Table 2.2). However AGS files are difficult to handle in case of large data volume in complicated projects. They cannot be spatially displayed or systematically queried similar to newer data formats which have been introduced in recent years as means to enhance machine-to-machine communications. AGS has outlived its file format and needs to be updated to take advantage of newer file formats with better capabilities than ASCII.

2.5.4. Two Table Tests

AGS describes data in two levels, i.e., summary and details. For instance as shown in Table 2.4, AGS subdivides the data dictionary for direct shear tests into two groups SHBG and SHBT. The first data group, which has a name ending with the 'G' suffix, presents the general test information and main test results. It contains 13 attributes, 6 of which are identifiers as denoted by an asterisk. The second data group contains the detailed data that support the main results. It contains 20 attributes including 7 identifiers. The two groups have the same first six identifiers. For a complete description of test results, both data groups are reported. However, when the detailed data is not required or not available, the first data group may be reported on its own as it remains meaningful for most engineering applications. This two-level documentation of data is useful for promoting AGS organization/format; engineering firms could choose to share only summary data, and therefore to spend less time in documenting data. Figure 2.8 shows an example of AGS data for direct shear test presented in two levels. The top table contains basic shear strength properties, i.e., friction angle and cohesion, while the second table lists additional data which help engineers to understand how shear strength was obtained from test results.



SHBG (Summary Information):

Test type=	*Circular shear box*
Peak friction angle ϕ_p =	43.0 deg
Peak cohesion intercept =	5.5 kPa
Residual friction angle ϕ_r =	36.0 deg
Residual cohesion intercept =	6.5 kPa

SHBT (Detailed Data):

	G_s	w_0	γ	γ_d	w_f	e	Dis-placemen	σ	Peak shear stress	δ_p	Residual shear stress	δ_r
Test #		%	kN/m^3	kN/m^3	%		mm/min	kPa	kPa	mm	kPa	mm
1	*2.65*	*0.00*	17.62	17.62	*0.00*	0.47	*1.00*	*31.35*	36.06	1.08	29.47	4.57
2	*2.65*	*0.00*	16.23	16.23	*0.00*	0.60	*1.00*	*53.07*	51.48	1.26	44.58	3.04
3	*2.65*	*0.00*	17.07	17.07	*0.00*	0.52	*1.00*	*66.04*	69.34	1.22	54.81	3.06
…	…	…	…	…	…	…	…	…	…	…	…	…

Figure 2.8 AGS two level data structure for direct shear test including summary information in one table (SHBG) and detailed data in another (SHBT)

Table 2.4 AGS data dictionary for direct shear test in two tables: Shear Box Testing General (SHBG) and Shear Box testing (SHBT) (AGS 2004)

Identifier	Heading	Unit	Description	Example
		Group Name : SHBG - Shear Box Testing - General		
*	HOLE_ID		Exploratory hole or location equivalent	6331/A
*	SAMP_TOP	m	Depth to TOP of test sample	6.50
*	SAMP_REF		Sample reference number	12
*	SAMP_TYPE		Sample type	D
*	SPEC_REF		Specimen reference number	2
*	SPEC_DPTH	m	Specimen Depth	6.50
	SHBG_TYPE		Test type e.g. small shear box, large shear box, ring shear	Circular shear box
	SHBG_REM		Test notes e.g. undisturbed, pre-existing shear, recompacted, rock joint, cut plane	Recompacted
	SHBG_PCOH	kN/m2	Peak cohesion intercept	5.5
	SHBG_PHI	deg	Peak angle o f friction	43.0
	SHBG_RCOH	kN/m2	Residual cohesion intercept	6.5
	SHBG_RPHI	deg	Residual angle friction	36.0
	FILE_FSET		Associated file reference	FS18
		Group Name : SHBT - Shear Box Testing		
*	HOLE_ID		Exploratory hole or location equivalent	6331/A
*	SAMP_TOP	m	Depth to TOP of test sample	6.50
*	SAMP_REF		Sample reference number	12
*	SAMP_TYPE		Sample type	D
*	SPEC_REF		Specimen reference number	2
*	SPEC_DPTH	m	Specimen Depth	6.50
*	SHBT_TESN		Shear box stage number	1
	SHBT_BDEN	kN/m3	Bulk density	17.62
	SHBT_DDEN	kN/m3	Dry density	17.62
	SHBT_NORM	kN/m2	Shear box normal stress	31.35
	SHBT_DISP	mm/min	Displacement rate	1
	SHBT_PEAK	kN/m2	Shear box peak shear stress	36.06
	SHBT_RES	kN/m2	Shear box residual shear stress	29.47
	SHBT_PDIS	mm	Displacement at peak shear strength	1.08
	SHBT_RDIS	mm	Displacement at residual shear strength	4.57
	SHBT_PDEN	kN/m3	Particle density, measured or (#) assumed	2.65
	SHBT_IVR		Initial void ratio	0.47
	SHBT_MCI	%	Initial moisture content	0
	SHBT_MCF	%	Final moisture content	0
	SHBT_REM		Remarks on test stage	Reached end of travel

2.5.5. Unit Types

The abbreviations to be used for standard units are given in a UNIT 'pick' list in AGS that are represented in Table 2.5. Where standard units are used the format must comply exactly with that given in the 'pick' list. As presented in Table 2.5, the units are classified as one of the following types: length, area, volume, force, mass, pressure, density, time, velocity, flow, concentration, and miscellaneous.

Table 2.5 AGS data dictionary abbreviation pick list for standard units (AGS 2004)

Group Name : Unit		**Definition of <UNITS>**	
UNIT-UNIT	UNIT_DESC	UNIT-UNIT	UNIT_DESC
Length		**Volume**	
mm	millimeter	cm3	cubic centimeter
cm	centimeter	m3	cubic meter
m	meter	l	liter
km	kilometer	in3	cubic inch
in	inch	ft3	cubic foot
ft	foot	yd3	cubic yard
yd	yard	gal	gallon
mi	mile	**Mass**	
Area		g	gram
cm2	square centimeter	kg	kilogram
m2	square meter	Mg	megagram (tonne)
km2	square kilometer	lb	pound
hect	hectare	t	ton
in2	square inch	kips	kilopound
ft2	square foot	**Force**	
yd2	square yard	N	Newton
mi2	square mile	kN	kiloNewton
acre	acre	MN	megaNewton
Pressure		lbf	pounds force
kN/m2	kiloNewtons per square meter	tonf	tons force
KPa	kiloPascal	kgf	kilograms force
MN/m2	megaNewtons per square meter	**Density**	

Table 2.5: Continued

Group Name : Unit		**Definition of <UNITS>**	
UNIT-UNIT	UNIT_DESC	UNIT-UNIT	UNIT_DESC
MPa	megaPascal	kN/m3	kiloNewtons per cubic meter
GPa	gigaPascal	Mg/m3	megaNewtons per cubic meter
psi	pounds per square inch	pcf	pounds per cubic foot
psf	pounds per square foot	g/cm3	grams per cubic centimeter
ksi	kips per square inch	kg/cm3	kilograms per cubic meter
ksf	kips per square foot	kg/m	kilograms per meter run
tsf	tons per square foot	**Time**	
kg/cm2	kilograms per square centimeter	s	second
bar	bar	min	minute
Velocity		hr	hour
mm/s	millimeters per second	day	day
cm/s	centimeters per second	month	month
m/s	meters per second	yr	year
km/hr	kilometers per hour	hhmm	hours minutes
ft/min	feet per minute	hhmmss	hours minutes seconds
mph	miles per minute	dd/mm/yyyy	day month year
Flow		**Concentration**	
l/s	liters per second	ug/l	micrograms per liter
l/min	liters per minute	mg/l	milligrams per liter
m3/s	cubic meters per second	g/l	grams per liter
gpm	gallons per minute	ug/kg	micrograms per kilogram
mgd	million gallons per day	mg/kg	milligrams per kilogram
cfs	cubic feet per second	ppb	parts per billion
Miscellaneous		ppm	parts per million
m2/MN	cubic meters per megaNewton	ppmv	Part per million volume
ft2/t	square feet per ton	%	percentage
m2/yr	square meters per year	% dry weight	percentage of dry weight
ft2/yr	square feet per year	% vol	percentage volume
ft2/day	square feet per day	ftu	Formazin turbidity unit
Nm	Newton meter	%LEL	percentage of Lower Explosive Limit
deg	degree (angel)	colonies/ml	colonies per milliliter
DegC	degree Celsius	colonies/l	colonies per liter
DegF	degree Fahrenheit	CFU/ml	colony forming per milliliter
uV	microVolt	CFU/g	colony forming units per gram
mV	milliVolt	MPN/ml	most probable number per milliliter
ohm	Ohm	MPN/100ml	most probable number per 100 milliliters

Table 2.5: Continued

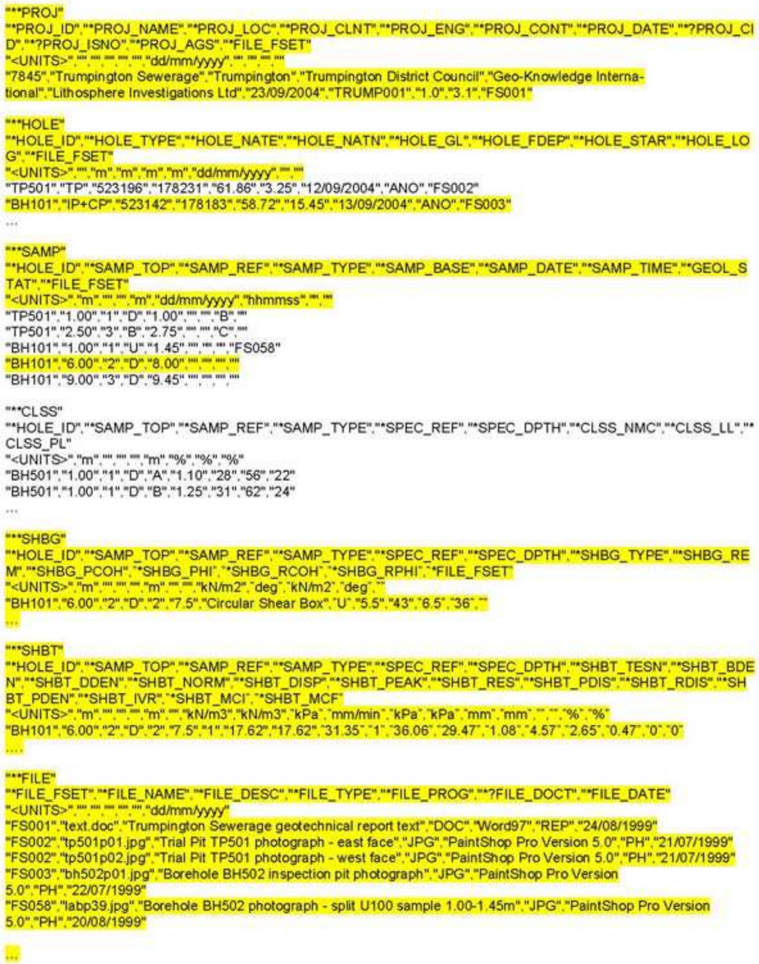
ohmcm	Ohm centimeter	MPN/l	most probable number per liter
uS/cm	microSiemens per centimeter		
kJ/kg	kilojoules per kilogram		
counts/s	counts per second		
Yes	Yes		
No	No		

2.5.6. Data Integrity and Transfer

As shown in the example of Figure 2.9, AGS data are made of a sequence of data groups, which are arranged in two dimensional arrays. The first line of each AGS data group contains the attributes' (fields) names and the second row the corresponding measurement units. The third and following rows contain actual data described by the first two lines. Data groups and attribute names are defined in the AGS data dictionary. For instance, the direct shear test data groups attributes (SHBG and SHBT attributes) are defined in Table 2.4. Data fields for each data group instance are listed in separate rows following the rows of attribute names and units.

AGS has difficulties in integrating data from different resources because it refers to data in cumbersome ways. In practice, projects are carried out by different contractors, e.g., tests can be performed in separate laboratories. AGS can only preserve the relationship between lab tests and overall project by including all data groups in data files including information on project, borehole and samples. For instance, the parts highlighted in Figure 2.9 show the referential information that needs to be included for shear box tests. This will also increase the possibility of errors. For instance some laboratory technician only interested in the shear test results might unintentionally change other parts of data transferred with shear test data.

Additionally, AGS does not provide an automated way to check the data structure and integrity of a file. It has, therefore, been left to the software providers to provide their own AGS checkers, many of which are available from the web, free of charge. However, each company has interpreted the rules differently, resulting in each checker giving slightly different error messages. This is in part due to use of outdated ASCII format for data exchange. Since the inception of AGS significant advances have happened in the field of data exchange formats with introduction of Extensible Markup Languages (XML) family. DIGGS format discussed in detail in the next section attempts to take advantage of XML. XML capabilities will also be reviewed in next chapter.



```
"**PROJ"
"*PROJ_ID","*PROJ_NAME","*PROJ_LOC","*PROJ_CLNT","*PROJ_ENG","*PROJ_CONT","*PROJ_DATE","*?PROJ_CI
D","*?PROJ_ISNO","*PROJ_AGS","*FILE_FSET"
"<UNITS>","","","","","","dd/mm/yyyy","","","",""
"7845","Trumpington Sewerage","Trumpington","Trumpington District Council","Geo-Knowledge Interna-
tional","Lithosphere Investigations Ltd","23/09/2004","TRUMP001","1.0","3.1","FS001"

"**HOLE"
"*HOLE_ID","*HOLE_TYPE","*HOLE_NATE","*HOLE_NATN","*HOLE_GL","*HOLE_FDEP","*HOLE_STAR","*HOLE_LO
G","*FILE_FSET"
"<UNITS>","","m","m","m","m","dd/mm/yyyy","",""
"TP501","TP","523196","178231","61.86","3.25","12/09/2004","ANO","FS002"
"BH101","IP+CP","523142","178183","58.72","15.45","13/09/2004","ANO","FS003"
...

"**SAMP"
"*HOLE_ID","*SAMP_TOP","*SAMP_REF","*SAMP_TYPE","*SAMP_BASE","*SAMP_DATE","*SAMP_TIME","*GEOL_S
TAT","*FILE_FSET"
"<UNITS>","m","","","m","dd/mm/yyyy","hhmmss","",""
"TP501","1.00","1","D","1.00","","","B",""
"TP501","2.50","3","B","2.75","","","C",""
"BH101","1.00","1","U","1.45","","","","FS058"
"BH101","6.00","2","D","8.00","","","",""
"BH101","9.00","3","D","9.45","","","",""

"**CLSS"
"*HOLE_ID","*SAMP_TOP","*SAMP_REF","*SAMP_TYPE","*SPEC_REF","*SPEC_DPTH","*CLSS_NMC","*CLSS_LL","*
CLSS_PL"
"<UNITS>","m","","","","m","%","%","%"
"BH501","1.00","1","D","A","1.10","28","56","22"
"BH501","1.00","1","D","B","1.25","31","62","24"
...

"**SHBG"
"*HOLE_ID","*SAMP_TOP","*SAMP_REF","*SAMP_TYPE","*SPEC_REF","*SPEC_DPTH","*SHBG_TYPE","*SHBG_RE
M","*SHBG_PCOH","*SHBG_PHI","*SHBG_RCOH","*SHBG_RPHI","*FILE_FSET"
"<UNITS>","m","","","","m","","","kN/m2","deg","kN/m2","deg",""
"BH101","6.00","2","D","2","7.5","Circular Shear Box","U","5.5","43","6.5","36",""
...

"**SHBT"
"*HOLE_ID","*SAMP_TOP","*SAMP_REF","*SAMP_TYPE","*SPEC_REF","*SPEC_DPTH","*SHBT_TESN","*SHBT_BDE
N","*SHBT_DDEN","*SHBT_NORM","*SHBT_DISP","*SHBT_PEAK","*SHBT_RES","*SHBT_PDIS","*SHBT_RDIS","*SH
BT_PDEN","*SHBT_IVR","*SHBT_MCI","*SHBT_MCF"
"<UNITS>","m","","","","m","","kN/m3","kN/m3","kPa","mm/min","kPa","kPa","mm","mm","","","%","%"
"BH101","6.00","2","D","2","7.5","1","17.62","17.62","31.35","1","36.06","29.47","1.08","4.57","2.65","0.47","0","0"
....

"**FILE"
"*FILE_FSET","*FILE_NAME","*FILE_DESC","*FILE_TYPE","*FILE_PROG","*?FILE_DOCT","*FILE_DATE"
"<UNITS>","","","","","","dd/mm/yyyy"
"FS001","text.doc","Trumpington Sewerage geotechnical report text","DOC","Word97","REP","24/08/1999"
"FS002","tp501p01.jpg","Trial Pit TP501 photograph - east face","JPG","PaintShop Pro Version 5.0","PH","21/07/1999"
"FS002","tp501p02.jpg","Trial Pit TP501 photograph - west face","JPG","PaintShop Pro Version 5.0","PH","21/07/1999"
"FS003","bh502p01.jpg","Borehole BH502 inspection pit photograph","JPG","PaintShop Pro Version
5.0","PH","22/07/1999"
"FS058","labp39.jpg","Borehole BH502 photograph - split U100 sample 1.00-1.45m","JPG","PaintShop Pro Version
5.0","PH","20/08/1999"

...
```

Figure 2.9 A part of AGS data file, showing PROJ, HOLE, SAMP, CLSS, SHBG, SHBT and FILE data groups. The highlighted parts should all be carried for preserving referential integrity if only shear box test information wants to be transferred.

2.6. Analysis of DIGGS Format

The focus of the DIGGS project is to compile the work of the AGS, COSMOS, the University of Florida, and others to create a new international data exchange format. DIGGS version 1.0, released in July 2008, contains a large array of definitions for the transfer of subsurface and substructure information, and is designed to be extensible. Subsurface and substructure features currently included in DIGGS v. 1.0 are: geotechnical ground investigation data, geoenvironmental data, deep foundations data, and borehole geophysical investigation data (DIGGSML 2010). DIGGS objective is to be GML-conformant (Weaver et. al. 2008), define a very detail and comprehensive data dictionary (Styler et. al. 2007), and allow extensions and customizations (Bray et. al., 2008).

DIGGS admits that even though the first version is released for review, the work is substantially unfinished. An invitational meeting was held in Orlando, Florida in March 25-26, 2009 to re-evaluate goals of the project, identify tools and reconsider the breadth of data types (DIGGS 2009). Experts from public agencies such as UK Highway Administration (Patterson 2009), software developers such as gINT (Corrona 2009), and Dataforensics (Deaton 2009), and AGS (Chandler 2009) raised concerns regarding the data model during this meeting.

Deaton (2009) argues that DIGGS terminology and documentation are inconsistent. In order for an interchange standard to be intuitive, the naming convention should be logically consistent. An interchange standard should be simple to understand and variables should be well defined. In addition, the data format does not need to reinvent the wheel. Calling upon acceptable test standards will ensure acceptance by the community.

One of the main potential advantages for XML-based organization of data is the self-validating nature of the format. However, Corrona (2009) finds numerous instances of invalid files passing DIGGS validation. DIGGS allows users to create files that are validated, but functionally invalid due to circular loops or other underlying design flaws. Patterson (2009) also points that schema alone does not fully validate data and data with no relation with other entities in a project can exist in a valid DIGGS file. He also notes that ease of extensibility promotes development of "non-standard standards". Chandler (2009) notes that validation takes too long using online schemas and mandatory elements are not tagged as such. Lab testing allows for tests to be carried out on more than one specimen. This is a significant validation problem happening since the relationships are not uniquely defined. Technical questions are also raised over current tabular data structure. CPT utilizes a table structure that is not used for other tabular data types such as pressuremeter. dialotometer, dynamic probes. This is an illustration of inconsistent data structure.

DIGGS version 1.0 has problems to be opened in an XML editor in a timely fashion. Currently, it is not possible to generate an empty file for later data input from schema. Patterson (2009) suggests that use of XML increases the number of available software programs that can potentially handle DIGGS data. However, DIGGS is a very complex format compared to AGS and as listed above, it has many structural inconsistencies and validation problems. Coronna (2009) raises similar issues as problems for implementation and adaptation of DIGGS by software vendors and government agencies.

DIGGS is a GML compliant format. Two advantages of GML compliant files are that any GML compliant program should be able to display GML objects and Web-based tools should be able to convert coordinate systems and perform geospatial data processing. At the current state, DIGGS files do not work in GIS-enabled software such as ArcMap and MapInfo or systems such as AutoCAD (Chandler 2009). Additionally, some researchers ask the question whether having a GML-complaint format is worth the added complexity (Chandler 2009; and Corrona 2009). At the same time, there are still not many GML compliant applications. GML format is not as widely accepted and used by data users as it was predicted at the time of its inception and it makes the schema far more complicated. Calling upon formats which are not yet accepted by the community will not provide for a successful and well accepted data exchange format.

As it can be seen, the challenges and obstacles of DIGGS are similar to NEES project. DIGGS is trying to define a data interchange format but is not successful in version 1.0. General consensus of experts is that DIGGS has become very complex and has a number of problems that need to be solved before being released for usage of the community or being the next version of AGS release. As a confirming point, it should be mentioned that AGS committee did not use DIGGS for AGS4. Once a version of DIGGS is released that mitigates these issues it can then be tested by software vendors as well as individual users.

2.7. Requirements for Geotechnical Data Exchange Format

Based on the shortcomings and advantages of past efforts reviewed in this chapter and information technology advancements a list of requirements is specifically developed for geotechnical exchange data format. After development of the requirement list, a road map can be established to develop the data exchange with the help of information technology advancements. The proposed data exchange format should be simple, yet well formed, to be accepted by the community. A proper data exchange format will improve efficiency and data quality and eventually will reduce costs.

Before going in more detail, it is important to have a clear understanding of some of the terms used in data modeling and establishing them for this study. As suggested by Elmasri and Navathe (2003), conceptual data models use concepts such as entities, attributes, and relationships. An entity represents a real world object or concept, such as project engineer or borehole log. An attribute presents some property of interest that further describes an entity, such as project engineer's name or borehole log's type (e.g. SPT or CPT). A relationship among two or more entities represents an association among them, for example, perform is the relationship between a technician and a lab test.

The list of requirements is grouped into two categories, as general criteria, and data specific criteria.

General criteria

G1) Consistent nomenclature: Is naming convention consistent? For instance, PROJ_NAME, PROJ_LOC, and PROJ_DATE from AGS data dictionary, all use a shortened version of project (PROJ) at the beginning of each heading for project data. The same convention is used for other objects of study throughout AGS data dictionary. Use of consistent conventions will make comprehension of names easier for the user.

G2) Unit Abbreviation: Are unit abbreviations according to physical units? Choices of units should be unique, clear and from a limited list. For example, KPa could be used for kiloPascal units and defined in the unit dictionary which clearly spells the unit abbreviations. KPascal could be intuitive for some but confusing for others if not defined. This will make usage of data exchange easier and prone to fewer unit errors.

G3) Integrability of data subsets: Can subsets be integrated back to a complete data set? Subsets of data should be uniquely related to the rest of data. They should be exchangeable and easily integrated back to the larger datasets. For instance, laboratory shear test resulting in separate files should be easily inserted into the complete data set in the right place without transferring the complete file to laboratory technician who is not interested in the rest of data. AGS only authorizes exchange of full data set and does not have a system for integrating data subsets.

G4) Standardized test procedures: Does exchange data come from well accepted test standards? The data exchange format is not intended to set standards for test procedures. Calling upon acceptable test standards will simplify the process and help with acceptance by the community. For example, most of practice-oriented geotechnical data is covered by standards such as ASTM, BS, and Eurocode. However, research tests have many variations which basically make data procedure a part of data and are less likely to be defined in standards.

G5) Exchangeability via email: Can data be exchanged via email? Email is one of the most common modes of communication within the engineering community. At this moment, the size of files should be smaller than 2MB for convenient exchange via email. For instance, AGS files are usually small, however some data formats can become large. This requirement will obviously evolve with increase of email capacities and new free FTP programs and other protocols.

G6) General acceptance by the community: Can the data exchange format be accepted and used by community? The ultimate test for an exchange format is its acceptance by the community. The exchange format needs to be simple enough for the community to a) understand, b) use and c) convert to and from the existing standards and data. It is fair to say that if an exchange format is difficult and sophisticated, it is likely to fail due to errors and wasted time associated with it. Acceptance is usually driven by users at large. For instance, NEES is a very powerful model; however its complexity has made its chances of adoption and usage by the geotechnical community slim. On the other hand, AGS has gained relative acceptance within the community.

Data/ technical specific criteria:

D1) Data dictionary: Are all variables defined based on common knowledge in the field using plain English (or other languages)? Users should not be responsible for naming the fields or measurement types. For instance, AGS uses STCN_RES as a heading abbreviation used for cone resistance in static cone penetration test. This is explicitly defined in AGS data dictionary. Consequently there is no ambiguity on what STCN_RES means. However, without the data dictionary the definition might not be clear for the user.

D2) Unit validation: Are all values associated with appropriate units? Errors in units are one of the most serious errors in engineering problems. All the numbers should be associated with units in the proper form. For example if the physical property measured is of length type, unit should also corresponds to length. However, having no units implies the quantity is dimensionless. If square meter is used for length unit data is not valid. Hence, a proper unit validation method should exist.

D3) Value format validation: Are the values in the right format, e.g. integer, string? For instance, for cone resistance if string format is used, it is not correct. As the value should be a real number, e.g. 2.32. NA is no valid. A proper format validation method should exist and validation should be done in an acceptable time window.

D4) Value range validation: Are the values in an acceptable range? For example, no negative friction angle or cohesion exists. At least some warning should be issued to the user for the values outside the commonly accepted ranges in the data set.

D5) Missing data recognition: Are missing fields recognized with error messages? For instance, depth value for cone tip resistance values should always be present; otherwise, the data will be meaningless. A proper validation method should exist to distinguish missing data fields with error messages.

D6) Interoperability: Can exchange data be used in different platforms and operating systems? A proper data exchange format is platform independent and operating system independent. If the data exchange format is tied to a specific platform, data users will be discouraged to use it. For example the data exchange format need to be able to exchange data across all the platforms such as Mac, UNIX, windows, dos and Linux.

D7) Wide support: Is the data format readily supported by commercial and non-commercial tools? Commercial and open standard programs should be able to view content and tabulate and locate data. For instance, GML is an exchange format for spatial data; however some of GML features are not supported by the main GIS programs. Therefore, using these features of GML will render data unusable with GIS programs. Although these new features might be supported in future, including them in exchange format will make it unusable at present.

D8) Archivable data: Can data be indexed for archival so that it can be retrieved later? It would be of benefit if data can be easily archived in some kind of data repository that can index them for faster retrieval in future. For example, AGS needs parsing (string recognition through the file) for indexing. Some newer formats like XML clearly separate tags from content so no additional parsing is needed.

D9) Machine searchable: Is it possible to search within the data file for particular value or attribute, using some kind of search engine? For instance, can all depths that tip cone resistance have a value larger than an affixed value, be retrieved without extra data parsing using regular search engines such as XQuery? For example, AGS files are difficult to search.

D10) Locatable: Can data be located on interactive maps such as Google Maps? Spatial data will relate the information to its location. For instance, is it possible to open the file and see where the data is located without any extra programming? AGS contains the coordinates but needs parsing. KML files on the other hand can be easily displayed.

D11) GIS-enabled: Are all objects in data following GIS accepted format so that all objects can be positioned in 3 dimensional spaces? GIS enabled data will be identified and presented by GIS software programs. Geotechnical data is derived from samples associated with a specific location. If the data is GIS-enabled, we should be able to distribute data on different geometric features such as lines, points, and segments. AGS data is not GIS-enabled.

D12) Unique relationships: Is the exchange format capable of identifying missing objects or objects with non-unique or missing relations? There should be no ambiguity in relationships. For example, if there are two shear tests with the same id it will create confusion or if a test calls upon a borehole that does not exist, it will be orphan and the data will not be usable. This can be avoided with some error messages during data validation that request data user to change values to preserve unique relationships.

D13) No circular relationship: Can data format detect circular relations between objects which create infinite circular loops? For instance, if an object calls upon itself, an infinite loop is created. Additionally, high depth of recursion may run the risk of having loops. Circular relations can be avoided by a well designed data structure for data.

D14) Random data generation: Is it possible to generate random instances of objects for populating data? There is a need for generation of empty files for inserting data later. For example, is it possible to generate file for shear test to be filled with the corresponding values obtained in the laboratory later? AGS does not have capability of generating random data. However, XML schemas can do random generation.

D15) Flatable structure: Can all relationship identifiers be replaced with object attributes to form tables without any relationships? The data structure should be designed in a way that flattening of data will be possible. For flattening, identifiers pointing to other objects will be replaced with the object information. Therefore, relationships are gone but the data is available. This test ensures that the data can be later archived in a DBMS if necessary. For instance, in the shear test, sample id can be replaced by the sample information. Therefore, the id can be removed from shear test and all the needed data for sample will be carried with the shear test data.

D16) Annotation of data: Can data exchange format hold annotations (additional information)? For instance, data remarks should be kept as annotation within the data set. AGS uses REM to amend data in various ways. However the text is free format and does not have any structure.

D17) Machine readable annotation: Are annotation free of structured metadata (data about data)? For instance, can data exchange be structured to relate objects beyond the defined relationships in the exchange format? AGS allow us to make connections in REM text, but this could be understood by human not machines. We do not want this now but envision it for future in chapter 5.

Table 2.6 List of requirements for a geotechnical data exchange format

Code	Name	Criteria Definition
G1	Consistent Nomenclature	Is naming convention consistent?
G2	Unit Abbreviation	Are unit abbreviations according to standard units?
G3	Integrability of Data Subsets	Can subsets be integrated back to larger data sets?
G4	Using Accepted Test Procedures	Does data come from well accepted test standards?
G5	Exchangeability via Email	Can data be exchanged via email?
G6	Acceptance by the Community	Can dataexchange be accepted and used by the community?
D1	Data Dictionary	Are all variables defined based on common knowledge in the field?
D2	Unit Validation	Are all values associated with units?
D3	Value Format Validation	Are the values in the right format, e.g., integer, string?
D4	Value Range Validation	Are the values being validated to be in acceptable range?
D5	Missing Data Recognition	Are missing fields recognized with error messages?
D6	Interoperability	Can exchange data be used in different platforms and operating system?
D7	Wide Support	Is the data format readily supported by non-commercial and commercial tools?
D8	Archivability	Can data be indexed for archival and later retrieval?
D9	Machine Searchable Data	Is it possible to search within the data file for particular value or attribute, using a search engine?
D10	Locatable	Can data be located on interactive maps?
D11	GIS-enabled	Are all objects in data following GIS accepted format, such as GML, so that they can be position in 3 dimensional spaces?
D12	Unique Relationships	Is the exchange format capable of identifying missing objects or objects with non-unique or missing relations?
D13	No Circular Relationship	Is data format free of circular relations between objects which create infinite circular loops?
D14	Random Data Generation	Is it possible to generate random instances of objects for populating data?
D15	Flatable Structure	Can all relationship identifiers be replaced with object attributes to form 2D tables without any relationships?
D16	Data Annotation	Can data exchange format hold annotations (additional information)?
D17	Machine Readable Annotation	Does annotation have capability of containing structured metadata?

Among the reviewed releases, AGS does meet some of the data exchange criteria, but not all of them. AGS has a consistent nomenclature (data dictionary), that uses accepted test procedures (British Standards). It has a unit enumeration, and has been accepted and been used within the community up to some extent. ASCII files are small in size and can easily be exchanged via email; however, after exchange the user needs to view the files in plain text or should have a program that is specifically programmed to view AGS data format. Integrating subsets of data needs to be done by an expert completely familiar with AGS and the data manually and there is a high chance of making mistakes. AGS data dictionary has undergone number of revisions with the release of each new version of AGS. Therefore, it is a very well defined and well established dictionary. As study with object oriented modeling will show in next chapter, relationships between different entities can uniquely be defined and no ill-defined circular relation exists within AGS data groups. It will be shown that AGS data is all defined in tables and is in relational format. So, flattening of data is possible. AGS only carries base data such as exploratory hole records and the test data required that would normally be contained within a final report. However, no systematic way exists for validation of units, value formats and value ranges. There is no way to recognize missing fields or search within a file either. Data positioning is defined within the text file, but data location cannot be automatically displayed on maps. Random data generation is not possible either. On the other hand DIGGS effort to define an up-to-date data format with the help of XML has resulted in a very complex format with validation problems and structural inconsistencies that will reduce the chance of being acceptable by the community. Consequently, there is plenty of room for developing a better data exchange format.

2.8. Summary

Understanding geotechnical community members and their roles in geotechnical data handling is critical to identify community needs for a proper data exchange format. In this chapter, a sample project was used as a mean to understand the extent of different possible data types in geotechnical projects. It was demonstrated that geotechnical information has many types and will go through many phases and will be changed and exchanged repeatedly during its lifetime. However, the two most commonly produced geotechnical information categories recognized in geotechnical projects are the site investigation components; in situ tests and laboratory tests. From these two primitive geotechnical information categories, the ones most commonly used in practice are selected and listed as the scope of this study. Previous efforts in organized geotechnical information release were studied. The data format published by the Association of Geotechnical and Geoenvironmental Specialists (AGS), was recognized as one of the more successfully accepted efforts and Data Interchange for Geotechnical and GeoEnvironmental Specialists (DIGGS) was listed as one of the most recent (still under development) efforts. AGS and DIGGS were analyzed in more depth. Even though, AGS benefits form a very well defined data dictionary that is conforming to British Standards; the ASCII format used for the data is rather outdated and needs additional processing for display and manipulation in geotechnical programs. In other word, special software or interpretation of information is required. DIGGS is currently in the process of developing a GML-compliant XML format for the geotechnical information. However, some expert reviews show that DIGGS current version is too complicated. Finally, based on the shortcomings and advantages of these studied efforts and the needs of geotechnical community a list of requirements for a domain-specific proper data exchange format was introduced. In the next chapter a methodology to achieve the proper data exchange format is developed with use of newer information technology advancements.

Chapter 3. From Relational to XML-Based Data Organization

In previous chapter, it was concluded that AGS has been successful and better adopted by the community compared to other efforts. However, AGS format is not updated and can be improved. In this chapter, based on in-depth analysis of existing conversion algorithms, and AGS data dictionary an XML-based data organization is introduced. This format is referred to as eXtensible AGS (XAGS). Careful considerations on the details of the new structure are discussed. The structure of XAGS 2.0 and its main elements are presented in more detail with the help of schema diagrams and sample instances. XAGS is directed to support the organization and exchange of geotechnical data. This approach can be expanded to encompass all types of geotechnical and geological information and in fact can be generalized to a wide scope of scientific and engineering data and knowledge.

3.1. AGS Data Structure Analysis

3.1.1. AGS Conceptual Modeling

AGS can be conceptually modeled using the Extended Entity Relationships (EER) data model. The Entity Relationships (ER) data model visually represents data objects as a set of entities and association between entities (Elmasri and Navathe 2003). The extended version of ER, also referred to as EER, is commonly used for database design. Figure 3.1

shows the partial EER diagram of AGS main entities and omits other entities and non-identifier attributes for simplicity. The entities shown are classification and shear tests, samples, holes, and attached files. The laboratory tests relate hierarchically to samples and boreholes; attached files are separate entities that can be connected to any level in the hierarchy.

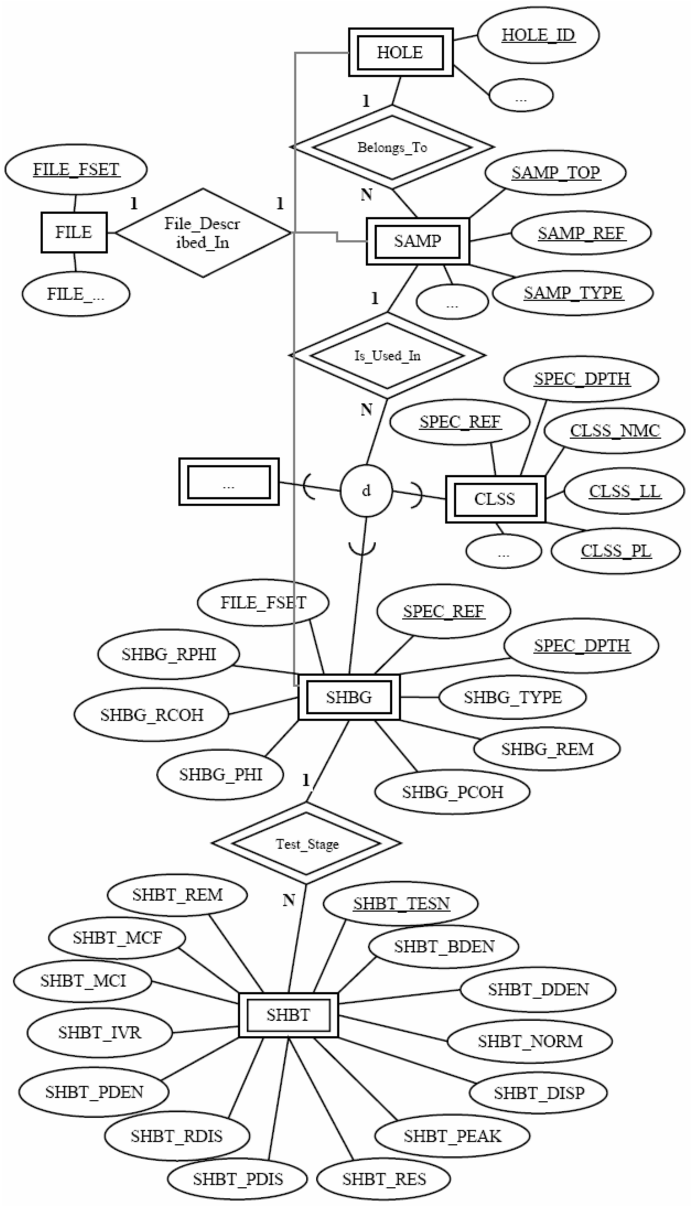


Figure 3.1 Partial EER diagram of AGS geotechnical laboratory tests including borehole (HOLE), sample (SAMP), classification (CLSS), shear tests (SHBG, SHBT) and file attachments (FILE).

As an ER diagram, Figure 3.2 details the relationship between borehole and specimen, and omits all the attributes that are not identifiers. Entities and relationships are distinguishable using various keys. A combination of one or more attributes that allow unique identification is called a super key. A minimal super key chosen to identify the entity set is the primary key of that entity set. As depicted in Figure 3.2, the primary key for HOLE is HOLE_ID. The ER model classifies entity sets as independent or dependent. An independent (strong) entity set does not rely on another entity set for identification, meaning it has enough attributes to form a primary key, such as HOLE entity. A dependent (weak) entity set relies on another entity set for identification; it has not enough attributes to form the primary key independently. For example, in the borehole-lab tests relation, SAMP and SPEC are weak entity sets correspondingly relying on HOLE and SAMP entity sets to define their primary keys. This analysis shows that there is a strong physical relationship between different components of a site investigation. These relationships are very clearly understood by the community.

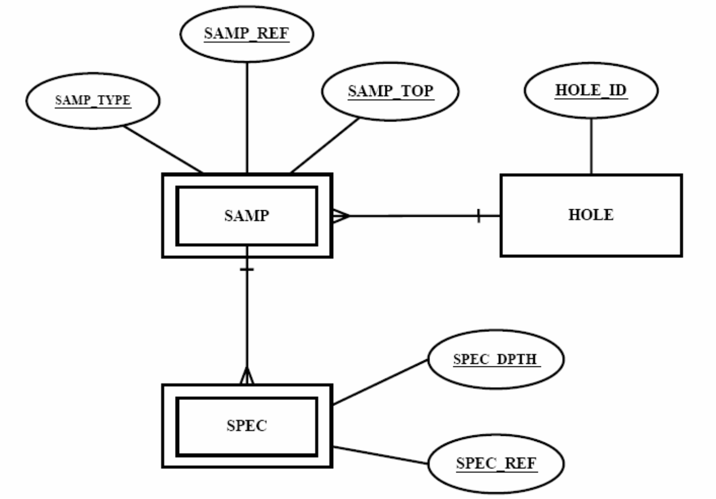


Figure 3.2 Entity relation diagram for AGS borehole components: borehole (HOLE), Sample (SAMP) and specimen (SPEC)

3.1.2. AGS Data Group Types

Table 3.1 lists the 27 main AGS data groups, which are PROJ, HOLE, ISPT, STCN, SAMP, CBRG, CBRT, CHLK, CLSS, CMPG, CMPT, CNMT, CONG, CONS, FRST, GRAD, MCVG, MCVT, PTST, RELD, ROCK, SHBG, SHBT, SUCT, TNPC, TRIG and TRIX. These 27 data groups represent the data most commonly used for site investigations in geotechnical projects. Some of the listed data groups are further discussed in this chapter.

The information of the data groups can be classified as two types for further structural classifications: name-value type, and serialized data type. Name-value type consists of a name and value. For example, in data group HOLE, HOLE_STAR contains date of start of excavation; HOLE_STAR is the name and 1992-11-10 is the value. On the other hand, serialized data type consists of a series of measurements over intervals. For instance, for STCN (Static Cone Penetration Test) data group, STCN_RES shows the measured tip resistance over intervals of STCN_DPTH. At the same time, the geotechnical information data groups have nine distinctive classes of immediate upper level data groups. These immediate upper level data groups can be: PROJ, HOLE, SAMP, and the corresponding general lab test (SHBG, CBRG, CMPG, CONG, MCVG and TRIG). Hence, all the geotechnical information data groups are one of the eighteen combinations presented in Table 3.2. Table 3.3 presents the combination type of each data group.

Table 3.1 The main data groups in AGS, their definitions and immediate higher level data group

AGS Data Group	Main Definition	Higher Level Data Group
PROJ	Project is a collection of holes belonging to the same project	---
HOLE	Holes drilled in the ground for the purpose of site investigation	PROJ
ISPT	SPT gives an indication of soil density and strength	HOLE
STCN	CPT measures the resistance of a cone that penetrates the ground	HOLE
SAMP	Sample is a segment from a hole taken for lab tests	HOLE
CBRG	California Bearing Ratio general information	SAMP
CBRT	CBR test results	CBRG
CHLK	Chalk test results	SAMP
CLSS	Classification test results	SAMP
CMPG	Compaction test general information	SAMP
CMPT	Compaction test results	CMPG
CNMT	Contamination and chemical test results	SAMP
CONG	Consolidation test general information	SAMP
CONS	Consolidation test results	CONG
FRST	Frost susceptibility test results	SAMP
GRAD	Particle size distribution analysis data	SAMP
MCVG	Moisture content value test general information	SAMP
MCVT	Moisture content value test results	MCVG
PTST	Permeability test results	SAMP
RELD	Relative density test results	SAMP
ROCK	Rock testing results	SAMP
SHBG	Shear box test general information	SAMP
SHBT	Shear box test results	SHBG
SUCT	Suction test results	SAMP
TNPC	Ten percent fines test results	SAMP
TRIG	Triaxial test general information	SAMP
TRIX	Triaxial test results	TRIG

Table 3.2 Possible combinations of immediate upper level data groups and data types

Immediate Upper Level Data Group	Name-value Data Only	Name-value Data and Serialized Data
PROJ	1	2
HOLE	3	4
SAMP	5	6
CBRG	7	8
CMPG	9	10
CONG	11	12
MCVG	13	14
SHBG	15	16
TRIG	17	18

3.1.3. Relational Characteristics of AGS

Relational Data Base Management Systems (RDBMS) are presently the most commonly used format for data archival. Relational database management systems (RDBMS) have a solid mathematical foundation (Codd, 1990; and Atzeni et. al., 1993; and Elmasri and Navathe, 2003). RDBMS store and handle information using the relational database management model. RDBMS manages data in tables. By definition, a table is an object that is defined and used to store data. Tables contain fields (or columns) that store different kinds of data. In the table, a primary key is used to define one or more fields that have a unique value for each record. Primary key can be used to link other table that contains relative information. Rules can be defined to ensure RDBMS data integrity. RDBMS provide many ways to work with data, especially using the powerful database language SQL (Structured Query Language). SQL is the common language of client/server database management (Jennings, 2007). By querying a table with SQL language, data information can be extracted.

Table 3.3 AGS data groups combination types based on Table 3.2 combinations

Data Group	Combination Number	Data Group	Combination Number
HOLE	1	MCVG	5
CBRG	5	MCVT	13
CBRT	7	PTST	5
CHLK	5	RELD	5
CLSS	5	SAMP	3
CMPG	5	ROCK	5
CMPT	9	SHBG	5
CNMT	5	SHBT	15
CONG	5	STCN	4
CONS	11	SUCT	5
FRST	5	TNPC	5
GRAD	5	TRIG	5
ISPT	4	TRIX	17

AGS is relational in nature. Each data group is represented as a table in AGS data format, similar to a Relational Data Model structure. The AGS data are written in the file through a sequence of data groups, which contain the related data. Within each data group, the data items are contained in data fields defined by the data dictionary. Each data field holds a single data variable. The order of the data fields within each data group is determined by a line following group identifier, which contains a set of data headings. Figure 3.3 by Zand (2005) demonstrates the schematic structure of an AGS data group.

As it can be seen in the figure, the headings line shall be followed by a "UNIT" line, which contains the units used in the data fields. Unit line is required for all data groups.

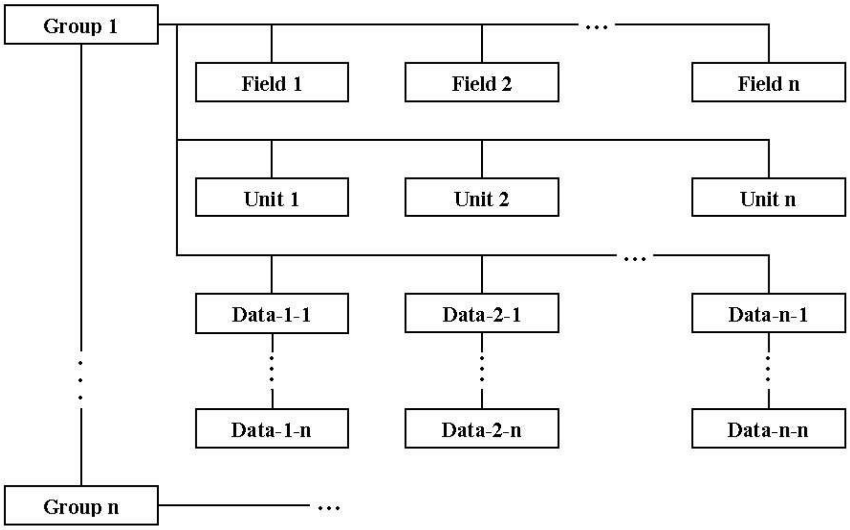


Figure 3.3 Schematic structure of AGS data files and data groups (Zand, 2005). Data structure is tabular with first row listing the fields, second row listing the units and following rows containing actual values.

Table 3.4 lists the keys of AGS laboratory tests, including foreign keys labeled A and B. Key A aggregates specific information from SAMP to create a unique identifier for SAMP. Key B adds additional information to Key A to identify specimens. The HOLE, SAMP, SHBG, SHBT entities are defined as relational tables and their column (attribute) names are listed. Foreign keys, including HOLE_ID, SAMP_TOP, SAMP_REF, SAMP_TYPE, SPEC_REF and SPEC_DPTH build up the relationship between these relational tables. FILE table can be related to any of the other tables to describe the external files linked to the real-world entity or relation.

Table 3.4 AGS geotechnical laboratory tests and the classification of identifiers into foreign and regular keys and their parent groups

Table	Foreign Keys	Keys	Parent group
HOLE	---	HOLE_ID	---
SAMP	HOLE_ID	SAMP_TOP, SAMP_REF, SAMP_TYPE	HOLE
CBRG	A	SPEC_REF, SPEC_DPTH	SAMP
CBRT	B	CBRT_TESN	CBRG
CHLK	A	SPEC_REF, SPEC_DPTH, CHLK_TESN	SAMP
CLSS	A	SPEC_REF, SPEC_DPTH	SAMP
CMPG	A	SPEC_REF, SPEC_DPTH	SAMP
CMPT	B	CMPT_TESN	CMPG
CNMT	A	SPEC_REF, SPEC_DPTH, CNMT_TYPE, CNMT_TTYP	SAMP
CONG	A	SPEC_REF, SPEC_DPTH	SAMP
CONS	B	CONS_INCN	CONG
FRST	A	SPEC_REF, SPEC_DPTH	SAMP
GRAD	A	SPEC_REF, SPEC_DPTH, GRAD_SIZE	SAMP
MCVG	A	SPEC_REF, SPEC_DPTH	SAMP
MCVT	B	MCVT_TESN	MCVG
PTST	A	SPEC_REF, SPEC_DPTH, PTST_TESN	SAMP
RELD	A	SPEC_REF, SPEC_DPTH	SAMP
ROCK	A	SPEC_REF, SPEC_DPTH	SAMP
SHBG	A	SPEC_REF, SPEC_DPTH	SAMP
SHBT	B	SHBT_TESN	SHBG
SUCT	A	SPEC_REF, SPEC_DPTH	SAMP
TNPC	A	SPEC_REF, SPEC_DPTH	SAMP
TRIG	A	SPEC_REF, SPEC_DPTH	SAMP
TRIX	B	TRIX_TESN	TRIG

A= HOLE_ID, SAMP_TOP, SAMP_REF, SAMP_TYPE
B= A+SPEC_REF, SPEC_DPTH

The relational database was implemented using MS Access and the same terminology as AGS. Figure 3.4 shows a part of the relational database built in Access based on AGS. RDBMS have limitations in modeling data, as pointed out by many researchers such as, Saake et al. (1995). RDBMS has a fixed structure, and is difficult to maintain. There are challenges in putting all types of data into a fixed table format. Database needs to be recompiled each time new type of data is added. The user needs to have a basic understanding of database structure for performing a SQL and extract data from complicated RDBMS. And the most important of all is that relational data models are not appropriate for exchange of data.

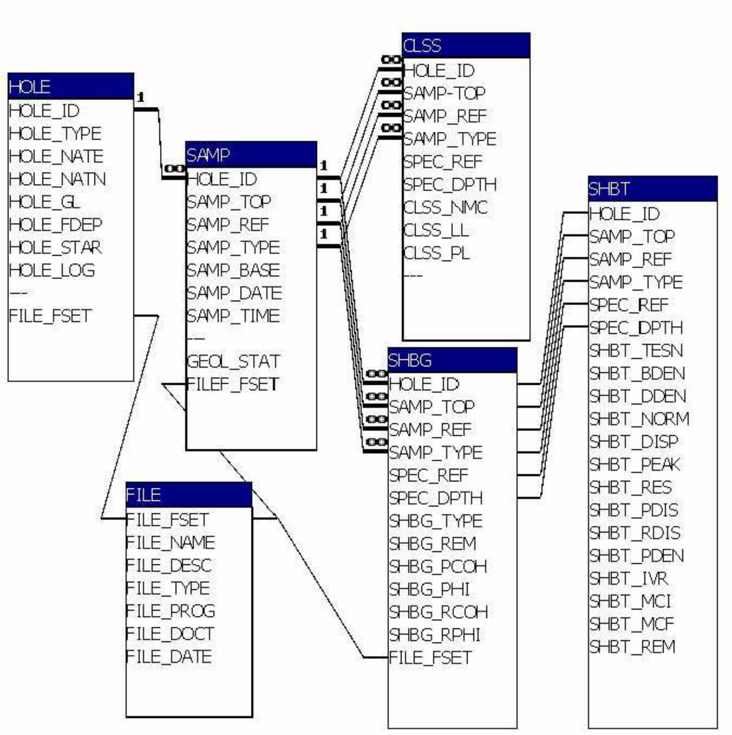


Figure 3.4 Partial AGS relational database diagram including borehole (HOLE), sample (SAMP), classification (CLSS), shear tests (SHBG, SHBT) and file attachments (FILE).

3.2. Transformation of Relational Information to XML Format

3.2.1. Extensible Markup Languages (XML)

Although relational databases are dominant in archiving data, the eXtensible Markup Language (XML, 2003) has become an emerging universal format for exchanging data on the World Wide Web and has gained a wide acceptance from the computer industry, Microsoft, AutoCAD, IBM and Oracle. The first official XML specification was published in 1998. XML is a markup language that uses DTD or XML schemas to define the structure, content and semantics of data models. XML data can be sent to Internet browsers that use stylesheet-extra information and be translated into other formats such as HTML (Hyper-Text Markup Language) (Boumphrey and Tittel, 2000). XML allows to clearly separate content from form (appearance). It is text-oriented and extensible.

The greatest advantage of XML is its ability to define an interchange format for transferring data. Additionally, XML data is stored in text format. Hence, upgrading or expanding to new operating systems, new applications, or new browsers, without losing data is possible. In addition to other advantages listed above, many new Internet languages are created with XML, e.g. XHTML, WSDL for describing available web services, WAP and WML as markup languages for handheld devices, RSS languages for news feeds, RDF and OWL for describing resources and ontology, and SMIL for describing multimedia on the web. Web services are self-describing and self-contained web-hosted applications that can be invoked through the Internet. The clients to web services are other computer applications or data users that communicate with web service using HTTP protocol. Client connects to the web service using XML standards including SOAP, WSDL and UDDI. XML in effect provides a structured syntactic self-describing specification of data to be exchanged, though, having a low level of semantic information.

XML uses elements and attributes, which are defined using markup rules in an XML schema (XML Schema, 2000). An XML schema typically expresses terms of constraints on the structure and content of documents of that type, above and beyond the basic syntactical constraints imposed by XML itself. These constraints are generally expressed using some combination of grammatical rules governing the order of elements, Boolean predicates that the content must satisfy, data types governing the content of elements and attributes, and more specialized rules such as uniqueness and referential integrity constraints (Vlist, 2002). A schema is a separate document that its location is inserted at the head of XML documents. XML Schema also provides a mechanism for creating a relationship between elements through key and keyref.

Future applications will exchange their data in XML. It is predicted that the future will give us word processors, spreadsheet applications and databases that can read each other's data in XML format, without any conversion utilities in between.

3.2.2. Conversion Algorithms

Several conversion algorithms have been proposed to exchange relational data as XML documents. The simplest relational-to-XML translation method, called Flat Translation (FT), translates the flat relational model to a flat XML model in a one-to-one manner.

The shortcomings of FT was remedied by introducing Nesting based Translation algorithm (NeT) by Lee et al. (2001, in press). This algorithm derives a nested structure from a flat relational schema by repeatedly applying the nest operator so that the resulting XML schema becomes hierarchical. NeT assigns the nesting of tables without any input from human experts with the objective of minimizing the number of cross-references between hierarchical substructures. The nest operator requires scanning of the entire set of tuples in a given table; consequently it can be quite time-consuming. Although NeT infers hidden characteristics of data by nesting, it is only applicable to a single table at a time. Therefore, it is unable to capture the overall picture of relational schema where multiple relations are interconnected.

Constraint Preserving (Liu et al. 2003, 2006) Algorithms look into the hierarchical mapping, by considering inclusion dependencies during the transformation. They merge multiple inter-connected relations into a hierarchical parent-child structure in the final XML schema by eliminating foreign keys used in the relational database. The parent table captures the core meaning of the schema and the child table captures the auxiliary information. Even though the transformation algorithms try to preserve integrity constraints, avoid data redundancy and explore the nested structures, domain-knowledge expertise is required to be able to explore all the nested structures and find a better mapping based on the semantics and usage of the underlying data.

3.2.3. Principles of Transformation

AGS nomenclature (data dictionary) is generally preserved and used for the XML-based elements unless otherwise stated. It was noted in chapter 2 that AGS is conforming to British Standards. However, at the time of conducting this research there was no well defined and well accepted data dictionary available for ASTM. Development of such data dictionary is a very large scope and time consuming work that needs the input of the community. Once such a data dictionary is developed, the same guidelines introduced in this dissertation can be used to develop another version of XAGS exchange format conforming to ASTM standards. Additionally, the same unit enumeration for AGS, presented in chapter two, will be used for XAGS.

3.3. XML-Based Data Organization

3.3.1. Preliminary Model: Implementation of Basic Mapping Rules

A preliminary XML schema was generated from the relational data using the automatic hierarchical mapping option in XMLSpy 2004. Due to the length of the schema, only a partial diagram is shown in Figure 2.7. Although this schema preserves all relationships, it is relatively complicated due to the presence of numerous key/keyrefs and XPaths required for building up relationships. In addition, the basic rules used as the backbone of the XMLSpy mapping cannot distinguish the foreign keys and keep them in each level of the hierarchy. This results in duplication of attributes and a complicated schema. This complexity in XML schema can be decreased by selecting hierarchical structures.

Alternate nested structures should be explored by domain experts so that the referential integrity constraints are enforced in the schema.

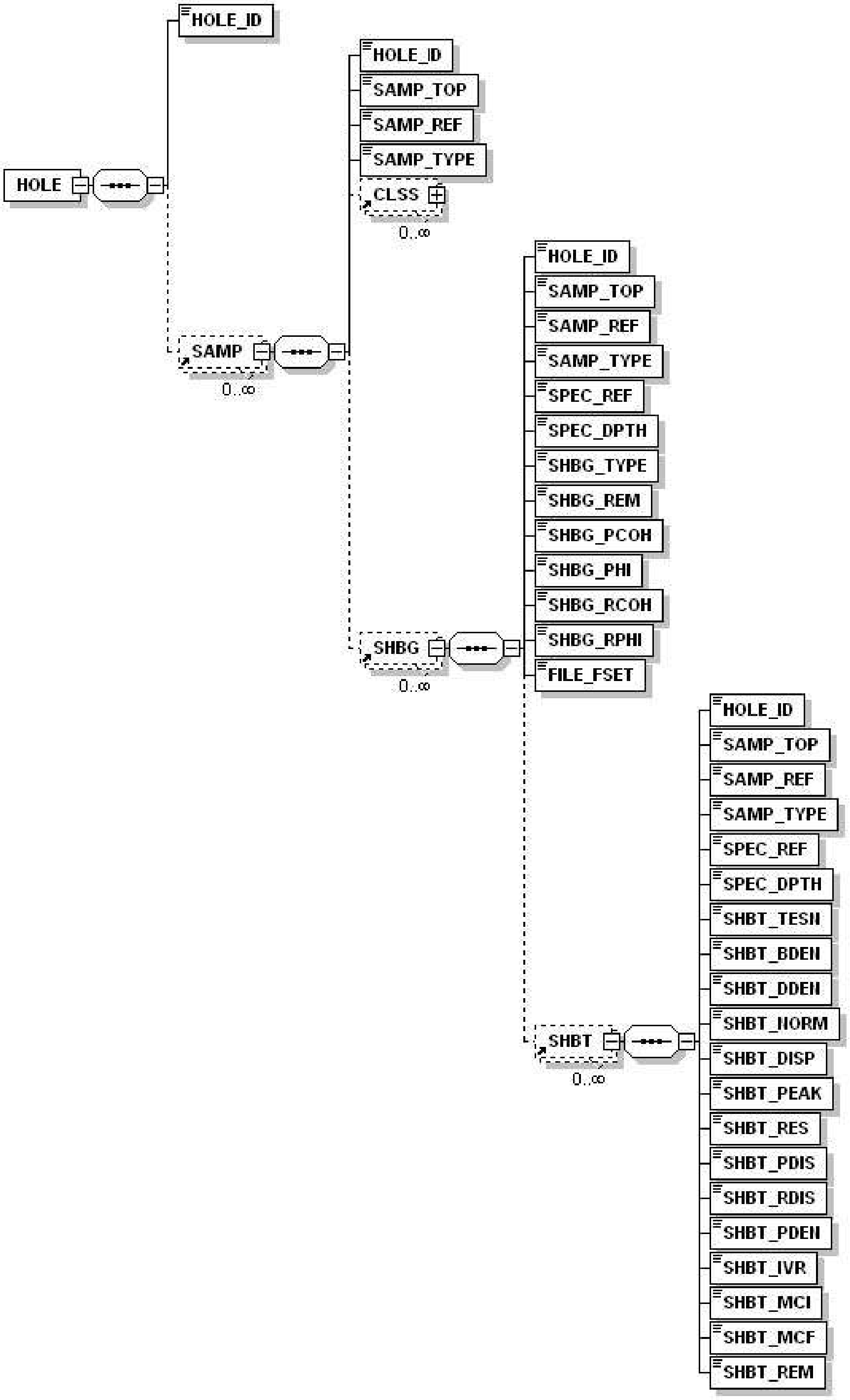


Figure 3.5 Partial schema diagram of preliminary model obtained by automatic implementation of basic mapping rules without domain experts' modification from AGS.

3.3.2. XAGS 1.0: Exploring Nested Structure

A more efficient XML schema, which is referred to as XAGS 1.0, was proposed by taking advantage of the AGS hierarchical structure. After studying the relationships between physical components of AGS, the hierarchical characteristic of the geotechnical laboratory tests is conceptualized in Figure 3.6.

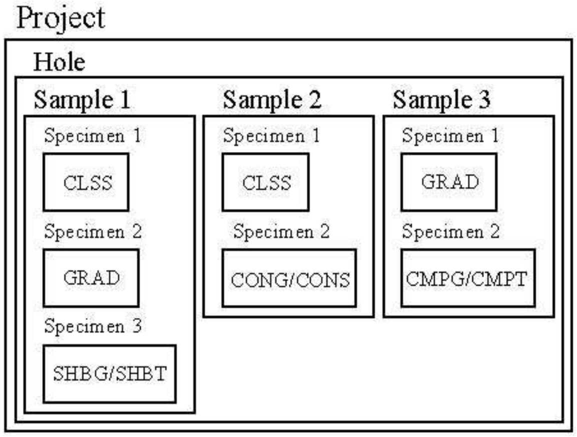


Figure 3.6 Demonstration of hierarchical characteristic of AGS geotechnical laboratory test components

Considering this nested structure, XAGS describes laboratory tests in two levels: (1) main entities containing borehole and lab tests relations (including HOLE, SAMP, and the name list of LAB_TESTS); and (2) schema for individual laboratory tests. The containment and nested structure of AGS is critical to omit unnecessary relationships and reduces the complexity of XML model automatically generated from relational data models. Figure 3.7 presents partial XAGS borehole-lab tests schema diagram. This diagram shows the translation of the nested structure in Figure 3.6 into the XML tree structure. As depicted, for each project containing borehole(s), each borehole contains samples and each sample can be divided into several specimens for conducting different laboratory tests.

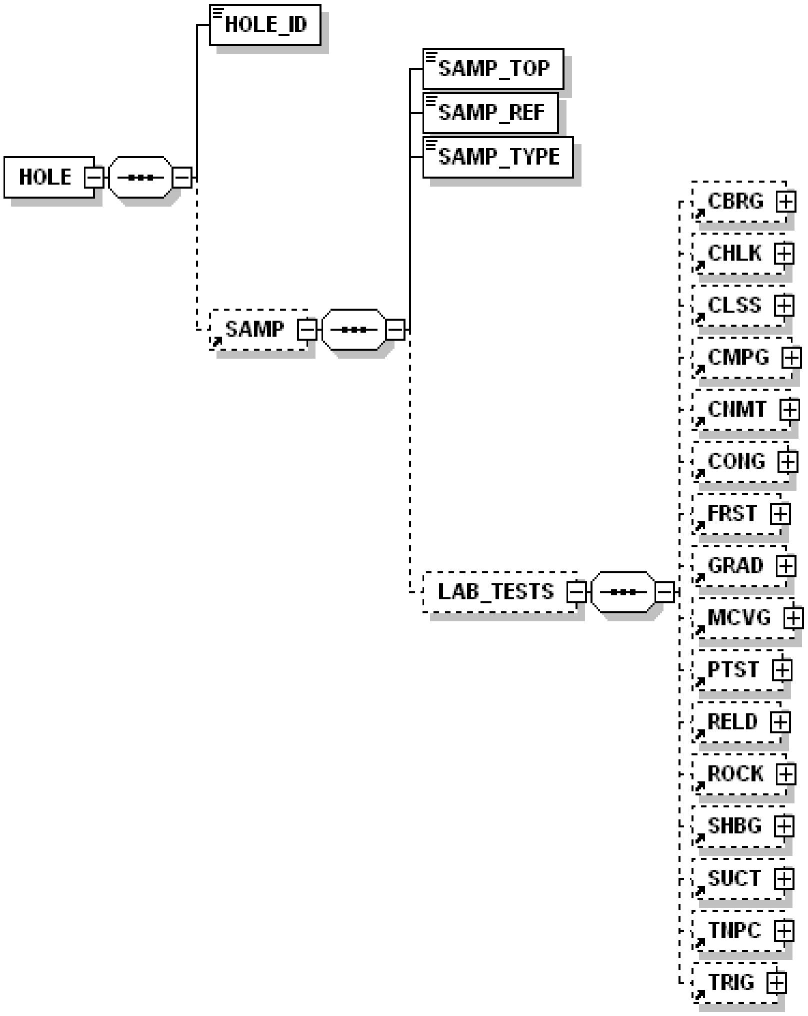


Figure 3.7 Partial XAGS 1.0 schema diagram illustrating Borehole- Laboratory tests relation

Figure 3.8 shows the tree structure for shear test data as an example of the laboratory tests. Comparing the XAG 1.0 schema to the automatically derived schema in the previous section shows that in XAGS 1.0 the relations are preserved while the duplicated attributes are omitted, e.g. HOLE_ID, SAMP_TOP, SAMP_REF, SAMP_TYPE, SPEC_REF, and SPEC_DPTH.

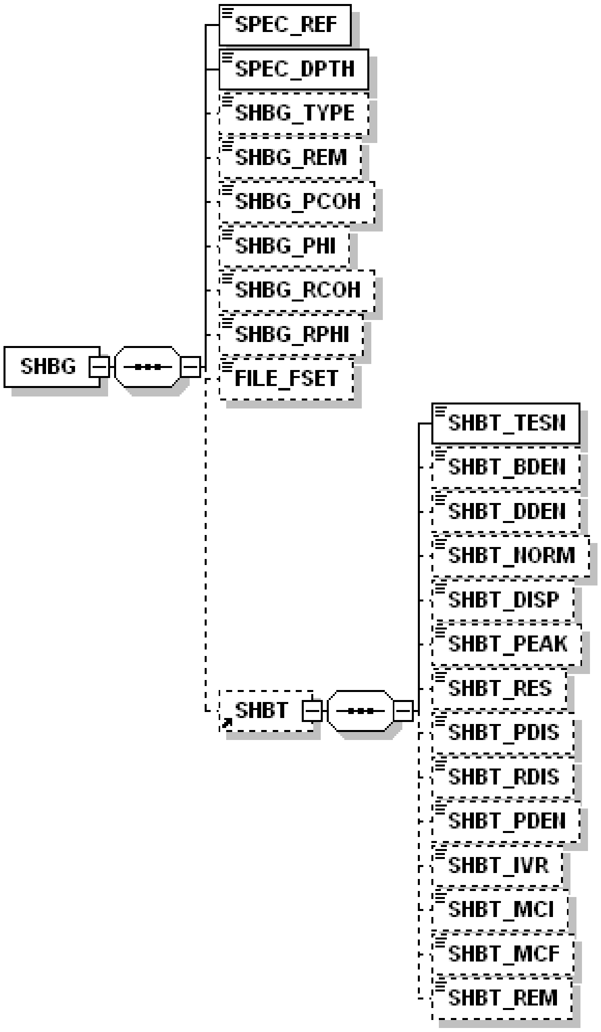
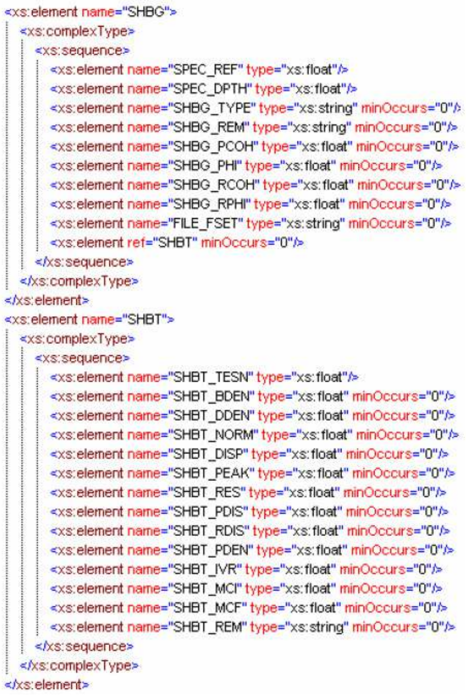


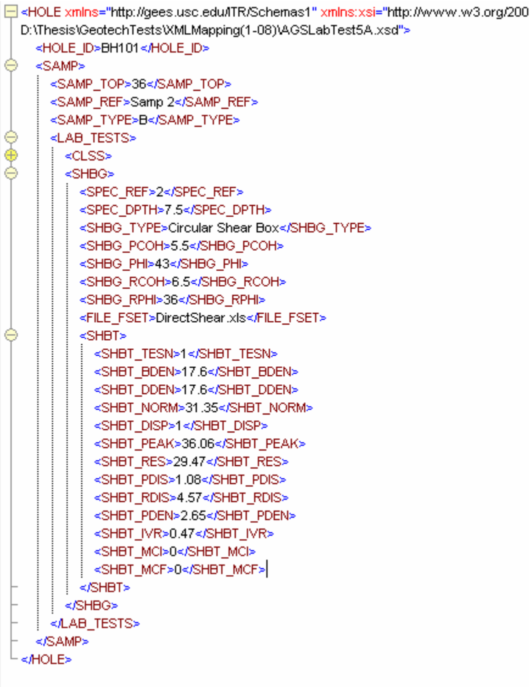
Figure 3.8 Partial XAGS 1.0 schema diagram illustrating Shear test as an example of laboratory tests

Figure 3.9 shows the actual proposed schema for Shear Box Test in XAGS 1.0, and Figure 3.10 is an example of XAGS 1.0 data instance corresponding to that schema.



```
<xs:element name="SHBG">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SPEC_REF" type="xs:float"/>
      <xs:element name="SPEC_DPTH" type="xs:float"/>
      <xs:element name="SHBG_TYPE" type="xs:string" minOccurs="0"/>
      <xs:element name="SHBG_REM" type="xs:string" minOccurs="0"/>
      <xs:element name="SHBG_PCOH" type="xs:float" minOccurs="0"/>
      <xs:element name="SHBG_PHI" type="xs:float" minOccurs="0"/>
      <xs:element name="SHBG_RCOH" type="xs:float" minOccurs="0"/>
      <xs:element name="SHBG_RPHI" type="xs:float" minOccurs="0"/>
      <xs:element name="FILE_FSET" type="xs:string" minOccurs="0"/>
      <xs:element ref="SHBT" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SHBT">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SHBT_TESN" type="xs:float"/>
      <xs:element name="SHBT_BDEN" type="xs:float" minOccurs="0"/>
      <xs:element name="SHBT_DDEN" type="xs:float" minOccurs="0"/>
      <xs:element name="SHBT_NORM" type="xs:float" minOccurs="0"/>
      <xs:element name="SHBT_DISP" type="xs:float" minOccurs="0"/>
      <xs:element name="SHBT_PEAK" type="xs:float" minOccurs="0"/>
      <xs:element name="SHBT_RES" type="xs:float" minOccurs="0"/>
      <xs:element name="SHBT_PDIS" type="xs:float" minOccurs="0"/>
      <xs:element name="SHBT_RDIS" type="xs:float" minOccurs="0"/>
      <xs:element name="SHBT_PDEN" type="xs:float" minOccurs="0"/>
      <xs:element name="SHBT_IVR" type="xs:float" minOccurs="0"/>
      <xs:element name="SHBT_MCI" type="xs:float" minOccurs="0"/>
      <xs:element name="SHBT_MCF" type="xs:float" minOccurs="0"/>
      <xs:element name="SHBT_REM" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Figure 3.9 Proposed shear test schema in XAGS 1.0 as an example of laboratory tests



```
<HOLE xmlns="http://gees.usc.edu/ITR/Schemas1" xmlns:xsi="http://www.w3.org/200
D:\Thesis\GeotechTests\XMLMapping(1-08)\AGSLabTest5A.xsd">
  <HOLE_ID>BH101</HOLE_ID>
  <SAMP>
    <SAMP_TOP>36</SAMP_TOP>
    <SAMP_REF>Samp 2</SAMP_REF>
    <SAMP_TYPE>B</SAMP_TYPE>
    <LAB_TESTS>
      <CLSS>
      <SHBG>
        <SPEC_REF>2</SPEC_REF>
        <SPEC_DPTH>7.5</SPEC_DPTH>
        <SHBG_TYPE>Circular Shear Box</SHBG_TYPE>
        <SHBG_PCOH>5.5</SHBG_PCOH>
        <SHBG_PHI>43</SHBG_PHI>
        <SHBG_RCOH>6.5</SHBG_RCOH>
        <SHBG_RPHI>36</SHBG_RPHI>
        <FILE_FSET>DirectShear.xls</FILE_FSET>
        <SHBT>
          <SHBT_TESN>1</SHBT_TESN>
          <SHBT_BDEN>17.6</SHBT_BDEN>
          <SHBT_DDEN>17.6</SHBT_DDEN>
          <SHBT_NORM>31.35</SHBT_NORM>
          <SHBT_DISP>1</SHBT_DISP>
          <SHBT_PEAK>36.06</SHBT_PEAK>
          <SHBT_RES>29.47</SHBT_RES>
          <SHBT_PDIS>1.08</SHBT_PDIS>
          <SHBT_RDIS>4.57</SHBT_RDIS>
          <SHBT_PDEN>2.65</SHBT_PDEN>
          <SHBT_IVR>0.47</SHBT_IVR>
          <SHBT_MCI>0</SHBT_MCI>
          <SHBT_MCF>0</SHBT_MCF>
        </SHBT>
      </SHBG>
    </LAB_TESTS>
  </SAMP>
</HOLE>
```

Figure 3.10 Shear test data in XAGS 1.0 format as an example of laboratory tests

3.3.3. XAGS 2.0: Building Relationships with Identifiers

Even though XAGS 1.0 would be an optimum way for final archival stage of geotechnical data, it is not the best option for data exchange, since carrying all the envelopes will make the integrating process of separate files (e.g., data subsets) manual and prone to human error. Hence another approach was considered for the second version of XAGS. Instead of using the nested structure, an independent structure that connects the objects with the help of unique IDs (key, keyref and XPath) was reconsidered. Keys are required for each object and each object will point to its immediate upper level data group with keyref technique. XAGS 2.0 implementation is discussed in great detail in the next section. XAGS enforces the same data integrity as AGS, but in addition its XML schema checks for data compatibility with predefined schemas.

3.4. XAGS 2.0 Implementation

XAGS 2.0 schema is presented in four schema files named: XAGS2.xsd, ISPT2.xsd, STCN2.xsd and LABTests2.xsd. The main schema, XAGS2.xsd, contains the global structure and PROJ, HOLE, and SAMP elements. SPT testing (ISPT2.xsd), CPT testing (STCN2.xsd) and laboratory testing (LABTests2.xsd) are imported with inclusion method ("include"). Element names are all uppercase letter acronyms (e.g., PROJ) as defined in AGS data dictionary. Figure 3.11 demonstrates global structure of XAGS 2.0 schema. Table 3.1 lists the 27 elements from AGS data groups implemented in XAGS 2.0 model. Appendix A presents all XAGS 2.0 schema files. Some of the listed elements are further explained in this chapter.

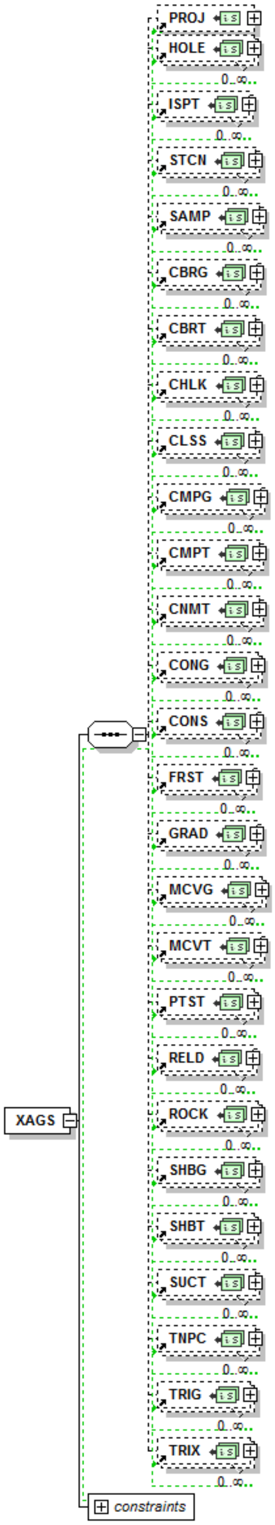
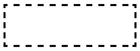
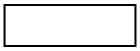
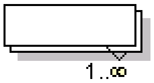
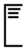
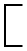
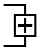
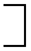
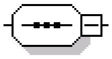
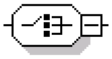
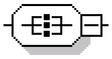
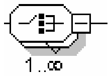


Figure 3.11 Schema diagram of XAGS 2.0 structural overview containing project, borehole, SPT, CPT, sample, and laboratory tests

3.4.1. Schema Symbols

For schema editing, presentation of schema diagrams and grid view of XML instances, XMLSpy (2010) XML editor is used. In the presented schema diagrams, only the elements are drawn and attributes are not visible. As presented in Table 3.5, optional elements are displayed with dashed borderlines and required elements with a solid line. Repeated elements are indicated by two stacked boxes. The cardinality of the element is indicated at the bottom right side of the elements. The content model of elements is symbolized on the left and right side of the element boxes as shown in Table 3.5. The left side indicates whether the element contains a simple type, such as text, numbers, dates or a complex type with further elements. A plus sign at the right side of the element symbol indicates that it contains child elements. If an element refers to a complex global type, the type is shown with a border and yellow background, as it will be seen in Figure 3.26. Elements can be defined as global elements in the schema. Global elements can then be reused in multiple places. In these cases, elements directly refer to the global element already defined with an arrow in the lower left corner. The schema diagrams presented in this chapter represent parts of XAGS 2.0 data exchange schema using the above mentioned symbols. Table 3.5 describes the symbolic representations used in the schema diagrams through out this chapter.

Table 3.5 Symbolic representation of XML schema diagrams used in this dissertation (after XMLSpy)

Symbol	Explanation
	Optional element. The dashed borderline means the element may appear once or not at all.
	Required single element. The solid borderline means the element must appear once.
	Element with maximum occurrence greater than one. It must appear at lease once or many times.
	Simple content
	Complex content
	Complex content with child elements
	No element content (simple type, attributes only, or empty element)
	Sequence of child element(s) which must be in a specific order.
	Choice of child element(s) which must be from a list but in any order.
	The "all" model, in which the sequence of elements is not fixed.
	Choice which must appear at least once or many times

3.4.2. Type, Simple Type and Complex Type

W3C XML schema provides “type”, “simpletype” and “complextype” for defining data type and structure as used in XAGS 2.0. “type” is used in XML schema to define built-in datatypes. Built-in datatypes are those which are defined in XML schema specification, such as string for character strings, boolean for binary-valued logic {true, false}, decimal for a subset of the real numbers represented by decimal numerals, float and double for IEEE single-precision 32-bit and double-precision 64-bit floating point types (IEEE 754-1985, 2006), date for intervals of exactly one day in length, and integer derived from decimal by disallowing the trailing decimal point (XML schema, 2000). Then, user-derived datatypes (simpletype) are derived by schema designers from the built-in datatypes by restriction, list and union (Vlist, 2002). The simple datatypes describe the value of an element or an attribute. Complex types (complextype) are, on the contrary, a description of the content model. They use “simpletype” to describe their leaf element nodes and attribute values.

The 27 elements listed in Table 3.1, use 4 built-in data types: string, date, float, and integer. Simpletype is used to define XAGS specific data types from the built-in datatypes with the help of restriction, list and union derivation methods as well. XAGS 2.0 utilizes simpletype and complextype to define the association types for unit enumeration based on AGS unit enumeration. The user-derived types to choose from are: LengthType, Temperature Type, PressureType, FlowType, DensityType, VelocityType, ForceType, PercentageType and MiscellaneousType. The unit enumeration types can be seen within XAGS2.xsd schema file presented in Appendix A.

3.4.3. Project

Project is a collection of holes belonging to the same engineering project. Project (PROJ) is called by borehole (HOLE) via keyref (PROJ_IDREF) attribute. Figure 3.12 demonstrates the schema diagram of PROJ complex type element in XAGS 2.0.

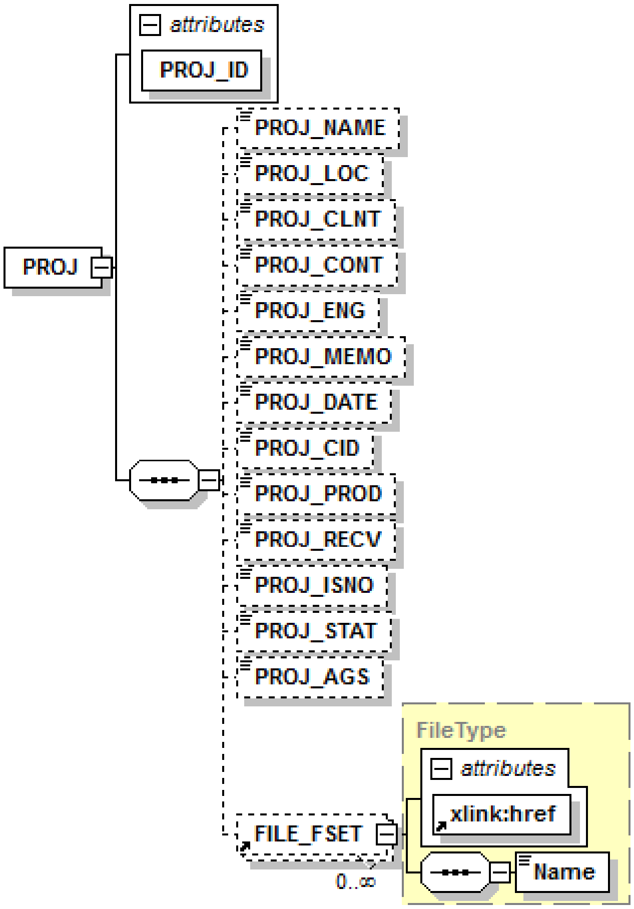
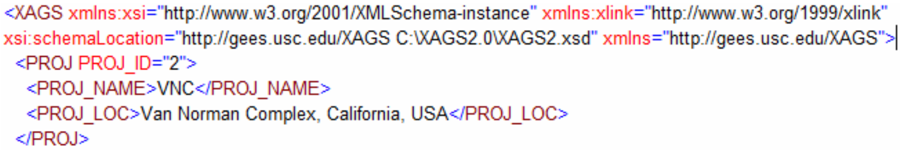


Figure 3.12 Schema diagram of PROJ complex type in XAGS 2.0

Figure 3.13 shows an instance of PROJ element at the beginning of a XAGS 2.0 XML instance. Elements correspond to markup tags and attributes stand for the values associated with specific tags. For instance, for the tag <PROJ PROJ_ID="2">, the element is 'PROJ' and the associated attribute is 'PROJ_ID', whose value is '2'.



```
<XAGS xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://gees.usc.edu/XAGS C:\XAGS2.0\XAGS2.xsd" xmlns="http://gees.usc.edu/XAGS">
  <PROJ PROJ_ID="2">
    <PROJ_NAME>VNC</PROJ_NAME>
    <PROJ_LOC>Van Norman Complex, California, USA</PROJ_LOC>
  </PROJ>
```

Figure 3.13 An instance of PROJ at the beginning of a XAGS 2.0 XML file

3.4.4. Borehole

Borehole (HOLE) is a single sampling station, from which soil materials are collected or described, or soil material properties are measured. Each Hole will refer to its parent Project with PROJ_IDREF. Figure 3.14 demonstrate the schema diagram of HOLE complex type in XAGS 2.0 and Figure 3.15 shows an instance of HOLE element within XAGS 2.0 XML file.

3.4.5. In Situ Tests

The Standard Penetration Test (SPT) and Cone Penetration Test (CPT) are modeled in second version of XAGS as prototypes of in-situ tests and are further described in this section.

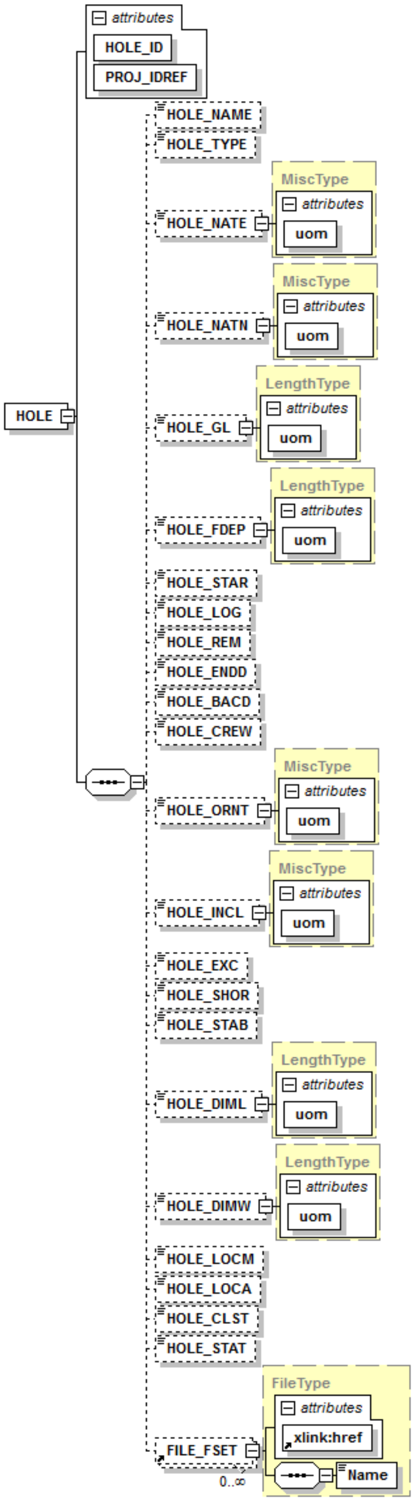
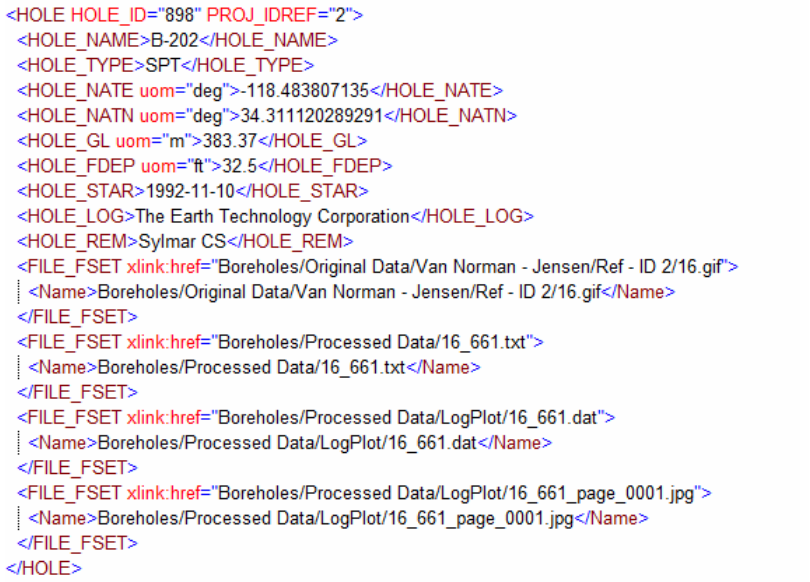


Figure 3.14 Schema diagram of HOLE complex type in XAGS 2.0



```
<HOLE HOLE_ID="898" PROJ_IDREF="2">
 <HOLE_NAME>B-202</HOLE_NAME>
 <HOLE_TYPE>SPT</HOLE_TYPE>
 <HOLE_NATE uom="deg">-118.483807135</HOLE_NATE>
 <HOLE_NATN uom="deg">34.311120289291</HOLE_NATN>
 <HOLE_GL uom="m">383.37</HOLE_GL>
 <HOLE_FDEP uom="ft">32.5</HOLE_FDEP>
 <HOLE_STAR>1992-11-10</HOLE_STAR>
 <HOLE_LOG>The Earth Technology Corporation</HOLE_LOG>
 <HOLE_REM>Sylmar CS</HOLE_REM>
 <FILE_FSET xlink:href="Boreholes/Original Data/Van Norman - Jensen/Ref - ID 2/16.gif">
  <Name>Boreholes/Original Data/Van Norman - Jensen/Ref - ID 2/16.gif</Name>
 </FILE_FSET>
 <FILE_FSET xlink:href="Boreholes/Processed Data/16_661.txt">
  <Name>Boreholes/Processed Data/16_661.txt</Name>
 </FILE_FSET>
 <FILE_FSET xlink:href="Boreholes/Processed Data/LogPlot/16_661.dat">
  <Name>Boreholes/Processed Data/LogPlot/16_661.dat</Name>
 </FILE_FSET>
 <FILE_FSET xlink:href="Boreholes/Processed Data/LogPlot/16_661_page_0001.jpg">
  <Name>Boreholes/Processed Data/LogPlot/16_661_page_0001.jpg</Name>
 </FILE_FSET>
</HOLE>
```

Figure 3.15 An instance of HOLE within a XAGS 2.0 XML file

SPT Test

The Standard Penetration Test (SPT) gives an indication of soil density and strength though the measurement of the resistance to dynamic penetration of hardened steel, split spoon sampler. During the SPT test the sampler is driven into the soil using hammer falling, and counting the number of blows required penetrating a specific depth based on the standard used such as BS or ASTM. As XAGS complies with AGS data dictionary SPT test is defined by the four letter acronym of ISPT. Figure 3.16 presents schema diagram of ISPT complex type in XAGS 2.0. Figure 3.17 shows an instance of ISPT element in a XAGS 2.0 XML file. Figure 3.18 demonstrates two instances in a grid view.

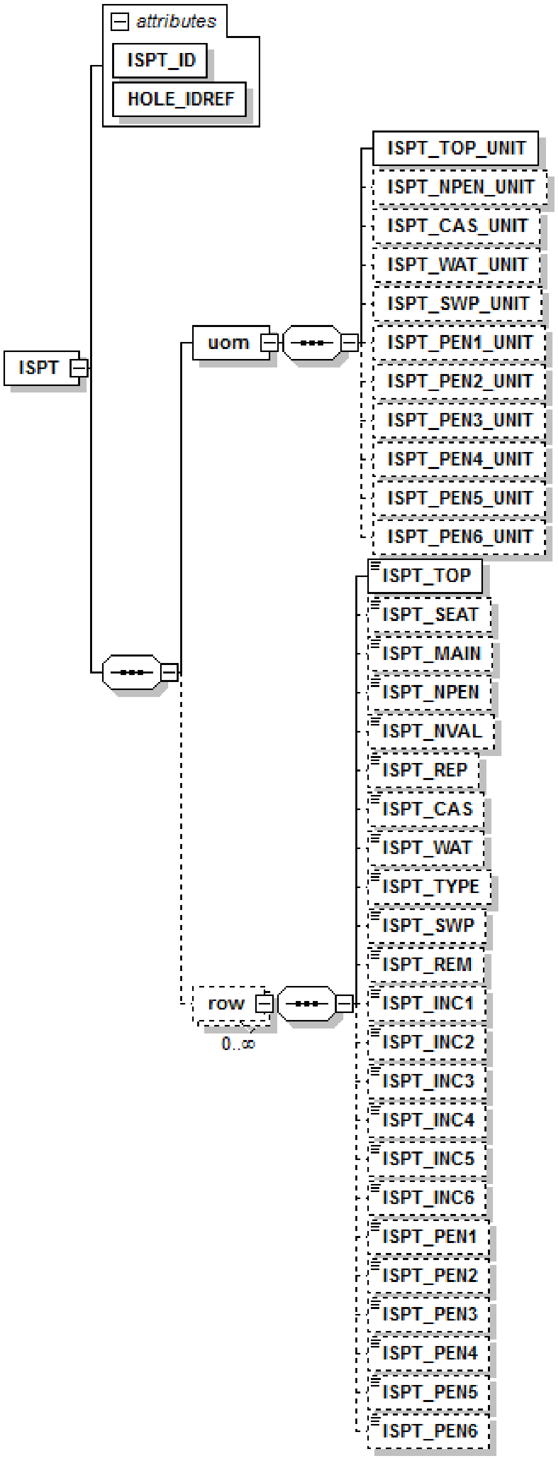
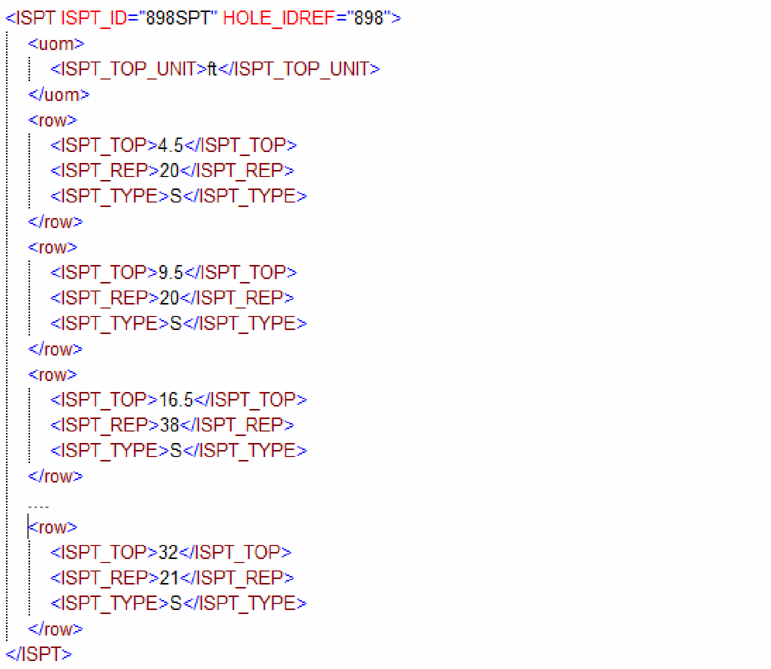
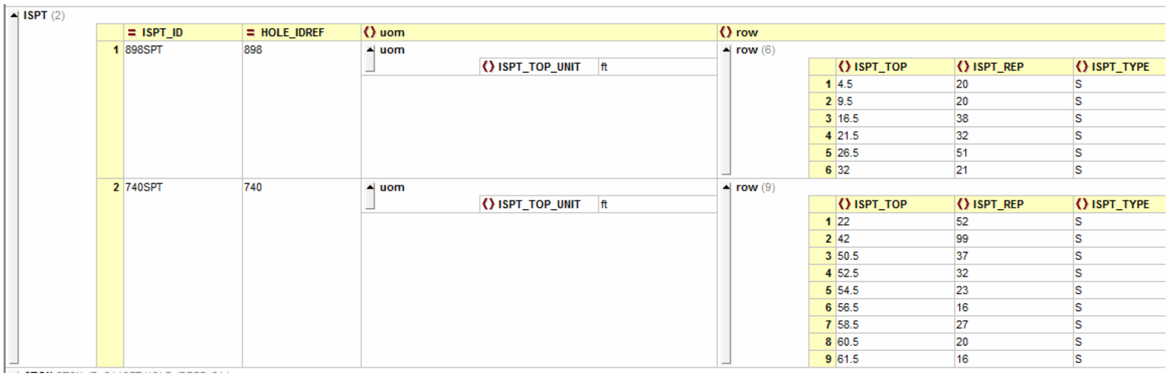


Figure 3.16 Schema diagram of ISPT complex type in XAGS2.0



```
<ISPT ISPT_ID="898SPT" HOLE_IDREF="898">
    <uom>
        <ISPT_TOP_UNIT>ft</ISPT_TOP_UNIT>
    </uom>
    <row>
        <ISPT_TOP>4.5</ISPT_TOP>
        <ISPT_REP>20</ISPT_REP>
        <ISPT_TYPE>S</ISPT_TYPE>
    </row>
    <row>
        <ISPT_TOP>9.5</ISPT_TOP>
        <ISPT_REP>20</ISPT_REP>
        <ISPT_TYPE>S</ISPT_TYPE>
    </row>
    <row>
        <ISPT_TOP>16.5</ISPT_TOP>
        <ISPT_REP>38</ISPT_REP>
        <ISPT_TYPE>S</ISPT_TYPE>
    </row>
    ....
    <row>
        <ISPT_TOP>32</ISPT_TOP>
        <ISPT_REP>21</ISPT_REP>
        <ISPT_TYPE>S</ISPT_TYPE>
    </row>
</ISPT>
```

Figure 3.17 An instance of ISPT test within a XAGS 2.0 XML file



ISPT (2)

	= ISPT_ID	= HOLE_IDREF	() uom	() row
1	898SPT	898	uom () ISPT_TOP_UNIT ft	row (6)
2	740SPT	740	uom () ISPT_TOP_UNIT ft	row (9)

row (6)

	() ISPT_TOP	() ISPT_REP	() ISPT_TYPE
1	4.5	20	S
2	9.5	20	S
3	16.5	38	S
4	21.5	32	S
5	26.5	51	S
6	32	21	S

row (9)

	() ISPT_TOP	() ISPT_REP	() ISPT_TYPE
1	22	52	S
2	42	99	S
3	50.5	37	S
4	52.5	32	S
5	54.5	23	S
6	56.5	16	S
7	58.5	27	S
8	60.5	20	S
9	61.5	16	S

Figure 3.18 ISPT instances within a XAGS 2.0 XML instance file in Grid view

CPT Test

The Cone Penetration Test (CPT) measures the resistance of a cone that penetrates the ground. It also determines the lateral friction on a friction sleeve, and in the case of the piezzocone, the pore water pressure around the cone. CPT standards can be found in BS and ASTM (2005a). As XAGS complies with AGS data dictionary CPT test is defined by the four letter acronym of STCN. Figure 3.19 presents schema diagram of STCN complex type in XAGS2.0. Figure 3.20 shows an instance of STCN element in a XAGS2.0 XML file. Figure 3.21 demonstrates the same instance in a grid view.

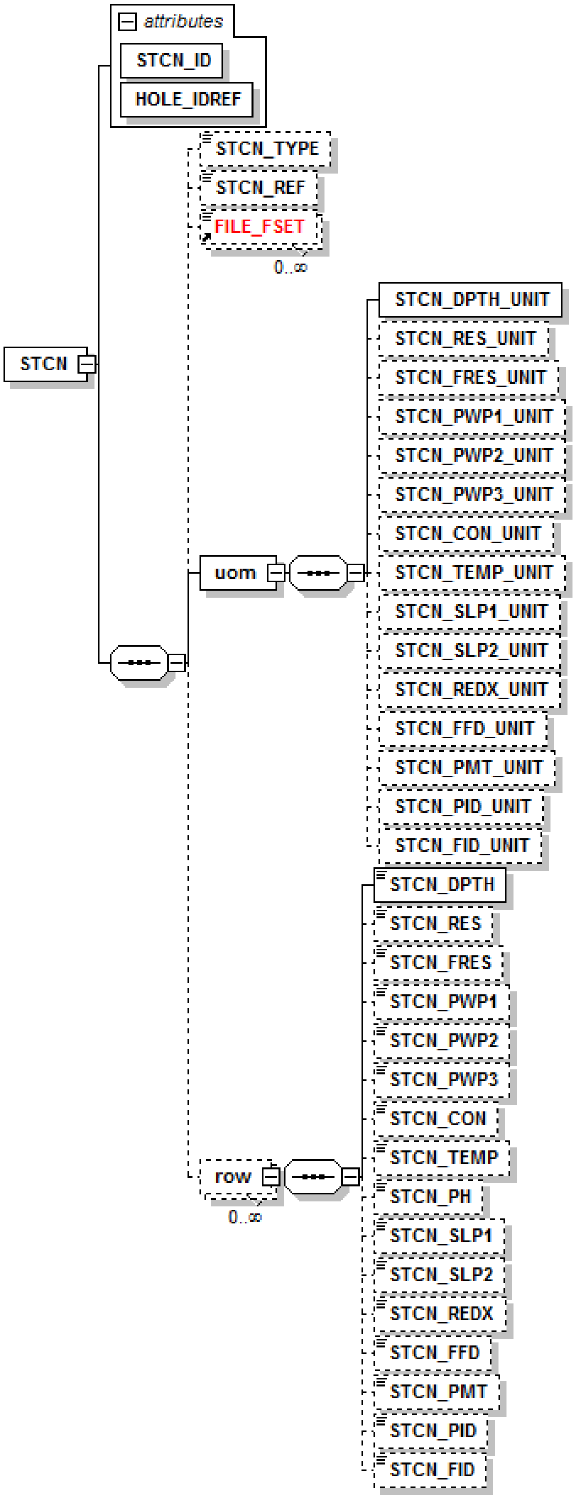
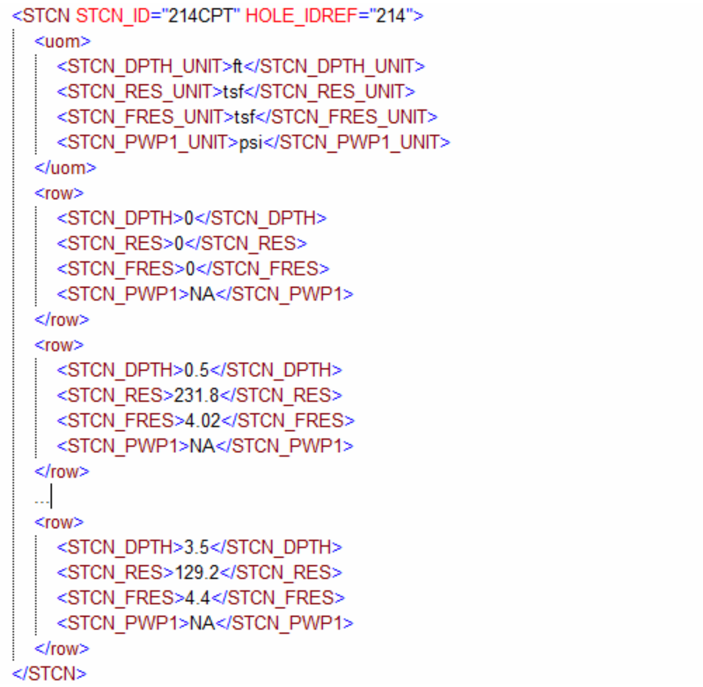
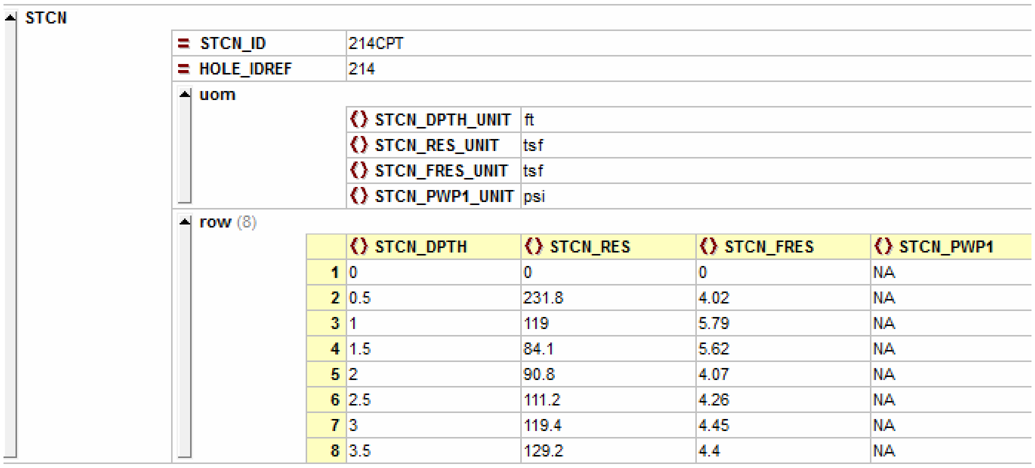


Figure 3.19 Schema diagram of STCN complex type in XAGS 2.0



```
<STCN STCN_ID="214CPT" HOLE_IDREF="214">
  <uom>
    <STCN_DPTH_UNIT>ft</STCN_DPTH_UNIT>
    <STCN_RES_UNIT>tsf</STCN_RES_UNIT>
    <STCN_FRES_UNIT>tsf</STCN_FRES_UNIT>
    <STCN_PWP1_UNIT>psi</STCN_PWP1_UNIT>
  </uom>
  <row>
    <STCN_DPTH>0</STCN_DPTH>
    <STCN_RES>0</STCN_RES>
    <STCN_FRES>0</STCN_FRES>
    <STCN_PWP1>NA</STCN_PWP1>
  </row>
  <row>
    <STCN_DPTH>0.5</STCN_DPTH>
    <STCN_RES>231.8</STCN_RES>
    <STCN_FRES>4.02</STCN_FRES>
    <STCN_PWP1>NA</STCN_PWP1>
  </row>
  ...
  <row>
    <STCN_DPTH>3.5</STCN_DPTH>
    <STCN_RES>129.2</STCN_RES>
    <STCN_FRES>4.4</STCN_FRES>
    <STCN_PWP1>NA</STCN_PWP1>
  </row>
</STCN>
```

Figure 3.20 An instance of STCN test within a XAGS 2.0 XML file



STCN

= STCN_ID	214CPT
= HOLE_IDREF	214

uom

() STCN_DPTH_UNIT	ft
() STCN_RES_UNIT	tsf
() STCN_FRES_UNIT	tsf
() STCN_PWP1_UNIT	psi

row (8)

	() STCN_DPTH	() STCN_RES	() STCN_FRES	() STCN_PWP1
1	0	0	0	NA
2	0.5	231.8	4.02	NA
3	1	119	5.79	NA
4	1.5	84.1	5.62	NA
5	2	90.8	4.07	NA
6	2.5	111.2	4.26	NA
7	3	119.4	4.45	NA
8	3.5	129.2	4.4	NA

Figure 3.21 STCN instance within a XAGS 2.0 XML instance file in Grid view

3.4.6. Sample

Sample (SAMP) is a segment from a hole, which is located using the depth relative to the top of borehole. Sample refers to its direct parent HOLE with HOLE_IDREF. Schema diagram of SAMP complex type for XAGS 2.0 can be seen in Figure 3.22.

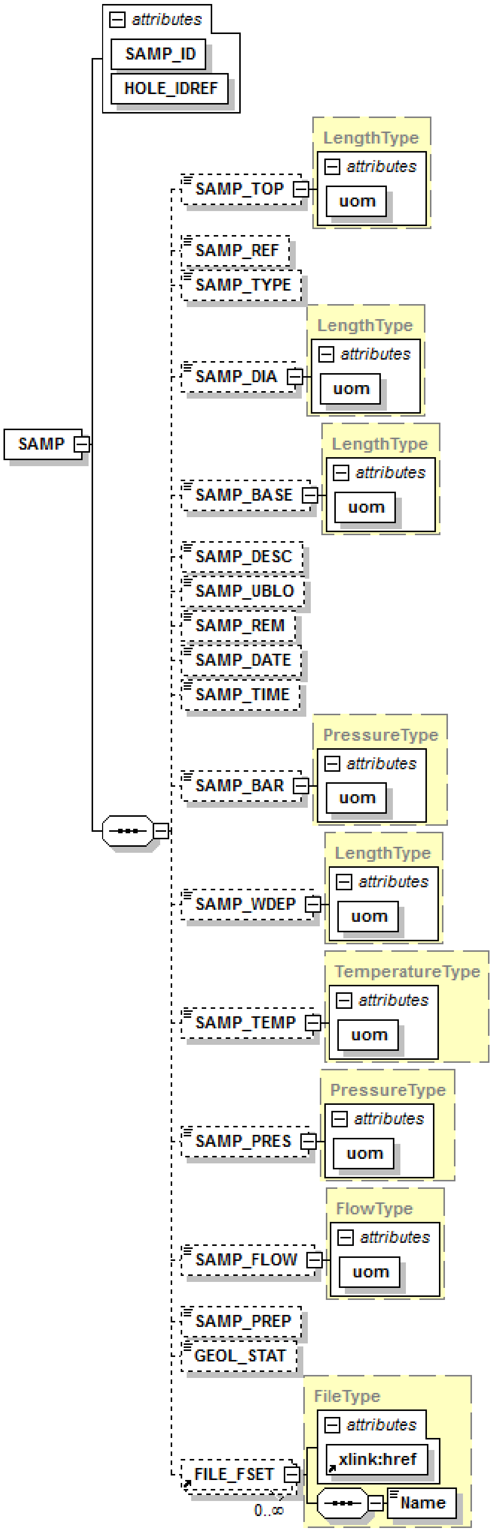


Figure 3.22 Schema diagram of SAMP complex type in XAGS 2.0

3.4.7. Laboratory Tests

In most geotechnical projects samples are collected from different depths within boreholes during site investigation. These samples are later used for laboratory testing. These laboratory tests may include soil classification, triaxial test, direct shear test, and consolidation test (Bardet, 1997). List of the laboratory test defined in XAGS 2.0 are presented in Table 3.1.

Shear Test

All laboratory tests are performed on specimens (SPEC) that are extracted from samples. As it was mentioned, sample is located using depth. Specimens are obtained and prepared form samples for the specific tests and are located within samples with an associate depth from top of the sample. All the laboratory tests listed in Table 3.1 are implemented in XAGS 2.0. As an example of lab tests, shear box test (SHBG and SHBT) schema diagrams are presented in Figure 3.23 and Figure 3.24. SHBG contain the summary information and the test results and SHBT carries the details of the shear test.

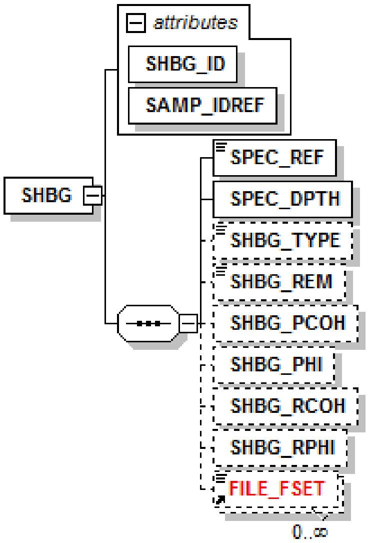


Figure 3.23 Schema diagram of SHBG complex type in XAGS2.0

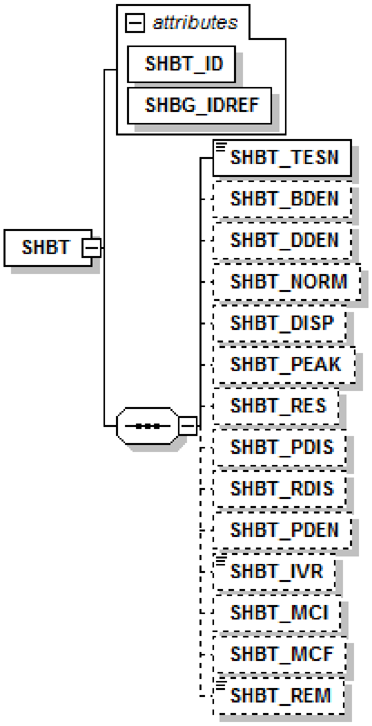


Figure 3.24 Schema diagram of SHBT complextype in XAGS2.0

All laboratory tests are following the same structure presented here for shear test. Laboratory tests refer to their direct parent SAMP with SAMP_IDREF. If the lab test is constructed of two levels, the second level, e.g. SHBT, refers to its corresponding parent lab test, e.g. in this case SHBG.

3.5. XAGS 2.0 Deliberations

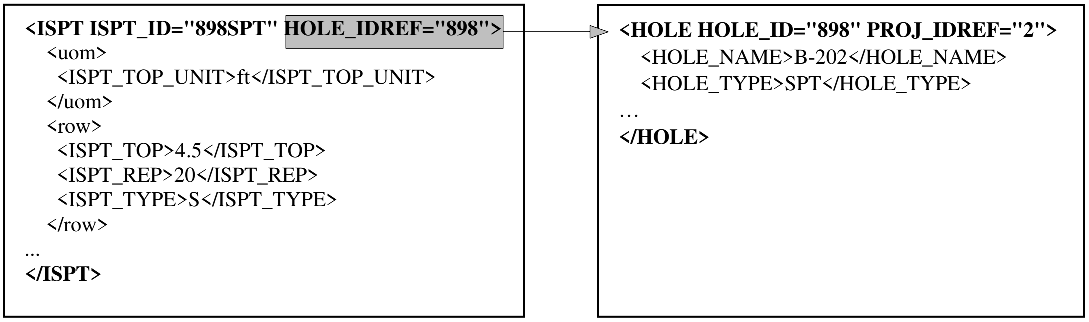
XML is a flexible format, meaning there are different ways to define an object with XML. Therefore, careful thought and additional studies are needed to make sure that each object is defined in the best way based on its usage and applications within geotechnical community. Some of these deliberations faced during XAGS development are explained in this section.

3.5.1. Unique Identifiers

XML Schema provides a mechanism for unique constraints. Keys and key references are used for creating relationships between elements through the value of an attribute or contained element. "key" and "keyref" elements each contain "selector" and "field" child elements (Vlist, 2002). Defining keys and keyrefs avoid redundant data in XML documents.

XAGS 2.0 generates XML documents in which elements are defined only once but may be referenced several times. For example, as highlighted in Figure 3.25, a HOLE named "B-202" is called several times in the XML document. This HOLE is assigned a unique HOLE_ID attribute value "898" (i.e., <HOLE HOLE_ID="898" PROJ_IDREF="2">).

HOLE_ID is the key defined in the schema. This HOLE is then called at several other places such as in SPT testing by using HOLE_IDREF (i.e., <ISPT ISPT_ID="898SPT" HOLE_IDREF="898">). HOLE_IDREF is defined by keyref in the schema.



```
<ISPT ISPT_ID="898SPT" HOLE_IDREF="898">
  <uom>
    <ISPT_TOP_UNIT>ft</ISPT_TOP_UNIT>
  </uom>
  <row>
    <ISPT_TOP>4.5</ISPT_TOP>
    <ISPT_REP>20</ISPT_REP>
    <ISPT_TYPE>S</ISPT_TYPE>
  </row>
...
</ISPT>
```

```
<HOLE HOLE_ID="898" PROJ_IDREF="2">
  <HOLE_NAME>B-202</HOLE_NAME>
  <HOLE_TYPE>SPT</HOLE_TYPE>
…
</HOLE>
```

Figure 3.25 HOLE_IDREF points to the object identified with HOLE_ID for unique identifying and building unique relationship between SPT test and borehole

3.5.2. File Element

FILE_FSET is the element used for defining the attached files for each entity in XAGS 2.0. Figure 3.26 shows the structure of FILE_FSET element defined in XAGS2.xsd. A complex type named FileType is defined for FILE_FSET. FileType has a string element, Name, and a required attribute that defines the link to where the file resides through xlink:href. Figure 3.27 demonstrates the FileType complextype definition for FILE_FSET in the schema.

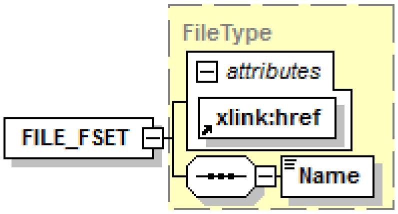
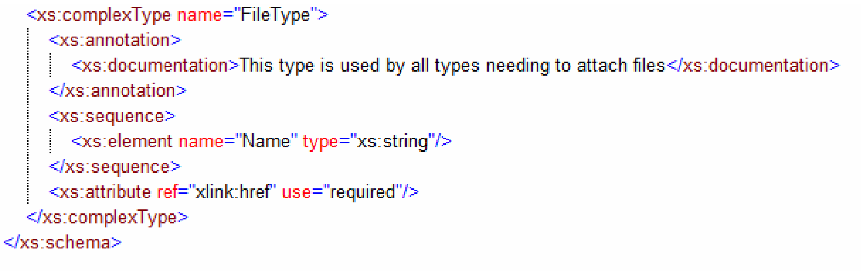


Figure 3.26 Schema diagram of FILE-FSET XAGS2.0. File Type is a modular complex type defined and called by FILE_FSET, including file name and link to its location



```
  <xs:complexType name="FileType">
    <xs:annotation>
      <xs:documentation>This type is used by all types needing to attach files</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
    </xs:sequence>
    <xs:attribute ref="xlink:href" use="required"/>
  </xs:complexType>
</xs:schema>
```

Figure 3.27 File Type complex type definition in XAGS 2.0 schema, corresponding to Figure 3.26

3.5.3. Coordinate System

Location of each borehole in AGS is being defined by northing and easting in HOLE data group. The terms easting and northing are geographic Cartesian coordinates for a point. Easting refers to the eastward-measured distance (or the x-coordinate), while northing refers to the northward measured distance (or the y-coordinate). The orthogonal coordinate pair is commonly measured in meters from a horizontal datum. This simple cartographic convention comes from a methodology for determining coordinates and areas, known as the method of latitudes and departures. Eastings are the coordinates that stretch along the bottom ("x") axis on the map, and northings stretch along the orthogonal side ("y") axis.

Currently, geographic coordinates, latitude and longitude, are used for XAGS version 2.0., instead of northing and easting as defined in AGS. World Geodetic System of 1984 (WGS84) datum is the default datum. WGS84 (dating from 1984 and last revised in 2004; Bolstad, 2005) is one of the most commonly used datums and is the reference coordinate system used by the Global Positioning System (GPS), Google Earth and Google Maps. Selection of geographic coordinates for XAGS is based on two reasons. Usage of one coordinate system and datum will significantly simplify XAGS format, and will eliminate conversion and interpretation problems. Additionally, the data will be easily located on interactive maps such as Google Earth and Google Maps as demonstrated in the next chapter with no additional processing.

In case original data is presented in other coordinate systems, conversion of coordinate system will be necessary before inputting data into XAGS format. However, it is possible to consider adding other coordinate system presentations to XAGS in future.

3.5.4. Data Structure Options for Serialized Data

For presenting serialized data, such as time histories, or CPT and SPT test results; there is no clear solution that solves all validation and consistency issues, while providing the desired flexibility and minimal file size. Where tabular values are being encoded, based on what the intended usage is, data specific objects can be defined.

There is extensive research available on presenting earthquake events in XML format. QuakeML (2010), Pacific Earthquake Engineering Research (PEER), Southern California Earthquake Center (SCEC) and US Geological Survey (USGS) all use XML presentation to archive the earthquake events. However, the recent attempts to use XML representation for actual earthquake records, still keep the data as a large string within the schema file (Tsuboi and Morino, 2003; XML-SEED). Representing the actual time series of physical quantities for earthquake records, such as waveforms, in XML format is not tackled by the seismic community yet. Within the geotechnical community, GML-conformant spatial modeling presented by Bardet and Zand (2009), reviewed in chapter 2, uses MultiPoint and utilizes coverage features of GML for presenting serilalized data such as CPT testing. However, this feature is only available for GML compliant files and as discussed in chapter 2, GML-compliant schemas are far more complicated and complex to be used and accepted by the geotechnical community at least at the current time. XML presentation for serialized values and time histories in the other fields such as general measuring and observation standards (OGC; Cox, 2007), sensors measurements (Botts et. al., 2008), hydraulic information systems (CUAHSI HIS, 2010), and even waveform representation in medical informatics (Wang et. al, 2004; Zheng et. al., 2010) have also been studied.

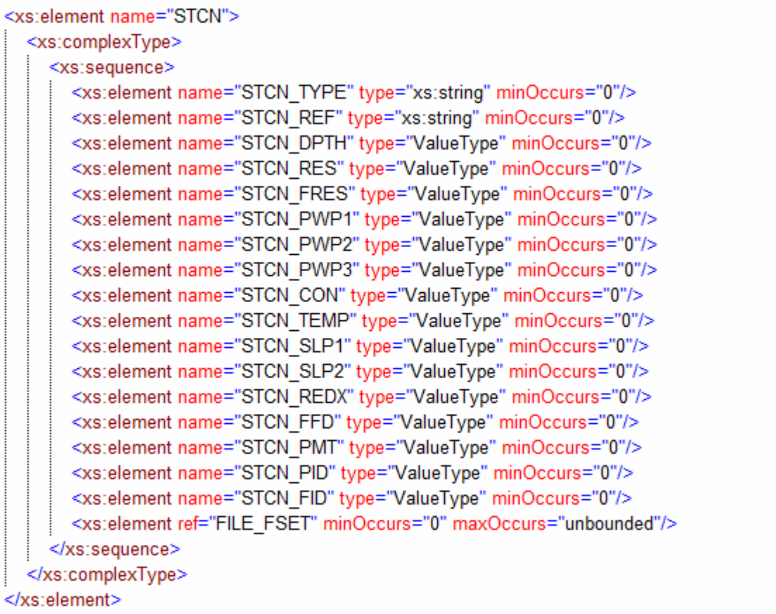
These studies show that generally, with exception of using GML, there are three possible data structures for serialized data in XML presentation:

Each Value One Tag (Regular Tabular Representation):

The automatic method of presenting tabular data in XML is to define each value within separate tags. The main benefit of this format is being able to specifically validate each single data in the serialized data set similar to the rest of XML data. However, in case of very large data sets, this format might result in large file sizes when XML element tags are repeated in very large numbers.

Definition of Columns:

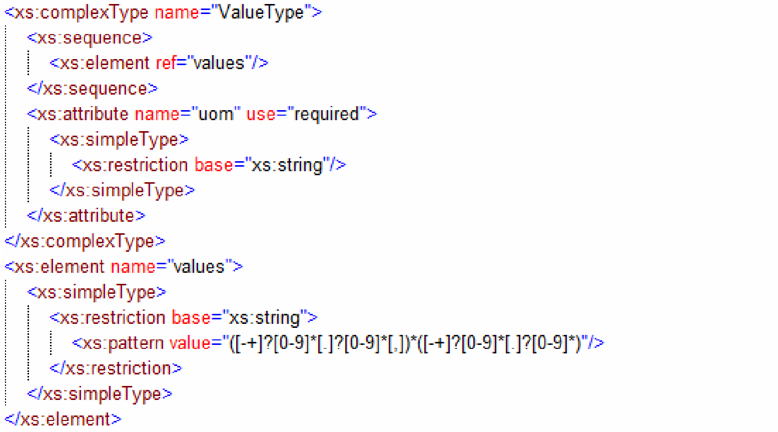
One of the ways to present a table in XML is to explicitly define each column expected in table. The Static Cone Test schema presented in Figure 3.28 is a sample of such a structure. Each column is of type ValueType. This is an abstract object that is a placeholder for a datatype-specific array that is defined in Figure 3.29.



```
<xs:element name="STCN">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="STCN_TYPE" type="xs:string" minOccurs="0"/>
      <xs:element name="STCN_REF" type="xs:string" minOccurs="0"/>
      <xs:element name="STCN_DPTH" type="ValueType" minOccurs="0"/>
      <xs:element name="STCN_RES" type="ValueType" minOccurs="0"/>
      <xs:element name="STCN_FRES" type="ValueType" minOccurs="0"/>
      <xs:element name="STCN_PWP1" type="ValueType" minOccurs="0"/>
      <xs:element name="STCN_PWP2" type="ValueType" minOccurs="0"/>
      <xs:element name="STCN_PWP3" type="ValueType" minOccurs="0"/>
      <xs:element name="STCN_CON" type="ValueType" minOccurs="0"/>
      <xs:element name="STCN_TEMP" type="ValueType" minOccurs="0"/>
      <xs:element name="STCN_SLP1" type="ValueType" minOccurs="0"/>
      <xs:element name="STCN_SLP2" type="ValueType" minOccurs="0"/>
      <xs:element name="STCN_REDX" type="ValueType" minOccurs="0"/>
      <xs:element name="STCN_FFD" type="ValueType" minOccurs="0"/>
      <xs:element name="STCN_PMT" type="ValueType" minOccurs="0"/>
      <xs:element name="STCN_PID" type="ValueType" minOccurs="0"/>
      <xs:element name="STCN_FID" type="ValueType" minOccurs="0"/>
      <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Figure 3.28 Standard Penetration Test (STCN) schema with standard column definition as ValueType (defined in Figure 3.29)

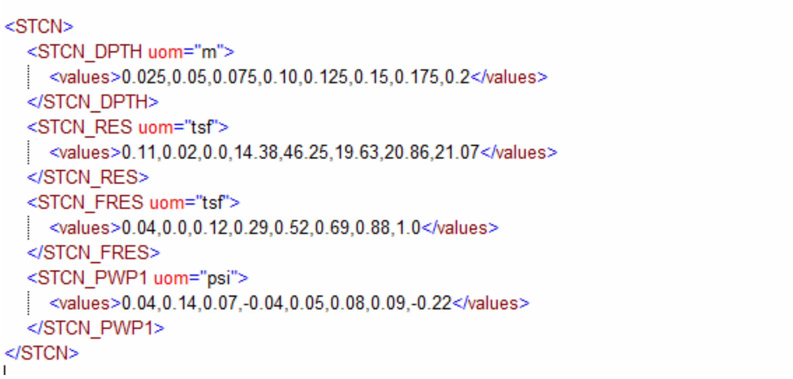
The row values for each column are validated by a regular expression, shown in Figure 3.29. In this case, the ValueType specialization contains an expression that forces the column values to be of float type.



```xml
<xs:complexType name="ValueType">
    <xs:sequence>
        <xs:element ref="values"/>
    </xs:sequence>
    <xs:attribute name="uom" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:string"/>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:element name="values">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:pattern value="([-+]?[0-9]*[.]?[0-9]*[,])*([-+]?[0-9]*[.]?[0-9]*)"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

Figure 3.29 ValueType definition for Standard Penetration Test (STCN) schema with standard column in Figure 3.28

Figure 3.30 shows a sample XML instance of this StaticConeTest object that is validated by the proposed schema.



```xml
<STCN>
    <STCN_DPTH uom="m">
        <values>0.025,0.05,0.075,0.10,0.125,0.15,0.175,0.2</values>
    </STCN_DPTH>
    <STCN_RES uom="tsf">
        <values>0.11,0.02,0.0,14.38,46.25,19.63,20.86,21.07</values>
    </STCN_RES>
    <STCN_FRES uom="tsf">
        <values>0.04,0.0,0.12,0.29,0.52,0.69,0.88,1.0</values>
    </STCN_FRES>
    <STCN_PWP1 uom="psi">
        <values>0.04,0.14,0.07,-0.04,0.05,0.08,0.09,-0.22</values>
    </STCN_PWP1>
</STCN>
```

Figure 3.30 Standard Penetration Test (STCN) instance XML file for STCN structure defined with standard columns validated by schema in Figure 3.29

The main benefits of this format are compactness and flexibility. The format will not result in large file sizes when the dataset is very large. The drawback of this approach is that XML schema validation cannot force each column to have the same number of rows, as would be expected in a table. This is the trade-off that allows the table data to be represented in a more compact format.

Definition of Rows:

Another way to present a table in XML is to explicitly define each row expected in the table. The benefits and drawback of this method are very similar to definition of column method. Main benefits of this format are compactness and flexibility. Each row type will be defined as an abstract object that is a placeholder for a datatype-specific array. The row values for each row are validated by a regular expression, similar to the one shown in Figure 3.29. In this case, the ValueType specialization contains an expression that forces the row values to be float type. This method will ensure that for each corresponding depth, a single measurement exists and it is being carried between the same tags. The drawback of this approach is that later manipulation of data through XSL and other codes will be more complicated as in each row values of different columns should be processed and get separated for further manipulations.

The solutions presented in (b) and (c) provide structures that can be only partially validated without requiring excessive file sizes. Solution (a) is the regular method of preesenting tabular data in XML. The resolution to the issue of when to use a tabular presentation as opposed to a non-tabular presentation is highly dependent on the expected amount of data, as well as the flexibility required for a particular construct.

As for XAGS 2.0, regular method of presenting tabular data in XML (a) is used. Tables are represented by complex elements. The complex elements have the same name as the table in AGS data dictionary. The columns are represented by sub-elements and attributes with the names identical to column names. The requirements of a proper data exchange format include the ability to automatically validate the data types, their value range and distinguishing the missing values. Niether solution (b) nor (c) can satisfy these requirements. Additionally, the largest serialized data sets in geotechnical projects are usually the CPT test results that are still a finite data set, and even with having separate tags for each value the file sizes will be acceptable. Figure 3.19 demonstrates the regular option of defining one tag for each value for CPT tabular data in XAGS 2.0 and Figure 3.20 shows an instance file of XAGS 2.0 CPT results conforming to the schema in Figure 3.19.

3.6. Summary

AGS satisfies some of the data exchange format requirements; however its file format (ASCII) lacks many of the needed capabilities of a versatile exchange format. On the other hand, Extensible Markup Languages (XML family) have become more and more popular in the data exchange world and now dominate data exchange. Many commonly used programs are already incorporating and processing XML files to some extent.

Existing conversion algorithms for transforming relational data structure to XML-based data organization were studied and transformation principles were defined. First, the automatic transformation was explored and modified by the expert domain knowledge to develop XAGS 1.0 that is based on the nested structure and hierarchical characteristics of in-situ and laboratory tests in geotechnical projects. Even though this format is well structured, it will be hard to integrate subsets of data back to the complete data set. Hence, a second version of XAGS was proposed. In XAGS 2.0, by using XPath, relationships are preserved through key and key ref attributes.

XAGS 2.0 schema implementation was reviewed in detail with the help of schema diagrams and XML instances. Scheme symbols used for the schema diagrams were first explained. Then the global structure and main elements present in XAGS 2.0 were defined. Data types used within XAGS were listed and the differences between simpletype and complextype were addressed. Afterward, XAGS 2.0 project, borehole, SPT and CPT tests, samples and laboratory tests elements are more elaborately explained with schema diagrams and XML instance files. Decisions about defining unique identifiers, file attachments, and representations of serialized data were discussed. Different ways to present serialized data were discussed and after careful attention and consideration of advantages and shortcomings of different possible ways the regular way of tabular presentation was selected as the better selection for in situ tests. Additionally, coordinate systems of AGS and XAGS and the reason for their differences are addressed. Complete files of XAGS 2.0 schema can be found in Appendix A.

In the next chapter XAGS will be assessed and validated by demonstration of its ease of usage, display, structural consistency checking, exchange and dissemination.

Chapter 4. XAGS 2.0 Data Usage, Validation, Exchange and Distribution

Representation of geotechnical data in XML (XAGS, which was defined in chapter 3), can be easily generated, modified, displayed, validated, exchanged, converted and distributed with the help of common software programs that have XML handling capabilities. XML editors and spreadsheets are discussed as two simple methods for XAGS data generation and modification. Afterward, a few XAGS data integration examples with XSL (Extensible Stylesheet Language) Transformation are demonstrated for spatial and non-spatial display of geotechnical boreholes over the internet or on local machines. Then data validation capabilities of XAGS schema are examined by some examples. XSL can also be used to convert XAGS data to AGS or other formats. It is also possible to convert other formats to XAGS data for transformation of legacy data to the new format, as demonstrated for a sample case history data set Additionally, this chapter is going to apply XAGS data exchange to (1) exchange data sets and data subsets between different teams and team members, and (2) to distribute the sample dataset via World Wide Web. The distribution over Internet with the use of recent information technologies enables (1) visualization of data sets and (2) accessibility of the files over the Internet

This discussion demonstrates the simplicity and effectiveness of XAGS in actual geotechnical projects. Programs used for XAGS manipulation are mainly free, common and simple-to-use such as internet explorer and XML Notepad. This availability and simplicity of software programs used will encourage adoptability and acceptance. At the end of this chapter XAGS will be evaluated against the requirement list developed in chapter two.

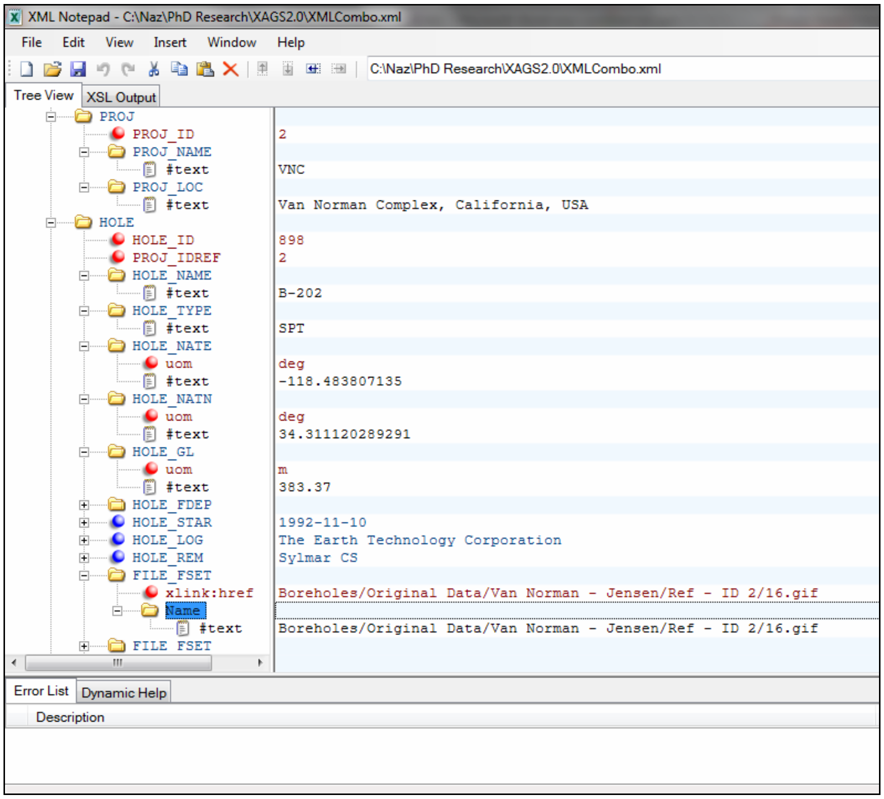
4.1. Data Generation and Modification

4.1.1. XML editors

XML documents can be created using any standard text editor. However, it is more convenient to create and edit XML documents using XML editors. An XML editor verifies the XML source based on an XML Schema or XML DTD. Some XML editors provide for the ability to run an XSLT transform, or series of transforms, over a document or even XSL debugging features.

Text XML editors generally provide features of element word completion and automatic appending of a closing tag whenever an opening tag is entered. Coloring element text differently from regular text, known as syntax highlighting is a basic standard for XML editors. These features can help to prevent typographical errors in the XML code. Displaying line numbers is another common and useful feature. The advantage of text editors is that they present exactly the information that is stored in the XML file. It is the best way to control the formatting of the file (such as indentations), to do low-level operations (such as a “find” or “replace” on element names) and to edit XML files without any schema or configuration file. Graphical editors based on GUIs (Graphical User Interface) on the other hand may be easier to use than text editors, and may not require knowledge of XML syntax. They use DTDs or XML schemas and/or configuration files to map XML elements to graphical components.

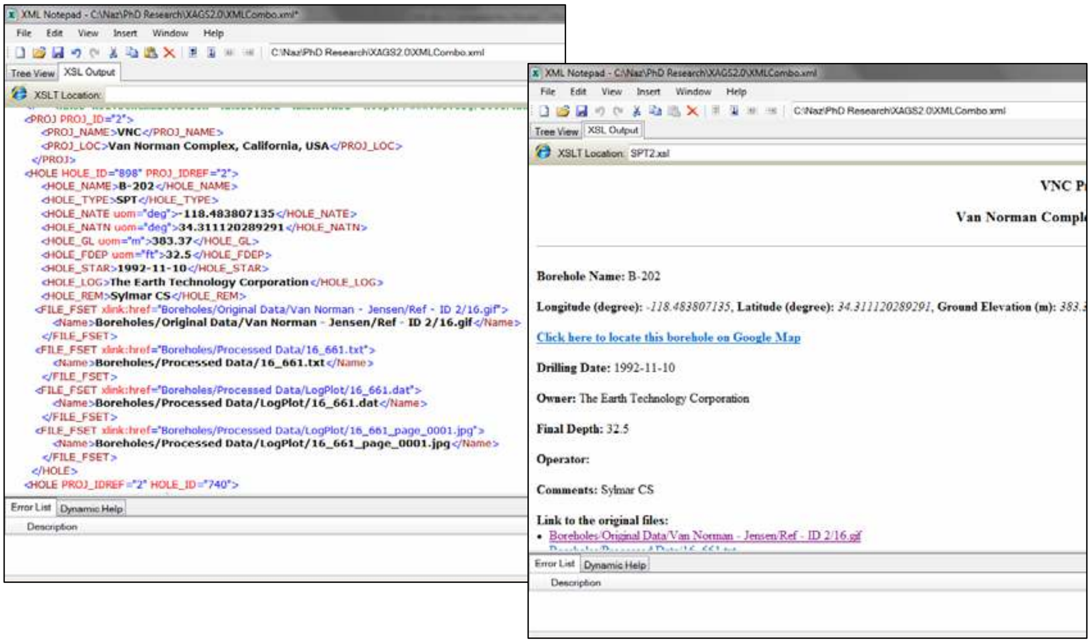
XML Notepad (2007) is a very simple and light XML editor developed by Microsoft which is available free to public. XML Notepad is only 1.9MB in size. XML Notepad offers an intuitive and simple user interface that graphically represents the tree structure of XML data. XML Notepad interface is a simple vertically split window of tree view at the left and the texts of values at the right, e.g., Figure 4.1.


XML Notepad - C:\Naz\PhD Research\XAGS2.0\XMLCombo.xml
File
Edit
View
Insert
Window
Help
C:\Naz\PhD Research\XAGS2.0\XMLCombo.xml
Tree View
XSL Output
Error List
Dynamic Help
Description

```
PROJ
    PROJ_ID             2
    PROJ_NAME
        #text           VNC
    PROJ_LOC
        #text           Van Norman Complex, California, USA
HOLE
    HOLE_ID             898
    PROJ_IDREF          2
    HOLE_NAME
        #text           B-202
    HOLE_TYPE
        #text           SPT
    HOLE_NATE
        uom             deg
        #text           -118.483807135
    HOLE_NATN
        uom             deg
        #text           34.311120289291
    HOLE_GL
        uom             m
        #text           383.37
    HOLE_FDEP
    HOLE_STAR           1992-11-10
    HOLE_LOG            The Earth Technology Corporation
    HOLE_REM            Sylmar CS
    FILE_FSET
        xlink:href      Boreholes/Original Data/Van Norman - Jensen/Ref - ID 2/16.gif
        Name
            #text       Boreholes/Original Data/Van Norman - Jensen/Ref - ID 2/16.gif
    FILE_FSET
```

Figure 4.1 XML Notepad view of a XAGS file. Tree view of XML editor showing the tree structure of XAGS file in left window and color coded values in the right window

XML Notepad simplifies the editing process by presenting the XML tags separate from the values they contain. The XSL preview lets you view the final XML output. As shown in Figure 4.2 (left) the output is color coded. If an XSL file is assigned to the XML file, the XSL preview will display the final output of XML file after the formatting is done by the XSL file. Figure 4.2 (right) shows the XSL output of the XML file demonstrated in Figure 4.2 (left). XML Notepad validates your document while you are editing and shows any errors or warnings in the Error List panel at the bottom of the interface. You can double click on errors to navigate to the error to fix it. Figure 4.1 and Figure 4.2 have empty Error List panels. Since the schema is assigned to the XML file, this empty error panel means that the XML file is valid against the schema with no error.


XML Notepad - C:\Naz\PhD Research\XAGS2.0\XMLCombo.xml*
File
Edit
View
Insert
Window
Help
C:\Naz\PhD Research\XAGS2.0\XMLCombo.xml
Tree View
XSL Output
XSLT Location:
Error List
Dynamic Help
Description

```
<PROJ PROJ_ID="2">
   <PROJ_NAME>VNC</PROJ_NAME>
   <PROJ_LOC>Van Norman Complex, California, USA</PROJ_LOC>
  </PROJ>
<HOLE HOLE_ID="898" PROJ_IDREF="2">
   <HOLE_NAME>B-202</HOLE_NAME>
   <HOLE_TYPE>SPT</HOLE_TYPE>
   <HOLE_NATE uom="deg">-118.483807135</HOLE_NATE>
   <HOLE_NATN uom="deg">34.311120289291</HOLE_NATN>
   <HOLE_GL uom="m">383.37</HOLE_GL>
   <HOLE_FDEP uom="ft">32.5</HOLE_FDEP>
   <HOLE_STAR>1992-11-10</HOLE_STAR>
   <HOLE_LOG>The Earth Technology Corporation</HOLE_LOG>
   <HOLE_REM>Sylmar CS</HOLE_REM>
  <FILE_FSET xlink:href="Boreholes/Original Data/Van Norman - Jensen/Ref - ID 2/16.gif">
    <Name>Boreholes/Original Data/Van Norman - Jensen/Ref - ID 2/16.gif</Name>
   </FILE_FSET>
  <FILE_FSET xlink:href="Boreholes/Processed Data/16_661.txt">
    <Name>Boreholes/Processed Data/16_661.txt</Name>
   </FILE_FSET>
  <FILE_FSET xlink:href="Boreholes/Processed Data/LogPlot/16_661.dat">
    <Name>Boreholes/Processed Data/LogPlot/16_661.dat</Name>
   </FILE_FSET>
  <FILE_FSET xlink:href="Boreholes/Processed Data/LogPlot/16_661_page_0001.jpg">
    <Name>Boreholes/Processed Data/LogPlot/16_661_page_0001.jpg</Name>
   </FILE_FSET>
  </HOLE>
<HOLE PROJ_IDREF="2" HOLE_ID="740">
```

XML Notepad - C:\Naz\PhD Research\XAGS2.0\XMLCombo.xml
File
Edit
View
Insert
Window
Help
C:\Naz\PhD Research\XAGS2.0\XMLCombo.xml
Tree View
XSL Output
Error List
Dynamic Help
Description

XSLT Location: SPT2.xsl

VNC P

Van Norman Compl

Borehole Name: B-202

Longitude (degree): *-118.483807135*, **Latitude (degree):** *34.311120289291*, **Ground Elevation (m):** *383.3*

Click here to locate this borehole on Google Map

Drilling Date: 1992-11-10

Owner: The Earth Technology Corporation

Final Depth: 32.5

Operator:

Comments: Sylmar CS

Link to the original files:

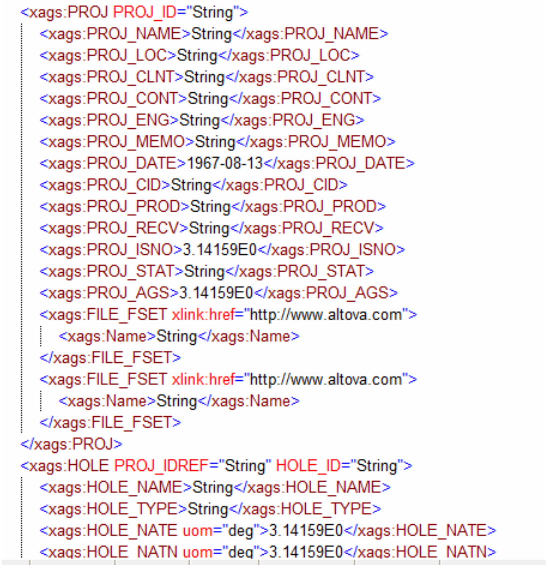
- Boreholes/Original Data/Van Norman - Jensen/Ref - ID 2/16.gif

Figure 4.2 XAGS output view showing plain text without presence of an XSL file (left) and XAGS data after transformation with SPT2.xsl (right)

Overall, XML Notepad does basic editing job of XAGS documents well. The tree view interface alone lets users view the structure of the document naturally. However, no table view of XML data is available in XML Notepad. Whatever XML editor used for XAGS data processing from commercial software programs to free XML editors results should be the same. The material in this research does not have any attachment to any specific XML editor and the users are able to use any other editor for working with XAGS documents.

4.1.2. Random Data Generation

As long as an XML editor that handles schema is in hand, a random XAGS document file can be generated. A sample of randomly generated XAGS data is partially displayed in Figure 4.3. This random data generated file can further be used to input the acquired information into a data file for avoiding structural mistakes.



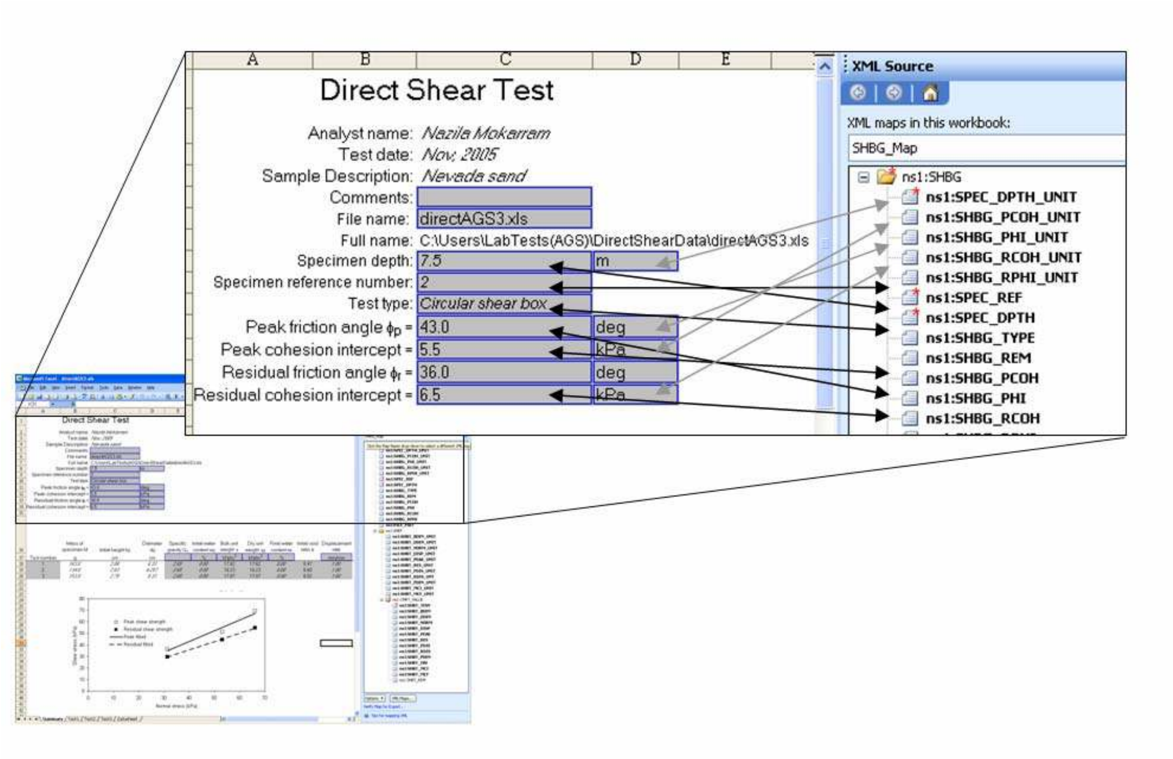
```
<xags:PROJ PROJ_ID="String">
  <xags:PROJ_NAME>String</xags:PROJ_NAME>
  <xags:PROJ_LOC>String</xags:PROJ_LOC>
  <xags:PROJ_CLNT>String</xags:PROJ_CLNT>
  <xags:PROJ_CONT>String</xags:PROJ_CONT>
  <xags:PROJ_ENG>String</xags:PROJ_ENG>
  <xags:PROJ_MEMO>String</xags:PROJ_MEMO>
  <xags:PROJ_DATE>1967-08-13</xags:PROJ_DATE>
  <xags:PROJ_CID>String</xags:PROJ_CID>
  <xags:PROJ_PROD>String</xags:PROJ_PROD>
  <xags:PROJ_RECV>String</xags:PROJ_RECV>
  <xags:PROJ_ISNO>3.14159E0</xags:PROJ_ISNO>
  <xags:PROJ_STAT>String</xags:PROJ_STAT>
  <xags:PROJ_AGS>3.14159E0</xags:PROJ_AGS>
  <xags:FILE_FSET xlink:href="http://www.altova.com">
    <xags:Name>String</xags:Name>
  </xags:FILE_FSET>
  <xags:FILE_FSET xlink:href="http://www.altova.com">
    <xags:Name>String</xags:Name>
  </xags:FILE_FSET>
</xags:PROJ>
<xags:HOLE PROJ_IDREF="String" HOLE_ID="String">
  <xags:HOLE_NAME>String</xags:HOLE_NAME>
  <xags:HOLE_TYPE>String</xags:HOLE_TYPE>
  <xags:HOLE_NATE uom="deg">3.14159E0</xags:HOLE_NATE>
  <xags:HOLE_NATN uom="deg">3.14159E0</xags:HOLE_NATN>
```

Figure 4.3   A sample random data generation from XAGS schema for further data input

4.1.3. Spreadsheets for Data Generation and Modification

Spreadsheet packages offer a user-friendly interface for data presentation. They are being used intensively by geotechnical community to report the geotechnical tests data. Templates are developed in every community according to their specific needs (Bardet, 1997). One of the powerful features of Microsoft Office Excel is support of user-defined XML schemas. XML schemas and XML data can be added to the workbooks. The features include: view the data in an XML list; view the data in a workbook; and create a schema map from the structure and import data into the map. Importing data into the map populates the cells with data. This is a very easy and accessible option for working with XML data. New data can be imported into a map or a system can be set up to update your worksheets automatically.

Figure 4.4 shows a worksheet with a mapped schema. Excel surrounds the mapped cells with a blue border. Excel is able to map from the same schema to different worksheets, if the structure can be preserved. XML files can be both imported and exported in Excel. Figure 3.10 is a part of the XML file exported from the Excel spreadsheet- XAGS mapping of Shear Test. However, Excel mapping features are not very flexible, and have some limitations. For instance, multi-cell array formulas are not allowed in XML mappings; a list of lists is not exportable because elements' relationship can not be preserved; and tabular (serialized) data should be defined in adjacent cells from left to right, otherwise excel is not able to construct the relationship between the two series.



Direct Shear Test

Analyst name: *Nazila Mokarram*

Test date: *Nov, 2005*

Sample Description: *Nevada sand*

Comments:

File name: directAGS3.xls

Full name: C:\Users\LabTests(AGS)\DirectShearData\directAGS3.xls

Specimen depth: 7.5 m

Specimen reference number: 2

Test type: *Circular shear box*

Peak friction angle ϕ_p = 43.0 deg

Peak cohesion intercept = 5.5 kPa

Residual friction angle ϕ_r = 36.0 deg

Residual cohesion intercept = 6.5 kPa

XML maps in this workbook: SHBG_Map

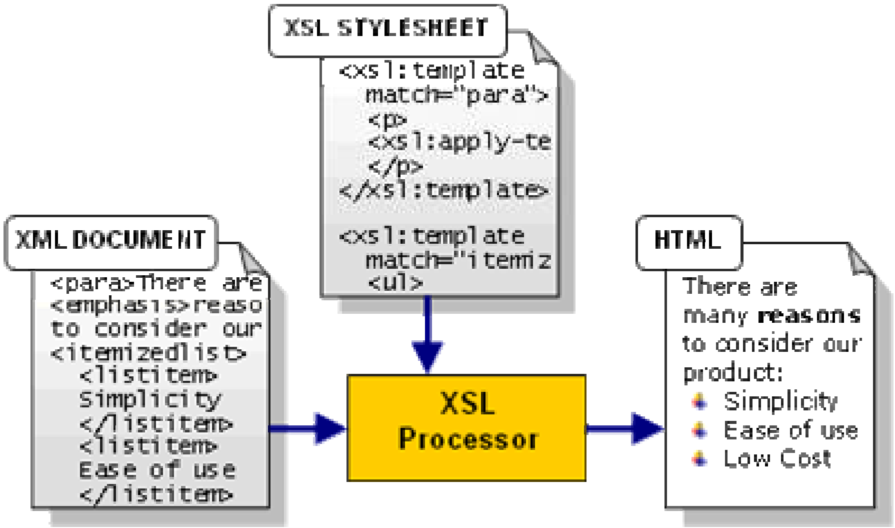
Figure 4.4 Excel spreadsheet and XAGS schema two-way mapping for data input and modification

4.2. Display of Borehole Data

Stylesheets are used to convert XAGS data into documents for display as web pages with links to locate the data on interactive maps. The transformation happens dynamically on data user's computer as the user opens the XAGS file for viewing in the internet browser. A sample XSL code used for creating the non-spatial and spatial presentation of SPT and CPT boreholes discussed in this section is presented in Appendix B.

4.2.1. Extensible Style sheet Language Family (XSL)

XML distinctively separates raw data from ancillary data used to display format. The term Extensible Style sheet Language (XSL) is used to refer to a family of languages used to transform and render XML documents into other formats (XSL, 2009). Many XML editors and tools are readily available to generate, edit and transform XML documents, e.g., Microsoft XML Notepad (2007) and Altova’s XMLSpy (2010). As shown in Figure 4.5, an XSL file is itself an XML file, and adheres to the same rules as a well-formed XML document does (Yank 2001). An XSL file is a mixture of tags and commands that format XML files for presentation. The line <?xml-stylesheet type="text/xsl" href="SPT2.xsl"?> at the top of the XML file calls the XSL file. The second line uses a namespace (e.g., xmlns:), which is used to avoid conflict between different elements when merging several XSL files. All the XSL elements are enclosed between <xls: stylesheet> and </xls: stylesheet> tags. The “/” in match="/" is shorthand to represent the XML document’s root element.



```
<xsl:template
  match="para">
  <p>
  <xsl:apply-te
  </p>
</xsl:template>

<xsl:template
  match="itemiz
  <ul>
```

```
<para>There are
<emphasis>reaso
to consider our
<itemizedlist>
  <listitem>
  Simplicity
  </listitem>
  <listitem>
  Ease of use
  </listitem>
```

There are many **reasons** to consider our product:
- Simplicity
- Ease of use
- Low Cost

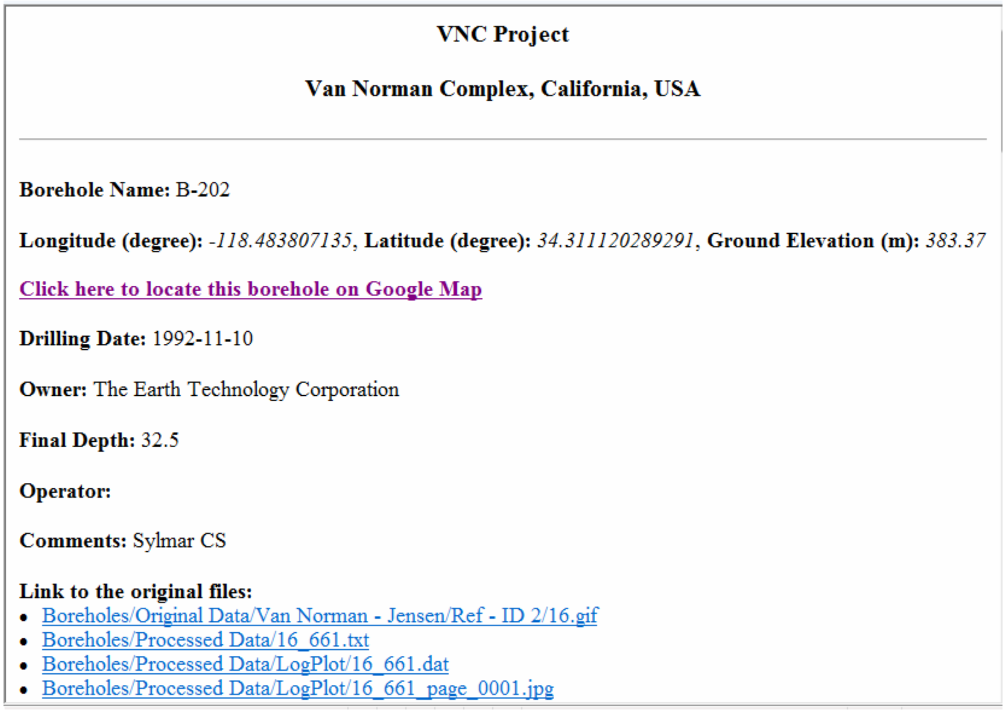
Figure 4.5 Basic elements and process flow of XSLT for transforming an XML document to other forms, e.g. HTML (Yank, 2001)

XSL specification eventually divided into three parts (XSL, 2009): XSL Transformations (XSLT), XSL Formatting Objects (XSL-FO), and XML Path Language (XPath). XSLT and XPath are used to present XAGS data. XSLT (Extensible Stylesheet Language Transformations) is a declarative, XML-based language used for the transformation of XML documents. The original document is not changed; rather, a new document is created based on the content of an existing one. The new document may be in standard XML syntax or in another format, such as Hyper Text Markup Language (HTML), Scalable Vector Graphics (SVG) or plain text. XPath, the XML Path Language, is a query language defined by W3C for selecting nodes from an XML document. The XPath language is based on a tree representation of the XML document, and provides the ability to navigate around the tree, selecting nodes by a variety of criteria.

4.2.2. Non-spatial Data Display with HTML

XAGS data can be sent to an XML editor processing XSL transformations or an internet browser that uses style sheet-extra information and be translated into other formats or be easily published on the web. Hyper Text Markup Language (HTML) is the language for publishing hypertext on the World Wide Web (HTML, 2010). It is a non-proprietary format based upon Standard Generalized Markup Language (ISO 8879, 1986), and can be created and processed by a wide range of tools, from simple plain text editors to sophisticated WYSIWYG (What You See Is What You Get) authoring tools. HTML uses tags such as <h1> and </h1> to structure text into headings, paragraphs, lists, hypertext links and other forms.

The sample XSL file presented in Appendix B is used to transform the XAGS data into an HTML file with links to interactive maps for borehole locations. The resulting borehole B-202 presentation is shown in Figure 4.6 with the aid of XSL in the Internet Explorer (IE) web browser.



VNC Project

Van Norman Complex, California, USA

Borehole Name: B-202

Longitude (degree): *-118.483807135*, **Latitude (degree):** *34.311120289291*, **Ground Elevation (m):** *383.37*

Click here to locate this borehole on Google Map

Drilling Date: 1992-11-10

Owner: The Earth Technology Corporation

Final Depth: 32.5

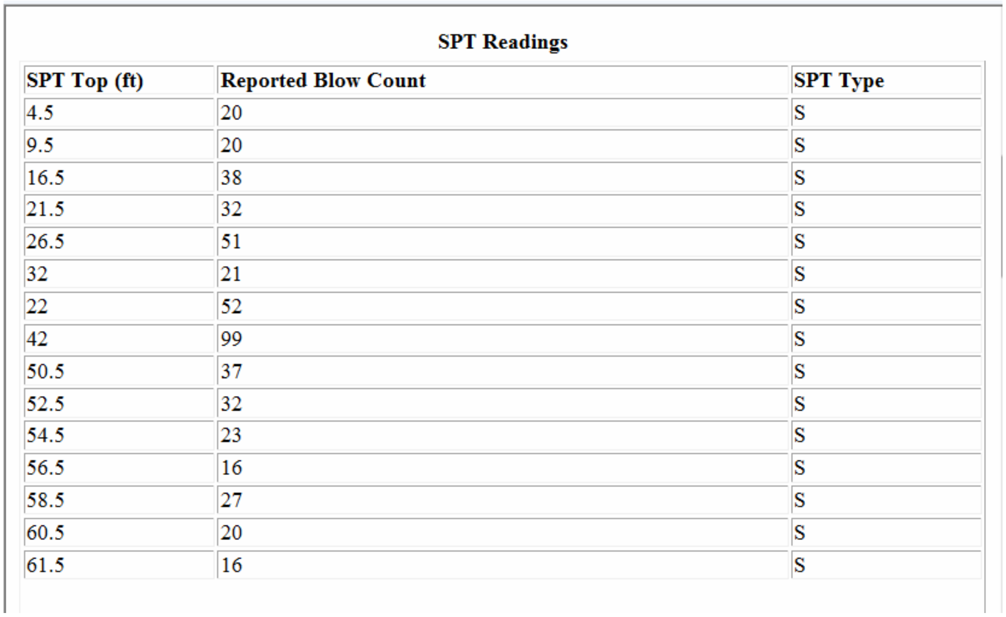
Operator:

Comments: Sylmar CS

Link to the original files:
- Boreholes/Original Data/Van Norman - Jensen/Ref - ID 2/16.gif
- Boreholes/Processed Data/16_661.txt
- Boreholes/Processed Data/LogPlot/16_661.dat
- Boreholes/Processed Data/LogPlot/16_661_page_0001.jpg

Figure 4.6 Partial presentation of borehole B-202 in IE web browser automatically generated by XSL transformation

XSL can also be used to tabulate data in tables. Figure 4.7 shows the SPT table presentation for the same borehole. Figure 4.8 displays a part of C-202 data including the CPT results table in a web browser and the map that shows its location. The XSL file is designed to dynamically distinguish what type of borehole is contained within the XAGS file and present the information within the corresponding representation design and tabulate the data versus corresponding depths in tables.



SPT Readings

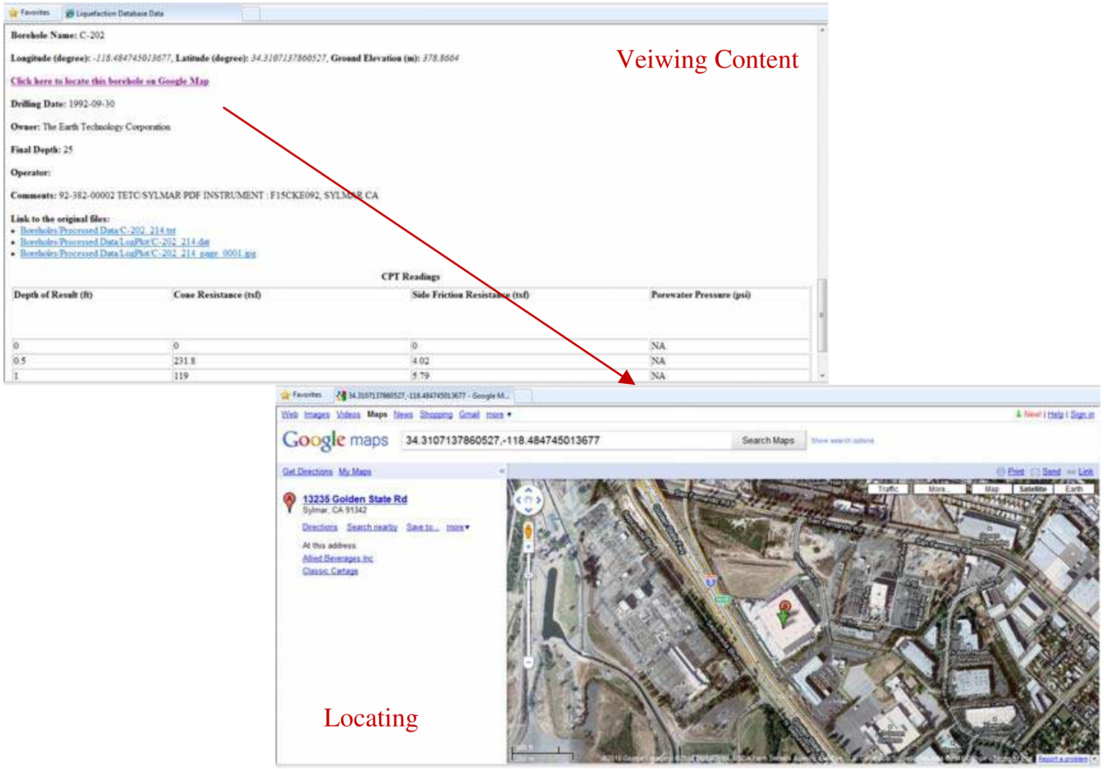
SPT Top (ft)	Reported Blow Count	SPT Type
4.5	20	S
9.5	20	S
16.5	38	S
21.5	32	S
26.5	51	S
32	21	S
22	52	S
42	99	S
50.5	37	S
52.5	32	S
54.5	23	S
56.5	16	S
58.5	27	S
60.5	20	S
61.5	16	S

Figure 4.7 Table presentation of B-202 SPT data in IE web browser automatically generated by XSL transformation

4.2.3. Spatial Data Display with Interactive Maps

Interactive maps such as Google Maps and Google Earth can be used for spatial display of data, if the spatial data exists within the data file. Google Maps is a web mapping service application and technology provided by Google, for non-commercial use that powers many map-based services and maps embedded on third-party websites via the Google Maps API (released in June 2005). It features a draggable and zoomable map that locates destinations and creates driving directions (Gibson and Erle, 2006). The map data used in Google Maps is provided by Tele Atlas and NAVTEQ (Schutzberg 2005). Google Earth is a virtual globe, map and geographic information program that maps the Earth by the superimposition of images obtained from satellite imagery, aerial photography and GIS 3Dglobe similar to Google Maps (Google Earth, 2010). Keyhole

Markup Language (KML) is an XML-based language schema for expressing geographic annotation and visualization developed specifically for Google Earth that can also be understood by Google Maps. Placemarks, ground overlays, paths, and polygons are some of the features of KML used for display of spatial data (KML, 2010).



Borehole Name: C-202

Longitude (degree): *-118.484745013677*, **Latitude (degree):** *34.3107137860527*, **Ground Elevation (m):** *378.8664*

Click here to locate this borehole on Google Map

Drilling Date: 1992-09-30

Owner: The Earth Technology Corporation

Final Depth: 25

Operator:

Comments: 92-382-00002 TETC/SYLMAR PDF INSTRUMENT : F15CKE092, SYLMAR CA

Link to the original files:
- Boreholes/Processed Data/C-202_214.txt
- Boreholes/Processed Data/LogPlot/C-202_214.dat
- Boreholes/Processed Data/LogPlot/C-202_214_page_0001.jpg

CPT Readings

Depth of Result (ft)	Cone Resistance (tsf)	Side Friction Resistance (tsf)	Porewater Pressure (psi)
0	0	0	NA
0.5	231.8	4.02	NA
1	119	5.79	NA

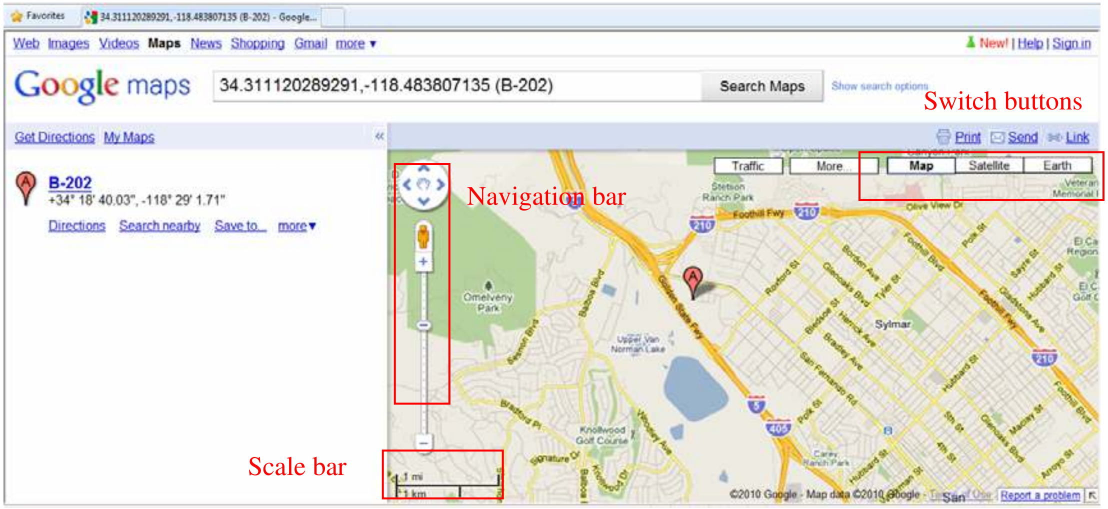
Favorites
34.3107137860527,-118.484745013677 - Google M...
Web Images Videos Maps News Shopping Gmail more ▾
New! | Help | Sign in
Google maps
Search Maps
Show search options
Get Directions My Maps
Print Send Link
Traffic More... Map Satellite Earth

34.3107137860527,-118.484745013677

13235 Golden State Rd
Sylmar, CA 91342
Directions Search nearby Save to... more▾
At this address:
Allied Beverages Inc
Classic Cartage

Figure 4.8 XAGS CPT borehole (C-202) data content viewing in HTML and locating in Google Maps with XSL processing

Figure 3.6 shows the location of B-202 on Google Maps. The XSL file presented in Appendix B creates the link to Google Maps with passing the corresponding coordinates to display the location of the borehole. The figure shows an interface of Google Maps that includes typical components, e.g., the navigation bar, mode buttons, and the scale bar. Google Maps has been used because of the simplicity of passing location parameters via a link from the main HTML page. Google Earth can be used for spatial representation of the boreholes as well. For this purpose, an XSL file is needed to construct a KML file containing the location of boreholes. This transformation has been performed for the case history data set example later in this chapter. At the current time KML has very limited capabilities and can not fully utilize all the XML-based data features for Google Earth display. For example, KML does not fully support XSL transformation. However if KML format becomes able to fully utilize all the characteristics of XML in future, it is recommended to study the possibility of having a KML-enabled version of XAGS. This will simplify the spatial presentation of data with use of Google Earth as well.



34.311120289291,-118.483807135 (B-202)

B-202
+34° 18' 40.03", -118° 29' 1.71"

Figure 4.9 Spatial presentation of B-202 in Google Maps automatically generated as a link by XSL transformation

Currently, from the existing web browsers only IE can process XSL to transform XML files. Only one output file can be presented with IE and only location parameters of one point can be passed to Google Maps at a time. It should be noted that commercial software will add more capacities and take data application to the next level. For example, with the use of a commercial XML editor such as XMLSpy it is possible to generate and save more than one output from one XML file with one XSL transformation. It is expected that some of the current limitations disappear in near future even for free programs, as they are increasingly adding to their capabilities.

4.3. Structural Consistency and Data Validation

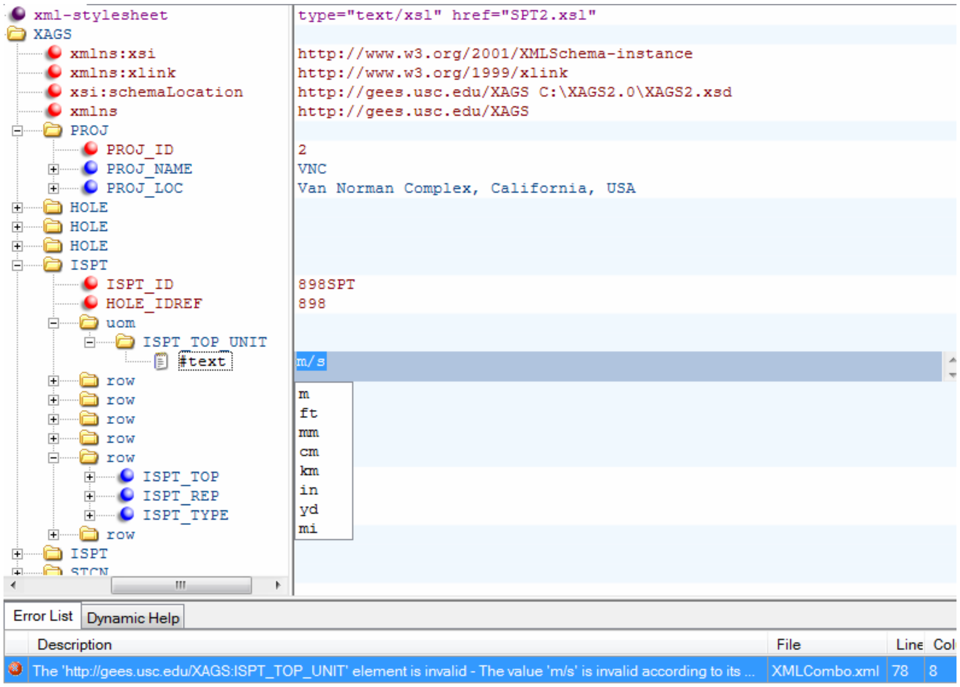
As seen in chapter two, AGS does not provide an automated way to check the data structure and integrity of a file. One of the most important contributions of introducing an XML-based format for geotechnical data is the possibility of data validation based on the schema definition file. Data validation is simple upon existence of an XML editor in hand. As long as correct addressing is provided at the beginning of data file, XAGS schema will be checked upon and non conforming points will be brought into view (e.g., in XML Notepad at the bottom of GUI) in the error list of the XML editor. XAGS schema is used to set up rich constraints for data in XAGS document and ensure internal consistency of the data. Some of the structural inconsistencies that can result in error messages are described here after:

4.3.1. Unit Enumeration

As described in chapter 3, pick lists are introduced for abbreviations of units in XAGS. This ensures the consistency and brevity of the data when incorporated in XML schema through enumeration data type.

Figure 4.10 shows an example of wrong units for ISPT_TOP_UNIT element that has provoked an "enumeration constraint failure" error message in the bottom window. The XML editor usually provides the acceptable units in a drop down menu to select from as shown in

Figure 4.10.



xml-stylesheet	type="text/xsl" href="SPT2.xsl"
XAGS	
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xmlns:xlink	http://www.w3.org/1999/xlink
xsi:schemaLocation	http://gees.usc.edu/XAGS C:\XAGS2.0\XAGS2.xsd
xmlns	http://gees.usc.edu/XAGS
PROJ	
PROJ_ID	2
PROJ_NAME	VNC
PROJ_LOC	Van Norman Complex, California, USA
HOLE	
HOLE	
HOLE	
ISPT	
ISPT_ID	898SPT
HOLE_IDREF	898
uom	
ISPT_TOP_UNIT	
#text	m/s
row	
row	
row	
row	
row	
ISPT_TOP	
ISPT_REP	
ISPT_TYPE	
row	
ISPT	
STCN	

m
ft
mm
cm
km
in
yd
mi

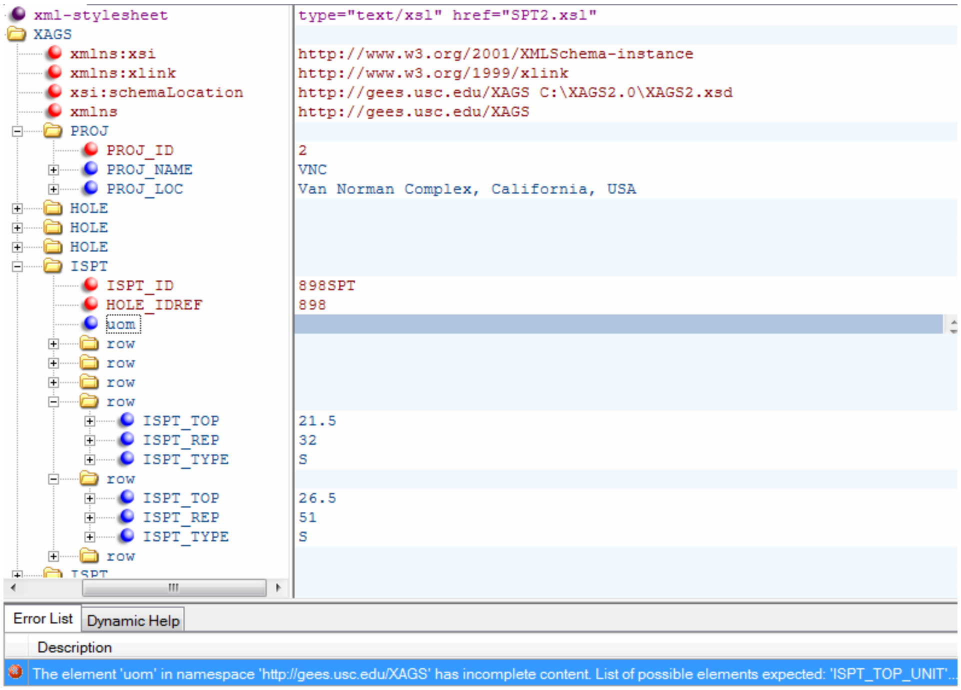
Error List | Dynamic Help

	Description	File	Line	Col
	The 'http://gees.usc.edu/XAGS:ISPT_TOP_UNIT' element is invalid - The value 'm/s' is invalid according to its ...	XMLCombo.xml	78	8

Figure 4.10 Wrong units and enumeration constraint fail error messages in the bottom window; drop down menu showing valid options

4.3.2. Missing Data Recognition and Unit Validation

As described in chapter three, data types for different fields are defined in XAGS schema. Simple and complex data types are also defined as part of XAGS schema. Other data constraints, such as required fields, and required units are incorporated in XAGS. A missing required element error is displayed in Figure 4.11 as an example. This required element is also a required unit. This is an example of how XAGS can recognize missing data and check for data validation as a proper data exchange format should.



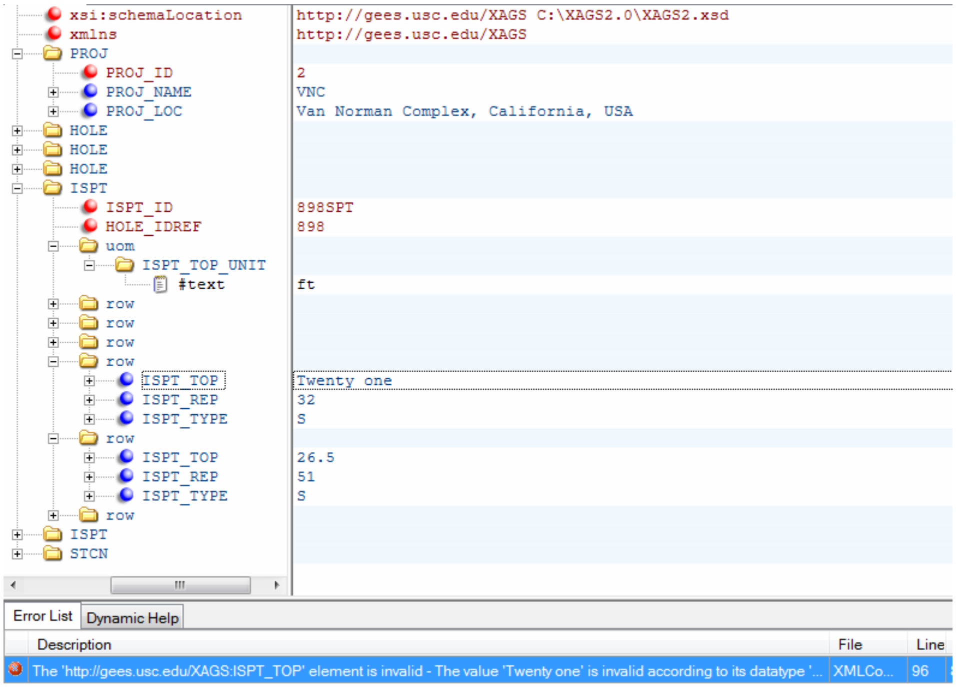
```
xml-stylesheet              type="text/xsl" href="SPT2.xsl"
XAGS
    xmlns:xsi               http://www.w3.org/2001/XMLSchema-instance
    xmlns:xlink             http://www.w3.org/1999/xlink
    xsi:schemaLocation      http://gees.usc.edu/XAGS C:\XAGS2.0\XAGS2.xsd
    xmlns                   http://gees.usc.edu/XAGS
    PROJ
        PROJ_ID             2
        PROJ_NAME           VNC
        PROJ_LOC            Van Norman Complex, California, USA
    HOLE
    HOLE
    HOLE
    ISPT
        ISPT_ID             898SPT
        HOLE_IDREF          898
        uom
        row
        row
        row
        row
            ISPT_TOP        21.5
            ISPT_REP        32
            ISPT_TYPE       S
        row
            ISPT_TOP        26.5
            ISPT_REP        51
            ISPT_TYPE       S
        row
    ISPT
```

The element 'uom' in namespace 'http://gees.usc.edu/XAGS' has incomplete content. List of possible elements expected: 'ISPT_TOP_UNIT'...

Figure 4.11   Missing required element error (ISPT_TOP_UNIT required element is missing)

4.3.3. Data Format and Range Validation

Other data constraints, such as data type, cardinality, default values, data length and acceptable ranges for numerical values are incorporated in XAGS. Figure 4.12 demonstrates an instance of data type validation error. ISPT_TOP should be of float data type value that is having a character string value in this example and should be modified.



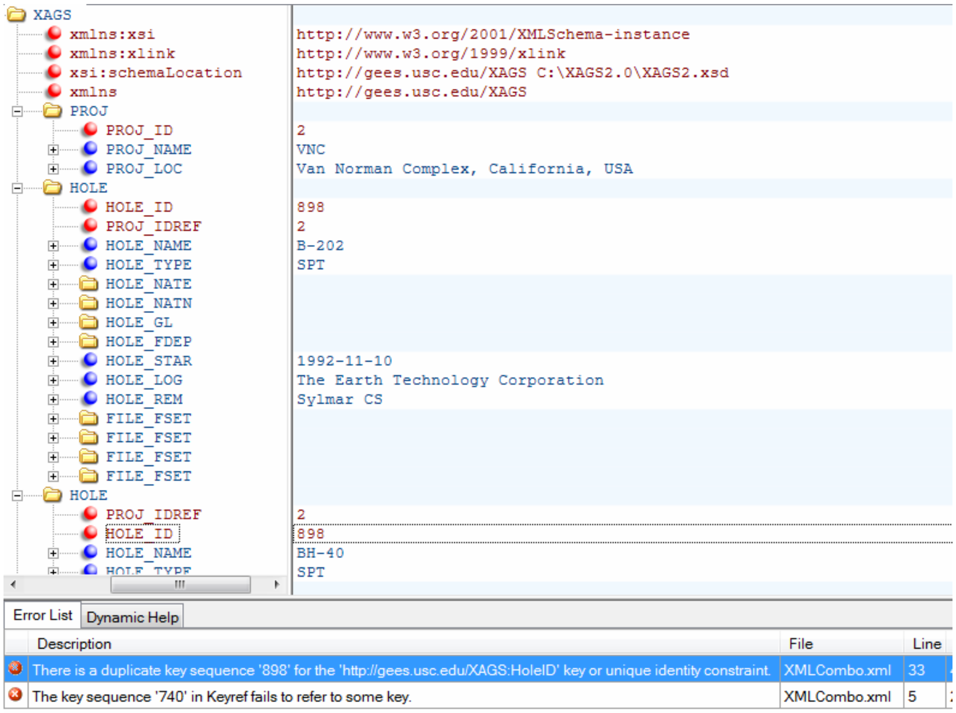
```
xsi:schemaLocation        http://gees.usc.edu/XAGS C:\XAGS2.0\XAGS2.xsd
xmlns                     http://gees.usc.edu/XAGS
PROJ
    PROJ_ID               2
    PROJ_NAME             VNC
    PROJ_LOC              Van Norman Complex, California, USA
HOLE
HOLE
HOLE
ISPT
    ISPT_ID               898SPT
    HOLE_IDREF            898
    uom
        ISPT_TOP_UNIT
            #text         ft
    row
    row
    row
    row
        ISPT_TOP          Twenty one
        ISPT_REP          32
        ISPT_TYPE         S
    row
        ISPT_TOP          26.5
        ISPT_REP          51
        ISPT_TYPE         S
    row
ISPT
STCN
```

Description	File	Line
The 'http://gees.usc.edu/XAGS:ISPT_TOP' element is invalid - The value 'Twenty one' is invalid according to its datatype '...	XMLCo...	96

Figure 4.12 Data type validation error; ISPT-TOP should be of float value while having a character string value in this example

4.3.4. Unique Relationships

Relationships between different data groups (defined in chapter three) are preserved with unique identifiers. If a duplicate identifier exists, an error message will be generated. Figure 4.13 features a duplicate HOLE_ID of 898 for two boreholes which has detected by the XML editor. Additionally, it shows that a SPT test is referring to a borehole with HOLE_ID of 740 which does not exist.



```
XAGS
    xmlns:xsi               http://www.w3.org/2001/XMLSchema-instance
    xmlns:xlink             http://www.w3.org/1999/xlink
    xsi:schemaLocation      http://gees.usc.edu/XAGS C:\XAGS2.0\XAGS2.xsd
    xmlns                   http://gees.usc.edu/XAGS
  PROJ
      PROJ_ID               2
      PROJ_NAME             VNC
      PROJ_LOC              Van Norman Complex, California, USA
  HOLE
      HOLE_ID               898
      PROJ_IDREF            2
      HOLE_NAME             B-202
      HOLE_TYPE             SPT
      HOLE_NATE
      HOLE_NATN
      HOLE_GL
      HOLE_FDEP
      HOLE_STAR             1992-11-10
      HOLE_LOG              The Earth Technology Corporation
      HOLE_REM              Sylmar CS
      FILE_FSET
      FILE_FSET
      FILE_FSET
      FILE_FSET
  HOLE
      PROJ_IDREF            2
      HOLE_ID               898
      HOLE_NAME             BH-40
      HOLE_TYPE             SPT
```

	Description	File	Line
●	There is a duplicate key sequence '898' for the 'http://gees.usc.edu/XAGS:HoleID' key or unique identity constraint.	XMLCombo.xml	33
●	The key sequence '740' in Keyref fails to refer to some key.	XMLCombo.xml	5

Figure 4.13 Duplicate HOLE_ID for borehole and a reference to a missing HOLE_ID error messages

4.4. Data Exchange between Teams and Team Members

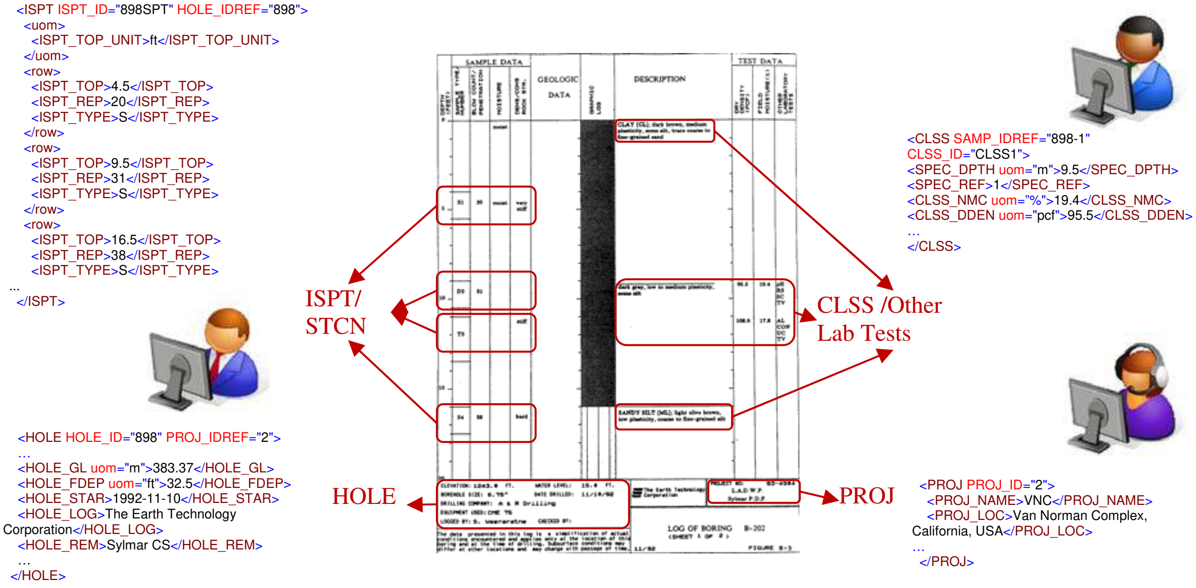
XML files are lightweight with a small size. The XAGS files can easily be emailed or transferred on any type of memory or via World Wide Web. Efficient data exchange plays a major role in the progress of projects.

4.4.1. Exchange of Complete Data Set

The full dataset can be exchanged as one file. The file can be validated against the schema in any XML editor as demonstrated earlier. The XSL file can transform the data file into an HTML file with links to Google Maps for spatial location viewing, as shown in Figure 4.8. It should be reminded again, that none of these requires any commercial software program.

4.4.2. Exchange of Data Subsets

Project engineer, field engineer, lab technician and other professionals work as a team to complete and deliver a geotechnical project. Each person might add or modify part of the data set based on their role in the project. Once a geotechnical project is finished the complete data set is sent to data owner or the next team such as the structural team for design of the infrastructures. It should be always considered that people in the chain of data exchange within a project team or with other teams come with different backgrounds in computer literacy. Therefore, exchange methods should be simple enough with the use of most common programs. As it can be seen in Figure 4.14, all the needed information for the borehole log is already defined in XAGS data dictionary and it comes from different pieces of XAGS.



```
<ISPT ISPT_ID="898SPT" HOLE_IDREF="898">
  <uom>
    <ISPT_TOP_UNIT>ft</ISPT_TOP_UNIT>
  </uom>
  <row>
    <ISPT_TOP>4.5</ISPT_TOP>
    <ISPT_REP>20</ISPT_REP>
    <ISPT_TYPE>S</ISPT_TYPE>
  </row>
  <row>
    <ISPT_TOP>9.5</ISPT_TOP>
    <ISPT_REP>31</ISPT_REP>
    <ISPT_TYPE>S</ISPT_TYPE>
  </row>
  <row>
    <ISPT_TOP>16.5</ISPT_TOP>
    <ISPT_REP>38</ISPT_REP>
    <ISPT_TYPE>S</ISPT_TYPE>
...
 </ISPT>
```

```
<CLSS SAMP_IDREF="898-1"
CLSS_ID="CLSS1">
<SPEC_DPTH uom="m">9.5</SPEC_DPTH>
<SPEC_REF>1</SPEC_REF>
<CLSS_NMC uom="%">19.4</CLSS_NMC>
<CLSS_DDEN uom="pcf">95.5</CLSS_DDEN>
...
</CLSS>
```

```
 <HOLE HOLE_ID="898" PROJ_IDREF="2">
 ...
 <HOLE_GL uom="m">383.37</HOLE_GL>
 <HOLE_FDEP uom="ft">32.5</HOLE_FDEP>
 <HOLE_STAR>1992-11-10</HOLE_STAR>
 <HOLE_LOG>The Earth Technology
Corporation</HOLE_LOG>
 <HOLE_REM>Sylmar CS</HOLE_REM>
 ...
</HOLE>
```

```
 <PROJ PROJ_ID="2">
  <PROJ_NAME>VNC</PROJ_NAME>
  <PROJ_LOC>Van Norman Complex,
California, USA</PROJ_LOC>
...
 </PROJ>
```

Figure 4.14 Different Data Resources for Compiling a Borehole Log

Figure 4.15 demonstrates a sample project data flow. A borehole log is used as a sample of a geotechnical project output to show how XAGS can help the development and exchange of data within the project. Project engineer is usually responsible to assign sub tasks to other members. In case of utilization of XAGS data format for the project, the project engineer will generate an empty XAGS file, assigns unique identifiers for different elements in the file and break the empty file to separate files for field tests and laboratory tests at the beginning of the project. Then the empty XAGS files will be sent to the appropriate project team member to be filled out as the data is produced. For example, the lab tests blank files will be sent to the laboratory technician and the in situ tests empty files will be sent to the field engineer. Therefore, not all the team members need to access all the XAGS files. This, in addition to other benefits, clearly decreases the potential for error and unintended data manipulation.

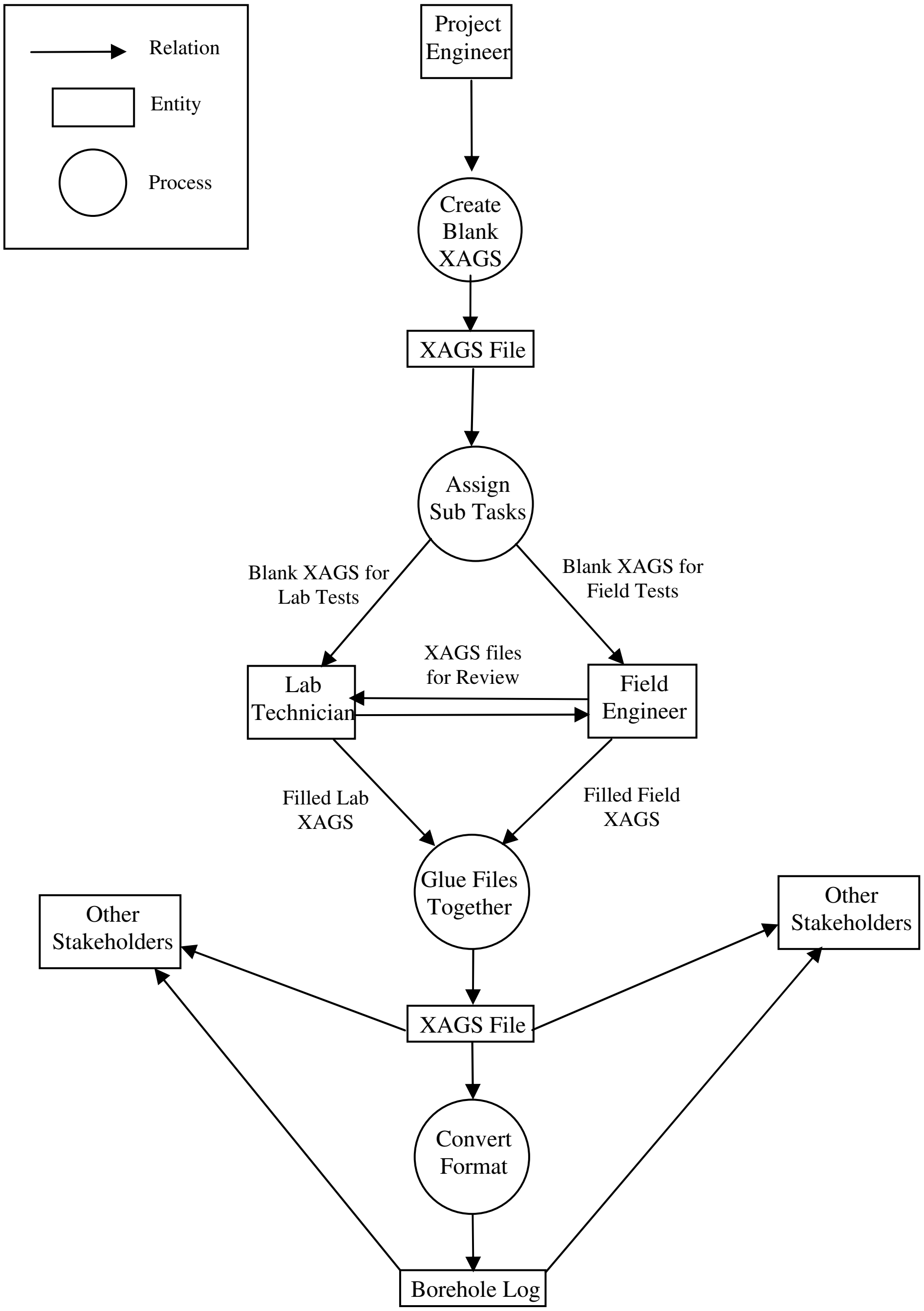


Figure 4.15 A high level scenario of XAGS data exchange within a project

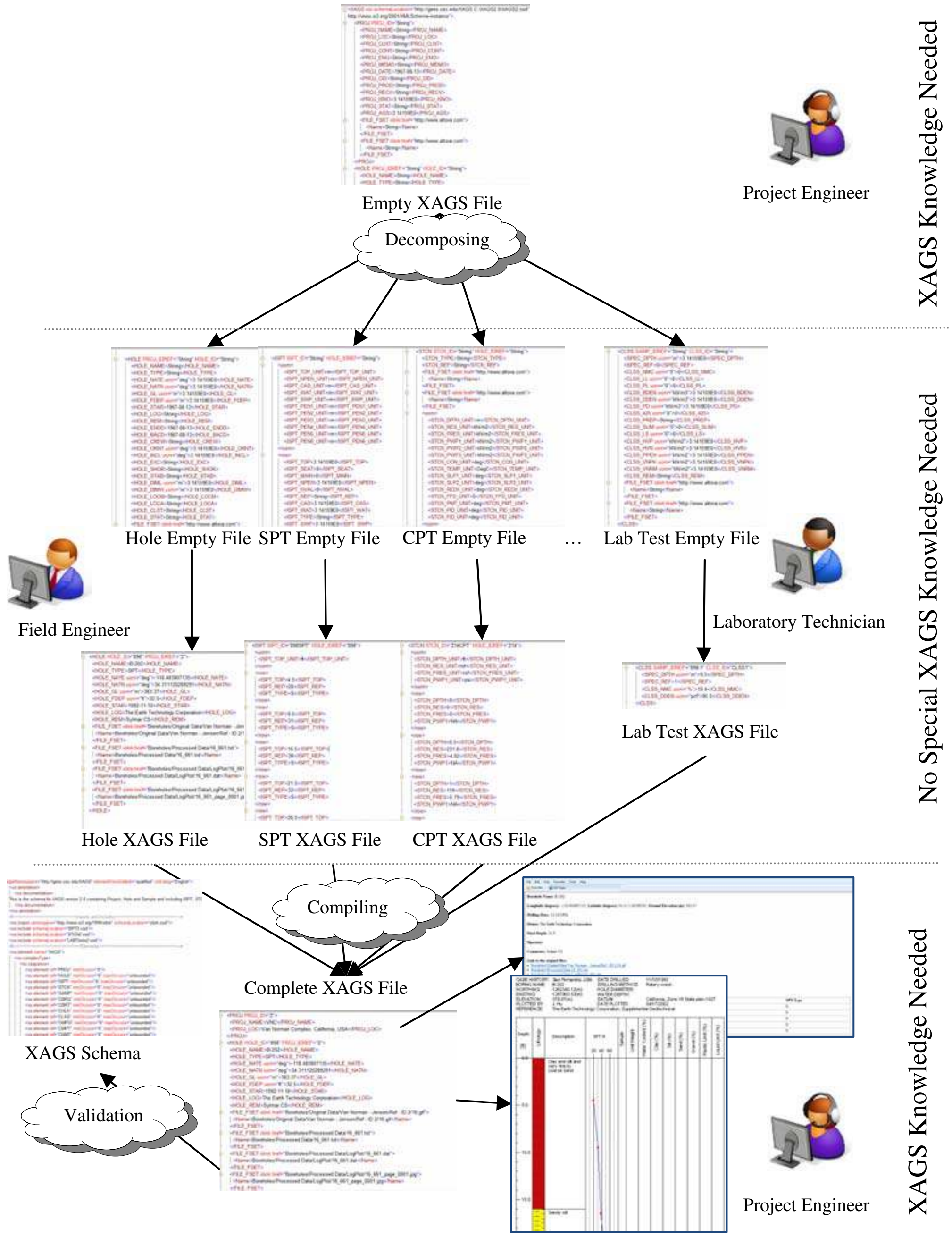


Figure 4.16 Process of Compiling a Borehole with Utilization of XAGS Data Exchange Format

Table 4.1 lists the elements that each team member will have access to during the project.

As shown in Figure 4.16, when a team works together to finish a geotechnical project, different members will work parallel to produce the needed data. The lab technician might run different tests on parallel or send the test results to the project engineer separately upon availability. XAGS is versatile and simple. As long as blank files are provided by the project engineer, lab technician or field engineer can fill in the data within the tags with pretty much no knowledge of the XAGS structure. XML editors can be used to view the data in the browser format with the help of XSL file for quality control. These separate files will be combined into one file by the project engineer later. The project engineer responsible for assigning sub tasks to other members and providing empty XAGS files will later compile the filled files received from team members. In this scenario, the project engineer needs some knowledge of XAGS structure to be able to merge all the files together in the correct form.

Table 4.1 A conceptual example of different hierarchy access levels of team members in a geotechnical projects

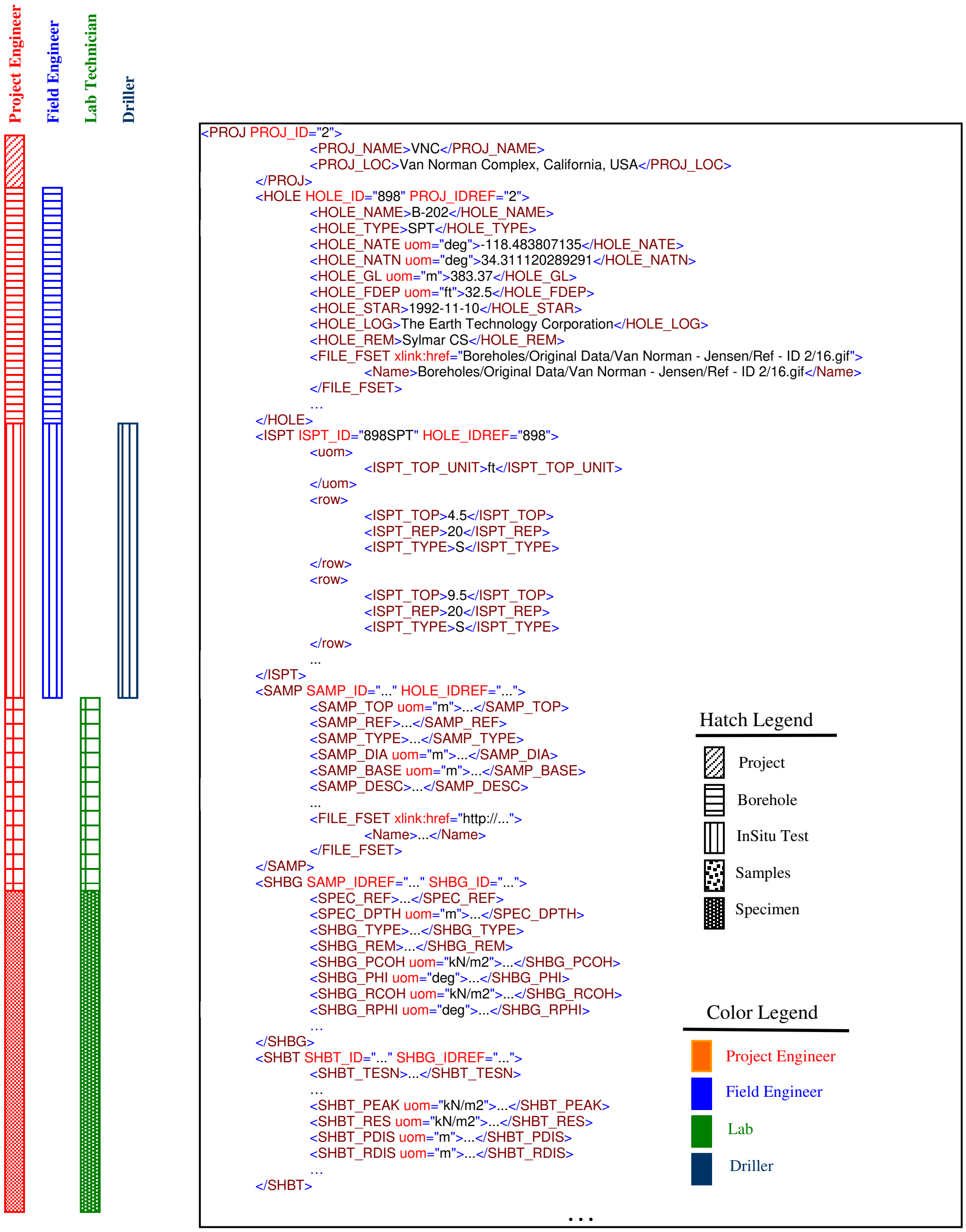
Hierarchy Level	Element	Team Members			
		Project Engineer	Field Engineer	Lab Technician	Driller
1	Project	🟥			
2	Borehole	🟥	🟦		
3	In Situ Test	🟥	🟦		🟦
3	Sample	🟥		🟩	
4	Specimen	🟥		🟩	

Figure 4.17 depicts a XAGS file, showing the part of data file that each team member has access or contributes to during the project. This complete file can be used in conjunction of developed XSL files to view the project data in HTML, tabular or borehole log forms as desired. Other people with interest in the project can request any of the files for review at any of the stages of project development. Validation of the final file against the schema will provide an additional step for quality control. Figure 4.14 through Figure 4.17 aim to clarify how data subsets are created and handled within a project and are integrated to produce the final result.

4.5. XAGS and Other Formats

4.5.1. Conversion of XAGS Data to Other Formats

In order to provide backward compatibility with some older existing geotechnical applications, a very simple XSL transformation file can be developed to convert XAGS data to any other format. Zand (2005) demonstrates transformation from XML-based information into AGS format. The same procedure can be modified to transform XAGS data into AGS format with the help of an XSL file.



```
<PROJ PROJ_ID="2">
            <PROJ_NAME>VNC</PROJ_NAME>
            <PROJ_LOC>Van Norman Complex, California, USA</PROJ_LOC>
      </PROJ>
      <HOLE HOLE_ID="898" PROJ_IDREF="2">
            <HOLE_NAME>B-202</HOLE_NAME>
            <HOLE_TYPE>SPT</HOLE_TYPE>
            <HOLE_NATE uom="deg">-118.483807135</HOLE_NATE>
            <HOLE_NATN uom="deg">34.311120289291</HOLE_NATN>
            <HOLE_GL uom="m">383.37</HOLE_GL>
            <HOLE_FDEP uom="ft">32.5</HOLE_FDEP>
            <HOLE_STAR>1992-11-10</HOLE_STAR>
            <HOLE_LOG>The Earth Technology Corporation</HOLE_LOG>
            <HOLE_REM>Sylmar CS</HOLE_REM>
            <FILE_FSET xlink:href="Boreholes/Original Data/Van Norman - Jensen/Ref - ID 2/16.gif">
                  <Name>Boreholes/Original Data/Van Norman - Jensen/Ref - ID 2/16.gif</Name>
            </FILE_FSET>
            …
      </HOLE>
      <ISPT ISPT_ID="898SPT" HOLE_IDREF="898">
            <uom>
                  <ISPT_TOP_UNIT>ft</ISPT_TOP_UNIT>
            </uom>
            <row>
                  <ISPT_TOP>4.5</ISPT_TOP>
                  <ISPT_REP>20</ISPT_REP>
                  <ISPT_TYPE>S</ISPT_TYPE>
            </row>
            <row>
                  <ISPT_TOP>9.5</ISPT_TOP>
                  <ISPT_REP>20</ISPT_REP>
                  <ISPT_TYPE>S</ISPT_TYPE>
            </row>
            ...
      </ISPT>
      <SAMP SAMP_ID="..." HOLE_IDREF="...">
            <SAMP_TOP uom="m">...</SAMP_TOP>
            <SAMP_REF>...</SAMP_REF>
            <SAMP_TYPE>...</SAMP_TYPE>
            <SAMP_DIA uom="m">...</SAMP_DIA>
            <SAMP_BASE uom="m">...</SAMP_BASE>
            <SAMP_DESC>...</SAMP_DESC>
            ...
            <FILE_FSET xlink:href="http://...">
                  <Name>...</Name>
            </FILE_FSET>
      </SAMP>
      <SHBG SAMP_IDREF="..." SHBG_ID="...">
            <SPEC_REF>...</SPEC_REF>
            <SPEC_DPTH uom="m">...</SPEC_DPTH>
            <SHBG_TYPE>...</SHBG_TYPE>
            <SHBG_REM>...</SHBG_REM>
            <SHBG_PCOH uom="kN/m2">...</SHBG_PCOH>
            <SHBG_PHI uom="deg">...</SHBG_PHI>
            <SHBG_RCOH uom="kN/m2">...</SHBG_RCOH>
            <SHBG_RPHI uom="deg">...</SHBG_RPHI>
            …
      </SHBG>
      <SHBT SHBT_ID="..." SHBG_IDREF="...">
            <SHBT_TESN>...</SHBT_TESN>
            …
            <SHBT_PEAK uom="kN/m2">...</SHBT_PEAK>
            <SHBT_RES uom="kN/m2">...</SHBT_RES>
            <SHBT_PDIS uom="m">...</SHBT_PDIS>
            <SHBT_RDIS uom="m">...</SHBT_RDIS>
            …
      </SHBT>
                                                  . . .
```

Figure 4.17 A complete XAGS file demonstrating access and contribution of different team members in production of final data set

4.5.2. Conversion of Existing Data Sets to XAGS Format

To transform the data from other formats such as AGS to XAGS format, a parser might be needed. An existing case history dataset is used to demonstrate conversion of existing data to XAGS format. Van Norman Complex (VNC) project is selected from a liquefaction-induced deformations study (Bardet et al. 1999). The site includes major water facilities operated by the Los Angeles Department of Water and Power (LADWP), such as the Los Angeles Reservoir with the Los Angeles Dam and the North Dike, the Los Angeles Aqueduct Filtration Plant, the Van Norman Bypass Reservoir, and the former Upper and lower Dams. This particular site witnessed two damaging earthquakes: the 1971 San Fernando earthquake and the 1994 Northridge earthquake in the past 40 years. After these earthquakes, ground failures (e.g., sand boil, surfacial cracks, lateral spreads and settlements) were widely observed within the VNC vicinity. One of the major reasons causing those damages was attributed to liquefaction in saturated granular soil deposits (USGS 1996). In response to the damages observed, in-depth post-earthquake investigations were performed to geotechnically characterize the site and measure liquefaction-induced ground deformations. The USC database (Bardet et al. 1999) includes site investigation information of the case history and has been used in Hu (2003) and Liu (2008) for further understanding of the mechanism of liquefaction-induced lateral ground deformations, and to evaluate the existing empirical procedures developed in the past. The USC database was originally stored in a relational Microsoft Access DBMS. Table 4.2 lists the main tables in the Access database.

Table 4.2 Main Tables defined in Access DB

Table Name	Description
TblLocation	Defines location of each project
TblCPTInfo	General information on CPT tests
TblSPTInfo	General information on SPT tests
TblCPT	CPT test results
TblSPT_Blow	SPT test results
TblSPT_Sam	Sample information and properties

Figure 4.18 illustrates a partial diagram of the database. There are five major tables: Location, SPTInfo, CPTInfo, SPT_Blow and CPT. They are illustrated as square boxes in Figure 4.18 with their main attributes denoted. Location relates to SPTInfo and CPTInfo by LocationID. SPT and CPT are spatial objects with attributes of location (Easting and Northing). The most valuable data associated with the site characterizations are the SPT and CPT boring logs. Additional attributes of geotechnical boreholes, although not listed in Figure 4.18, were in fact included in the data structure, such as drilling date, drilling equipment, ground water depth. Data were transferred into XAGS format with the help of Visual Basic coding (Figure 4.19). Then, XSL transformation can produce HTML files from the XAGS files.

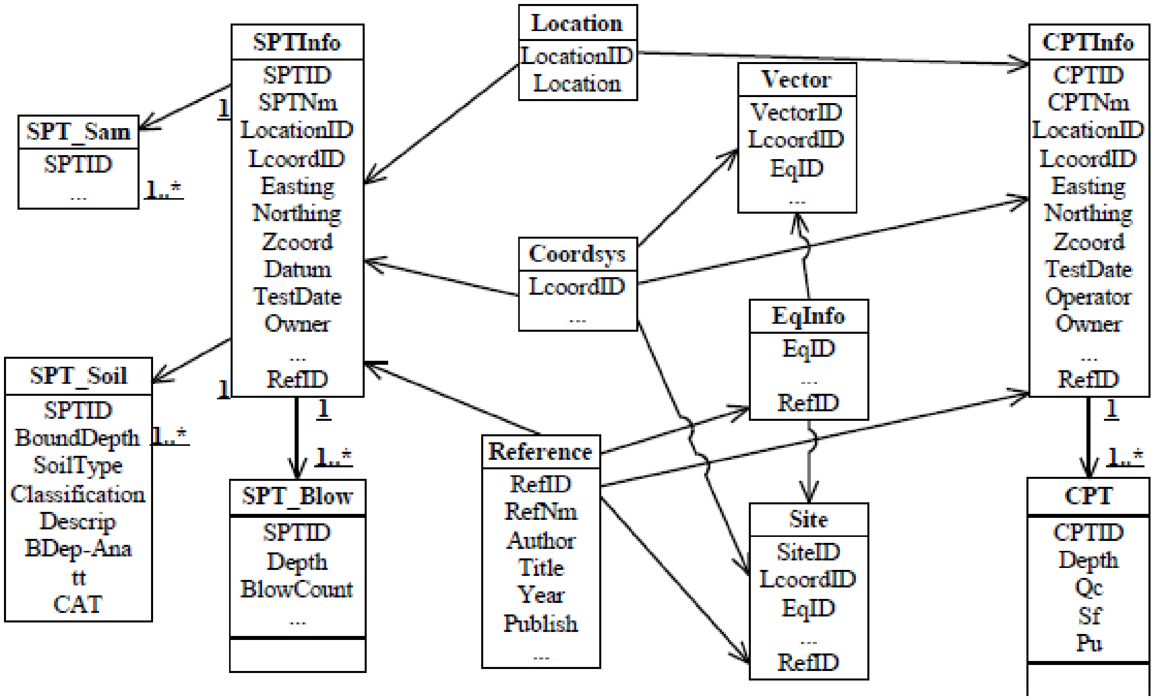


Figure 4.18 Partial diagram of liquefaction data set

Zand (2005) illustrates transformation of AGS files to XML format with a standard implementation of DOM (Document Object Model) to build the XML files. Then, a Java application is developed in order to transform the data files from AGS format to XML format (Zand 2005). The same procedure can be adopted to transform AGS or other formats to XAGS.

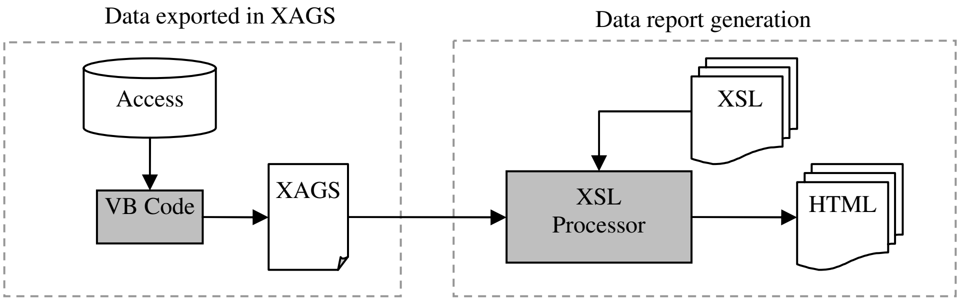


Figure 4.19 From Access database to HTML presentation of XAGS data

4.6. Archive and Distribution via World Wide Web

Web-service techniques were proposed to tunnel the geotechnical data and massive potential users via World Wide Web by Zimmermann et al. (2006). After introduction of AJAX technologies (Asynchronous JavaScript and XML) (Wusteman and O'hlceadha 2006), Liu (2008) proposed a new information system to distribute case histories over the Web. The proposed system benefited from AJAX, and Web and database technologies. Now with development of XAGS, we are looking for an even simpler system to disseminate geotechnical data.

4.6.1. System Architecture

As previously mentioned in this chapter, Google Maps can be used for spatial representation of borehole locations. Google Maps is a web mapping service application that can be embedded into an external website, on to which site specific data can be overlaid. Google Maps offers a quick solution for lightweight web-mapping systems such as the ones presented in Liu (2008). Google Maps is not a product designed for spatial analysis. However, it is an ideal solution to those mapping applications whose primary function is to visualize and disseminate location-related data with limited expectation of data analysis. Liu (2008) argues that Google Maps enables fast mapping performance due to the presence of AJAX technology. As the Google Maps code is almost entirely JavaScript and XML, the degree of difficulty is relatively low for developing a Google Maps application in comparison to the application of Web services. However, since XAGS data is already in an XML format, the architecture can be simplified one more step.

It is proposed to construct a KML file that lists the location of all the boreholes in the data set with the valid URL pointing to the XAGS files for each borehole. With this KML file, the need for the third party program is eliminated. The proposed system consists of three components: (1) Google Maps/Google Earth front-end; (2) KML file as the middle component between front end application and backend data and (3) XAGS data files as back end. Figure 4.20 conceptualizes the information system architecture for distribution and display of the VNC dataset in XAGS with the aid of Google Maps/Google Earth. A Google Maps interface interacts with clients and handles their requests. Upon user request, data are retrieved from the XAGS files through KML links, and are ultimately presented in the front end as HTML pages. As explained in Section 4.2.3, this KML files is created by an XSL Transformation on the XAGS files.

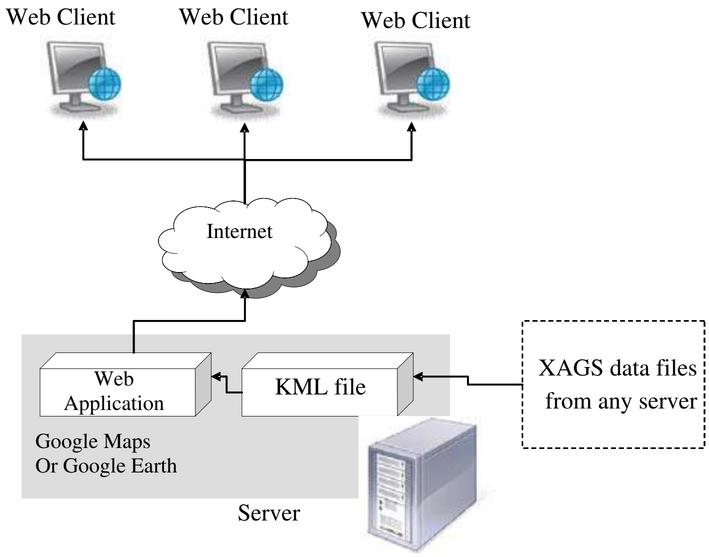


Figure 4.20 Architecture for XAGS data distribution via Web (After Liu 2008). Only a file server and a KML index file are needed. No Database System is required.

4.6.2. User Interface and Major Functions

Figure 4.21 shows the entry page for the proposed system using Google Maps. The map type can be switched among three modes, i.e., street map, satellite imagery, and terrain map, by clicking on the switch buttons at the top right corner. The map is zoomable through a navigation bar on the left hand side. In this entry page, by selecting SPT and/ or CPT buttons the borehole locations are displayed. The clickable balloons have a hidden information window that will appear upon clicking and provides information about the selected borehole. The SPT and CPT layers can be toggled on and off.

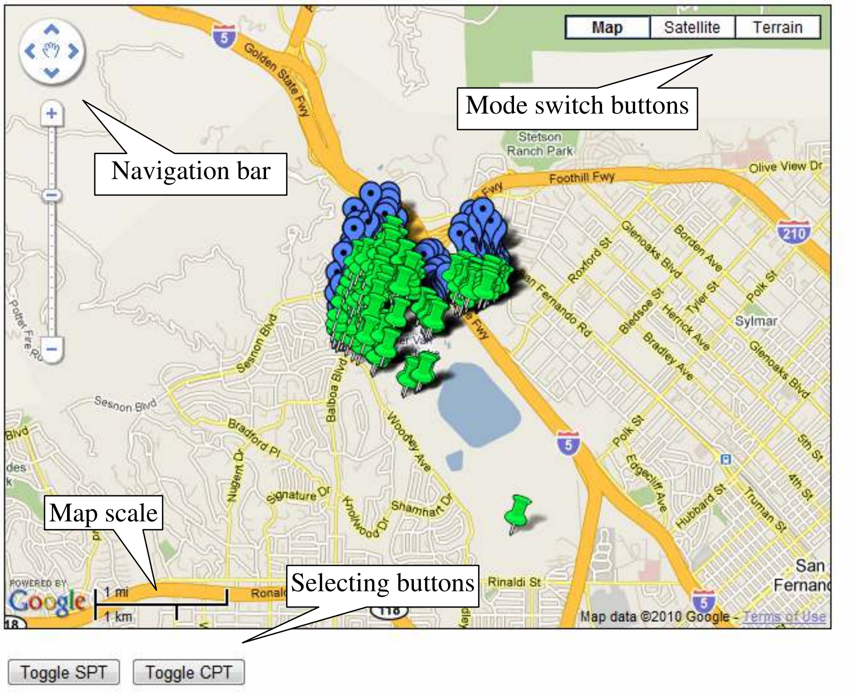


Figure 4.21 Available boreholes for the Van Norman Complex, San Fernando, California (map from Google Maps)

Figure 4.22 shows an interface using Google Earth satellite imagery that contains the same data set as Figure 4.21.

The system can conduct the following three functions:

(1) Visualization of location of boreholes. Once the SPT and/or CPT data set buttons are selected, the KML files are parsed and borehole locations are plotted on Google Maps or Google Earth.

(2) Visualization of geotechnical boreholes. Once the web user clicks on any borehole, the XAGS file including the information is called. As the XAGS file is including the

address to the XSL file, the XSL file is fetched and the XAGS file is parsed against the XSL file to produce an HTML file including the borehole information and the actual in situ test results in a tabulated format in the HTML page.

(3) Download data. Data can be downloaded as a plain text that tabulates relevant information, e.g., the components of a borehole, and its coordinates. Boring log files are also downloadable.

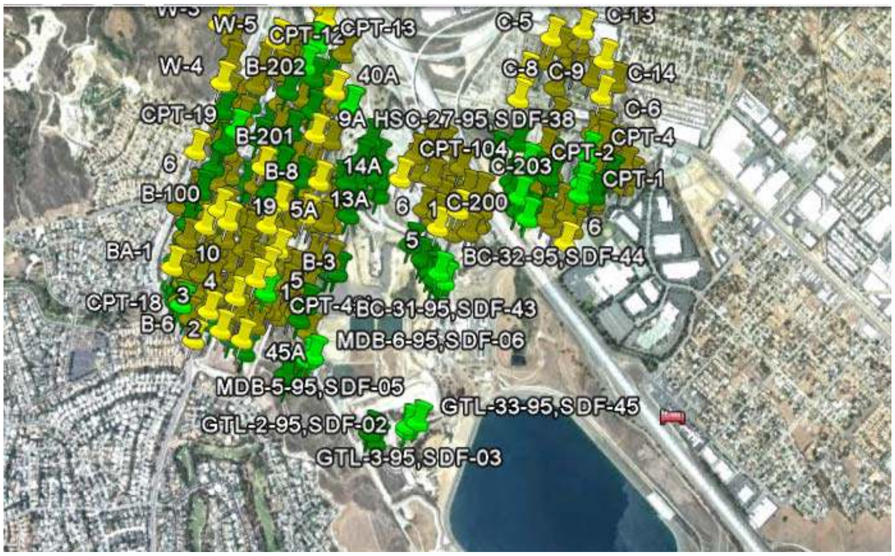
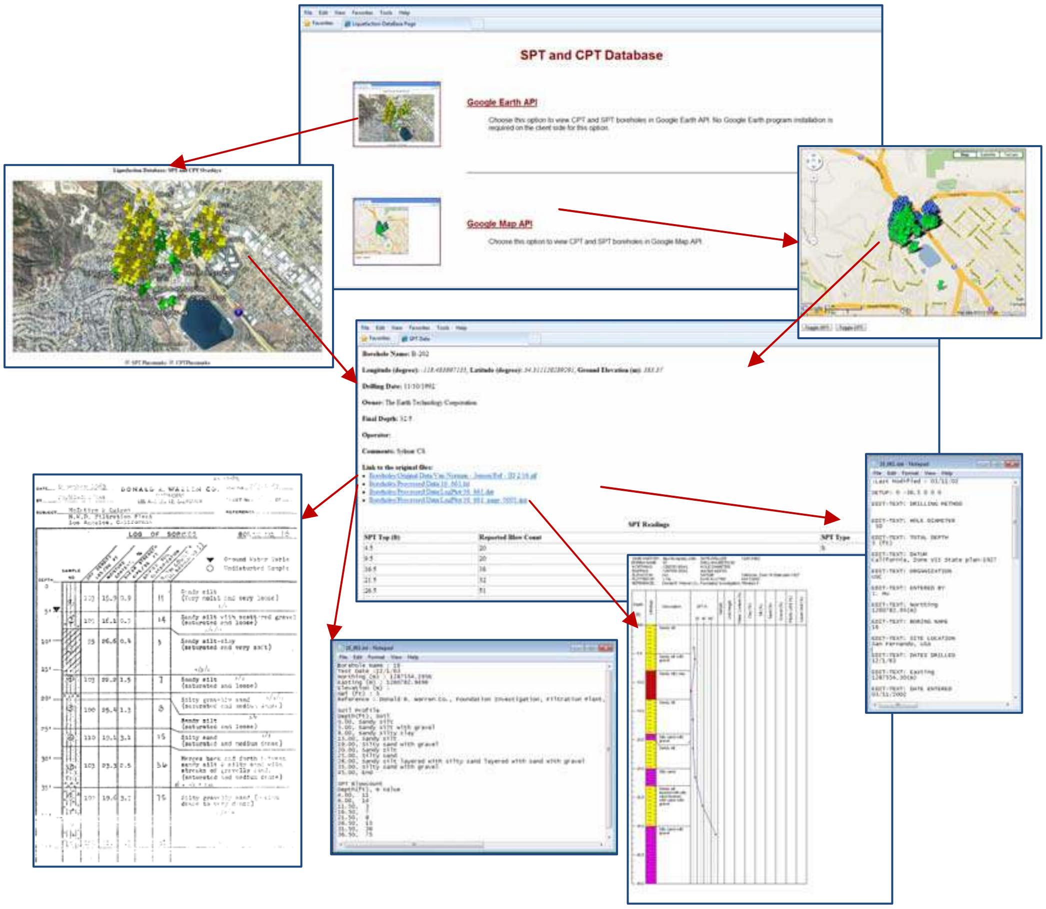


Figure 4.22 Satellite imagery as the background for available boreholes for the Van Norman Complex, San Fernando, California (satellite imagery from Google Earth)

Figure 4.23 shows a global view of XAGS data dissemination through web serving. It should be noted that the system proposed with the help of XAGS data format and the KML layer is far more simplified than the light weight system proposed with the third party program (AJAX, Liu 2008). This proposed system only needs familiarity with XML program, as KML like XAGS is only a data format defined within XML, compare to the previous methods of data serving that needed extensive programming in either AJAX (Liu 2008) or web services (Zimmerman et al. 2006).



File Edit View Favorites Tools Help
Favorites
Liquefaction Database Page

SPT and CPT Database

Google Earth API

Choose this option to view CPT and SPT boreholes in Google Earth API. No Google Earth program installation is required on the client side for this option.

Google Map API

Choose this option to view CPT and SPT boreholes in Google Map API.

File Edit View Favorites Tools Help
Favorites
SPT Data

Borehole Name: B-202

Longitude (degree): -118.483807135, **Latitude (degree):** 34.311120289291, **Ground Elevation (m):** 383.37

Drilling Date: 11/10/1992

Owner: The Earth Technology Corporation

Final Depth: 32.5

Operator:

Comments: Sylmar CA

Link to the original files:

- Boreholes/Original Data/Van Norman - Jensen/Ref - ID 2.16.pdf
- Boreholes/Processed Data/18_B83.txt
- Boreholes/Processed Data/LogPlot/18_B83.dat
- Boreholes/Processed Data/LogPlot/18_B83_page_0001.jpg

SPT Readings

SPT Top (ft)	Reported Blow Count	SPT Type
4.5	20	S
9.5	20	
16.5	16	
21.5	32	
26.5	51	

```
Borehole Name : 18
Test Date :12/1/83
Northing (m) : 1287154.2956
Easting (m) : 1280782.9498
Elevation (m) :
GWT (ft) : 5
Reference : Donald R. Warren Co., Foundation Investigation, Filtration Plant,

Soil Profile
Depth(ft), Soil
5.00, Sandy silt
7.00, Sandy silt with gravel
8.00, Sandy silty clay
13.00, Sandy silt
19.00, Silty sand with gravel
20.00, Sandy silt
25.00, Silty sand
26.00, Sandy silt layered with silty sand layered with sand with gravel
35.00, Silty sand with gravel
45.00, End

SPT Blowcount
Depth(ft), N value
4.00, 11
6.00, 14
11.50, 7
16.50, 7
21.50, 8
26.50, 13
31.50, 38
36.50, 75
```

```
:Last Modified : 01/11/02

SETUP: 0 -16.5 0 0 0

EDIT-TEXT: DRILLING METHOD


EDIT-TEXT: HOLE DIAMETER
 50

EDIT-TEXT: TOTAL DEPTH
5 (ft)

EDIT-TEXT: DATUM
California, Zone VII State plan-1927

EDIT-TEXT: ORGANIZATION
USC

EDIT-TEXT: ENTERED BY
I. Hu

EDIT-TEXT: Northing
1280782.95(m)

EDIT-TEXT: BORING NAME
18

EDIT-TEXT: SITE LOCATION
San Fernando, USA

EDIT-TEXT: DATES DRILLED
12/1/83

EDIT-TEXT: Easting
1287554.30(m)

EDIT-TEXT: DATE ENTERED
01/11/2002
```

Figure 4.23 XAGS data distribution through web serving

4.7. XAGS Evaluation

The requirement list developed in this research specifically for assessment of geotechnical exchange data exchange formats is used here to evaluate XAGS in this section and compare it with the existing data formats in Table 4.3.

General criteria

G1) Is naming convention consistent? Use of AGS data dictionary makes the naming convention for XAGS consistent, as AGS data dictionary is reviewed with the geotechnical community several times during the revisions of AGS format and the ambiguities are eliminated.

G2) Are unit abbreviations according to standard units? Choices of units are unique, clear and present with the help of the unit enumeration list borrowed form AGS data dictionary.

G3) Can subsets be assembled back to make a complete data set? In this chapter, it was discussed in detail how subsets of data are exchangeable and can be easily integrated back to the complete data set. For establishing the unique relationships, key and keyref elements were introduced in the schema.

G4) Does data exchange come from well accepted test standards? XAGS is calling upon British Standards through AGS data dictionary that is an already acceptable test standard and is used in some European countries. However, for better acceptance in United States conversion to ASTM tests is recommended. At the time of writing this thesis there was no well defined data dictionary known to conform to ASTM standards. Developing such data dictionary needs a lot of time and effort and is due to general consensus of community and many years of review and revision. Therefore, it was outside the scope of this research. However, if such data dictionary exists, development of another version of XAGS with the guidelines of this dissertation will be very easy.

G5) Can data be exchanged via email? The small size of files will makes exchange of XAGS files via email easy and fast. It will be possible to exchange files with email, USB and other hardware as well.

G6) Can the defined data exchange format be accepted and used by community? It is demonstrated that the exchange format is simple enough for the community to a) understand, b) use and c) convert to the existing standards and data.

Data/ technical specific criteria:

D1) Are all variables defined based on common knowledge in the field using plain English (or other languages)? In XAGS, users are not responsible for naming the fields or measurement types. Variables are already defined based on common knowledge in the field of geotechnical engineering.

D2) Are all values associated with appropriate units? XAGS validation methods checks that the unit for each element in the file is selected form a list of valid units defined in the unit enumeration list. Since the schema design has been kept simple, validation of files are done in very reasonable amount of time.

D3) Are the values in the right format, e.g. integer, string? XAGS schema explicitly defines the data type that each element can contain. An error message will be passed to the user if any of the values in the XAGS file is not in the right format.

D4) Are the values within an acceptable range? A proper validation method for the range of values exists in XAGS schema. If the values are not in the acceptable range, an error message will be passed to the user.

D5) Are missing fields recognized with error messages? If a required element is missing in the XAGAS file it will be distinguished during the validation against the schema with some error messages.

D6) Can exchange data be used in different platforms and operating systems? XAGS format is platform independent and operating system independent.

D7) Is the data format readily supported by commercial and non-commercial tools? The content of a XAGS file can be viewed, tabulate and located on maps with non-commercial programs. Commercial programs are able to read the data as well.

D8) Can data be indexed for archival so that it can be retrieved later? As demonstrated in this chapter, XAGS data can be easily archived and retrieved for later use, locally or over the web.

D9) Is it possible to search within the data file for particular value or attribute, using some kind of search engine? XAGS data is searchable data; XQuery method for searching the files is available.

D10) Can data be located on interactive maps such as Google Maps? XAGS data is spatial. In other words, XAGS will relate the information to its location.

D11) Are all objects in data following GIS accepted format, so that all objects can be positioned in 3 dimensional spaces? XAGS data is not GIS-enabled. GIS-enabled data will be identified and presented by GIS software programs. This is not possible for XAGS. It was concluded that considering the geotechnical community background in data information technologies at the current state, usage of GIS conformant formats will make the data format complicated for the users. As some of the previous efforts showed this complication is beyond the threshold of the community at this time.

D12) Is the exchange format capable of identifying missing objects or objects with non-unique or missing relations? XAGS key and keyref system makes sure there is no ambiguity in the relationships. As shown in this chapter, when relationships are duplicated the XAGS file will not be validated and an error message is passed to the user.

D13) Is data format free of circular relations between objects which create infinite circular loops? XAGS is free of any circular relations as the tree structure used in version 1.0 clearly shows that no loop exists in the data.

D14) Is it possible to generate random instances of objects for populating data? XAGS empty files can be easily generated with any XML editor for later insertion of data.

D15) Can all relationship identifiers be replaced with object attributes to form tables without any additional relationships? XAGS has a flatable structure; the data structure is designed in a way that flattening of data will be possible. For flattening, identifiers will be replaced with object attributes. Therefore, relationships are gone but the data is available. This test ensures that the data can be later archived in an RDBMS, if necessary.

D16) Is data exchange format capable of holding annotations (additional information)? XAGS carries Remark for each entity for holding annotations.

D17) Are annotations free of structured metadata (data about data)? XAGS clearly keeps the test results only, e.g. data. No metadata is mixed with the data.

Table 4.3 Comparison of the data formats in satisfying data exchange format requirements for geotechnical projects

	Criteria	AGS	DIGGS	XAGS
G1	Is naming convention consistent?	yes	no	yes
G2	Are unit abbreviations according to standard units?	yes	no	yes
G3	Can subsets be assembled back to larger data sets?	maybe	maybe	yes
G4	Does it come from well accepted test standards?	yes	no	yes
G5	Can data be exchanged via email?	yes	no	yes
G6	Can it be accepted and used by the community?	yes	maybe	yes
D1	Are all variables defined based on common knowledge in the field?	yes	no	yes
D2	Are all values associated with units?	no	no	yes
D3	Are the values in the right format, e.g., integer, string?	no	yes	yes
D4	Are the values being validated to be in acceptable range?	no	yes	yes
D5	Are missing fields recognized with error messages?	no	no	yes
D6	Can exchange data be used in different platforms and operating systems?	yes	yes	yes
D7	Is the data format readily supported by non-commercial and commercial tools?	maybe	no	yes
D8	Can data be indexed for archival and later retrieval?	no	maybe	yes
D9	Is it possible to search within the data file for particular value or attribute, using a search engine?	no	yes	yes
D10	Can data be located on interactive maps?	no	yes	yes
D11	Are all objects in data following GIS accepted format so that they can be positions in 3 dimensional spaces?	no	maybe	no
D12	Is the exchange format capable of identifying missing objects or objects with non-unique or missing relations?	no	no	yes
D13	Is data format free of circular relations between objects which create infinite circular loops?	yes	no	yes
D14	Is it possible to generate random instances of objects for populating data?	no	no	yes
D15	Can all relationship identifiers be replaced with object attributes to form tables without any more relationships?	yes	no	yes
D16	Is data exchange format capable of holding annotations (additional information)?	yes	yes	yes
D17	Are annotations free of structured metadata?	yes	no	yes

4.8. Summary

Field and office crews working on a geotechnical project all have different backgrounds in computer literacy. Hence, for proposing a successful data exchange format the main objective should be simplicity of application, usage, generation, modification and validation of data. XML editors were introduced for random data generation, data modification and data validation. It was demonstrated that spreadsheets such as Excel spreadsheets can also be used for XAGS data viewing, input and modification with some simple XAGS-Excel mapping. XSL was used to present XAGS data in HTML format, including tabular data and presenting the location spatially on Google Maps. Transformation from XAGS to AGS and other formats and from AGS and other formats to XAGS was discussed. It was illustrated that some of the important requirements for a proper data exchange format that are missing in AGS are fully satisfied with XAGS schema. XAGS validation process distinguishes structural problems such as unit enumeration errors, missing data, unit errors, data type and range errors and non-unique relationships and appropriate error messages are passed to the user to modify the data for schema conformity. Validation process can be done with any available XML editor program, including many free options already available to public. Change of the XML editor program will have no effect in the validation process. Additionally, exchange of XAGS complete data sets and subsets were discussed between teams and team members. As a validation of XAGS format, a new information system was developed to distribute a liquefaction study case history in XAGS format over the Word Wide Web. The system enables a lightweight Web application with responsive and user-friendly interfaces to disseminate geotechnical investigation data. Using the results of this validation, XAGS is evaluated against the criteria list developed for proper data exchange formats and is compared with some of the data formats reviewed in chapter two. It is shown that even though XAGS does not satisfy all the requirements for data exchange format, it is a great improvement and big step forward compared to the existing options.

Chapter 5. Future Direction: Metadata Models

Previous chapters defined a format for exchanging data when the process is well-defined. However, not all processes are standard. For not well-documented data and additional layer of information is needed. In future, as data exchange gets more comprehensive and versatile, standards will not cover all the geotechinical data. This will create the need for interpretation of data by geotechinical community as data being exchanged. This interpretation, most probably, will increase the chance of misunderstanding and misuse of data. To avoid this, additional data will be needed to understand how data is obtained. The modeling of this data about data (commonly called metadata) is proposed for linking procedures with data. Metadata should be kept separate from actual data. In most cases the users of data and metadata are different entities and the person interested to study the results is not usually interested in the metadata. It is somehow a necessary part of information exchange for more experimental and research-oriented and collaborative tests.

This chapter discusses the development of a metadata model for collaborative experimental research in geotechnical earthquake engineering as a sample of not well-documented data. Most of the work presented in this chapter is already published (Bardet et. al., 2010). The model proposed here has been used by Lee (2006) to document centrifuge tests preformed at USC. In the future, the same guidelines could be implemented to define a metadata model for not well-documented data of geotechnical projects.

5.1. Background

Projects in geotechnical earthquake engineering yield a very large amount of complex data from experiments and computer simulations that are not well-documented or covered by standards. Understanding and exchanging these complicated and voluminous data sets prompted the development of metadata models that document the processes of data generation, and facilitate the collaboration and exchange of information between researchers. The present metadata model was designed to document and exchange a large number of large data files in geotechnical earthquake engineering, but is applicable to other fields of engineering and science. The model was conceived based on a series of former metadata models, which were unduly complicated and limited to few types of experiments. Simpler than its predecessors, the present metadata model applies to all kinds of geotechnical earthquake engineering experiments. It was developed in the object-oriented framework using Protégé. Its applications are illustrated with examples from centrifuge experiments.

5.2. Metadata Modeling History for Collaborative Research

Fifteen years ago, the experiments and computer simulations in geotechnical earthquake engineering were carried out mostly by single investigators at their own institutions; their resulting data sets were rarely documented in great detail, and only a small fraction of these data sets were reported in technical reports, journal papers and conference proceedings. In the last fifteen years, a few collaborative research projects in earthquake engineering (VELACS, ROSRINE, COSMOS, and CUREE) have pioneered the documentation, preservation and exchange of research data sets. At the turn of the 21rst century, the George E. Brown Jr. Network for Earthquake Engineering Simulation (NEES) expanded the realm of experiments and computer simulations in earthquake engineering by promoting collaborative experiments and computer simulations. NEES developed an early data model, which organizes data in relation to equipment, software, individuals and organizations (Bardet et. al., 2004; Peng and Law, 2004). The model was revised numerous times to accommodate a flurry of user requirements, and gradually became too complicated for practical use. It introduced too many objects, defined them without any convention, and had difficulties to relate objects with one another (Swift et. al., 2004). This data model was the first attempt to apply recent advances in information technology for documenting data in earthquake engineering.

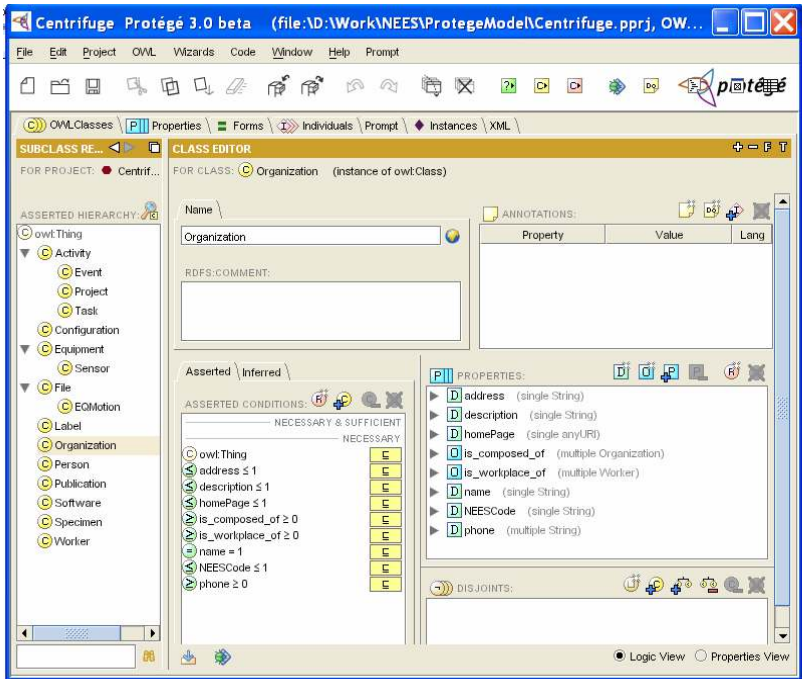
This chapter describes a new metadata model which stems from former data models. Although the model was initiated for documenting research results in earthquake engineering, it is also applicable to other collaborative research in engineering and sciences.

5.3. Background on Metadata Modeling

The proposed metadata model is easier to understand after reviewing a few concepts of object-oriented (O-O) modeling. O-O models represent information as objects (Rumbaugh et. al., 1991). An object encapsulates related data as attributes, and connects with other objects through relationships. These relationship types may in turn impose certain O-O features (e.g., inheritance). O-O data model can be formulated using the Object Definition Language (ODL) and the Unified Modeling Language (UML). ODL (Rumbaugh et. al., 1991) is a proposed standard language for writing and translating directly O-O models into O-O databases. UML (Arlow and Neustadt, 2001) is the industry-standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. O-O modeling and programming are covered by many textbooks (Rumbaugh et. al., 1991; Booch, 1993; Gamma et. al., 1993; Meyer, 1997; Jacobsen, 1994). O-O modeling is based on the concept of an "object" which is a data structure (abstract data type). O-O programming encapsulates these objects with a set of routines called "methods."

A few terms (i.e., class, object, attribute, inheritance, cardinality, and relationships) are necessary to understand O-O models. A *class* is a blueprint, or prototype, that defines the variables common to all objects of a certain kind. An *object* represents a specific instantiation of a class. The common metaphor is to consider that a class is a cookie cutter whereas an object is a cookie. A class is “abstract,” when it cannot be instantiated otherwise it is “concrete.” A class has *attributes* with particular data types (e.g., string, float, and date). The attributes of a class can be *inherited* by subclasses. Object attributes have cardinality, i.e., a constraint on their number. The cardinality is equal to 1 when the attributes have a required unique value. It is larger or equal to zero when the attributes are optional and may have multiple values. It is smaller or equal to 1 when the attributes are optional but cannot have multiple values. A *relationship* establishes a logical connection among objects.

Data models can be designed for specific applications using data modeling tools (Duineveld et. al., 2000). For instance, Protégé (Protégé, 2004; Gennari et. al, 2002) is a data modeling tool with a convenient graphical user interface (GUI). Protégé is open-source software, the capabilities of which can be enhanced by user-defined plug-ins. Some plug-ins export Protégé data models into different formats, including UML, OWL, XML Schema, and RDF. The graphical interface of Protégé enables description of particular objects using classes and relates them through various types of relationships. Figure 5.1 shows an example of the Protégé GUI, with classes and subclasses shown in the left window. The lower right window shows detailed attributes of these classes. Protégé allows design of entry forms to input data and query subsets of data.


Centrifuge Protégé 3.0 beta (file:\D:\Work\NEES\ProtegeModel\Centrifuge.pprj, OW...
File Edit Project OWL Wizards Code Window Help Prompt
OWLClasses Properties Forms Individuals Prompt Instances XML
SUBCLASS RE...
CLASS EDITOR
Asserted Inferred
Logic View
Properties View

FOR PROJECT: Centrif...

FOR CLASS: Organization (instance of owl:Class)

ASSERTED HIERARCHY:

- owl:Thing
  - Activity
    - Event
    - Project
    - Task
  - Configuration
  - Equipment
    - Sensor
  - File
    - EQMotion
  - Label
  - Organization
  - Person
  - Publication
  - Software
  - Specimen
  - Worker

Name: Organization

RDFS:COMMENT:

ANNOTATIONS:

Property	Value	Lang

ASSERTED CONDITIONS:

NECESSARY & SUFFICIENT

NECESSARY

owl:Thing ⊑
address ≤ 1 ⊑
description ≤ 1 ⊑
homePage ≤ 1 ⊑
is_composed_of ≥ 0 ⊑
is_workplace_of ≥ 0 ⊑
name = 1 ⊑
NEESCode ≤ 1 ⊑
phone ≥ 0 ⊑

PROPERTIES:

- address (single String)
- description (single String)
- homePage (single anyURI)
- is_composed_of (multiple Organization)
- is_workplace_of (multiple Worker)
- name (single String)
- NEESCode (single String)
- phone (multiple String)

DISJOINTS:

Figure 5.1 Protégé display of the metadata model

5.4. The metadata model

5.4.1. Premises of metadata model

The early data model (Bardet et. al, 2004; Peng and Law, 2004) was a pioneering and challenging effort in the absence of data standards in earthquake engineering. It expanded rapidly its classes (119) and class attributes (263) for capturing data from shaking tables and centrifuges. With the proliferation of new classes, the model documentation (i.e., data dictionary) demanded an increasing amount of effort to stay updated and consistent. Data entry into the model became rapidly a formidable task. The generalization of the data model from a few experiments equipments to other types of experiments (e.g., geophysical exploration, structures, and wave basins) turned into complicated models, likely to be rejected by the researchers and users community.

The present metadata model was designed to circumvent the shortcomings of early data models, and to offer a pragmatic solution for operating data repository and exchanging information in collaborative research projects. The current model does not focus on data details but on high-level metadata. Metadata have different meanings depending on the domains of expertise. Metadata are usually defined as data about data, an added layer of data which helps to understand data. Hereafter, metadata are defined as the data that explain how data were generated during experiments and computer simulations. Metadata glue logically together dispersed data fragments (e.g., video, photo, text, publication, and binary files). The metadata model is not a data model. It only contains the data necessary for: 1) understanding the processes of data generation; and 2) extracting subsets of data relevant to particular applications. In other words, the model documents (1) the information necessary to understand how data were generated and (2) the relations of numerous information pieces required to build query paths. These modeling assumptions form the basic premises of the metadata model.

Figure 5.2 illustrates the concepts of data and metadata using two layers. The bottom layer contains data stored in any types of files and archived on various file servers, e.g., Apache file servers and central data repository. Files are organized by a file system, i.e., hierarchy of drives and directories on personal computers. The top layer contains metadata about these files, i.e., additional information to that of the file system. This additional information is specific to domain experts (i.e., earthquake engineers) as described hereafter. The metadata glue together the various data files to produce a comprehensive picture of how they were generated from experiments and computer simulations. The metadata should contain sufficient information so that researchers can reproduce experimental and simulation data. Metadata describe the activity-dependent attributes of files, e.g., which processes produced the data files, who contributed to the data generation processes, when the processes took place, which equipment was used to generate data, and how the equipment was configured. As shown in Figure 5.2, the data files are identified uniquely in the metadata layer through universal resource identifier (URI), a concept endorsed by RDF.

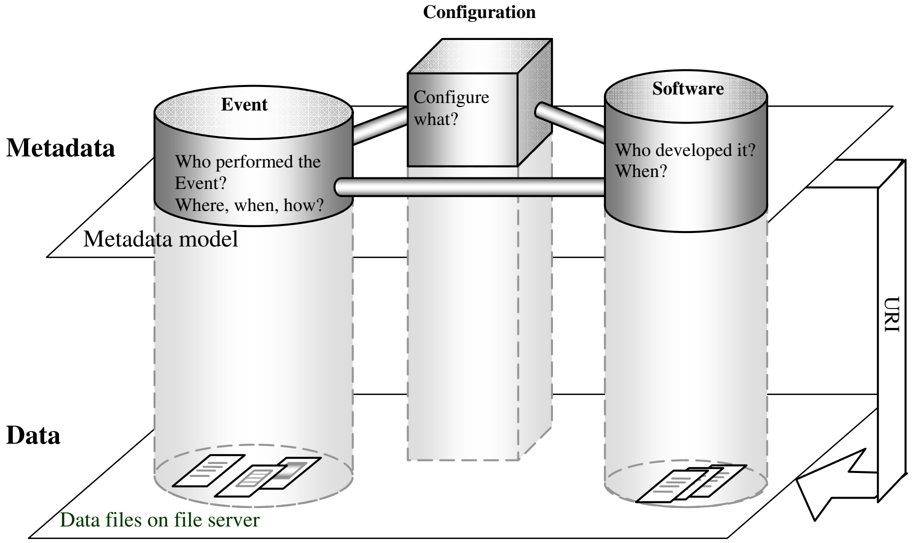


Figure 5.2 Relation between data and metadata

5.4.2. Classes

Table 5.1 summarizes the 16 classes of the metadata model. Organization; Person and Worker; Publication; Activity, Project, Task and Event; File and EQMotion; Configuration; Software; Equipment and Sensor; Specimen; and Label represent the main object types in experiments and computer simulations in geotechnical earthquake engineering. The class attributes have 6 data types, i.e., String, Date, Year, dateTime, Float, and URI, which are identical to those in XML schemas (http://www.w3.org/XML/Schema). Classes and class attributes are named according to the following convention. Class names start with an uppercase letter (e.g., Person) while attribute names start with a lowercase letter (e.g., name). Attribute names may contain uppercase letters for improving readability (e.g., firstName). The attributes relating objects (e.g., is_performed_at) have names with verbs instead of noun.

Table 5.1 The 16 classes of the metadata model

class	main definition	inherit from class
Organization	organization used to conduct research and business.	
Person	individual described independently of activity.	
Worker	person and his/her role in an activity or organization.	
Publication	technical document in print and work in progress.	
Activity	abstract class inherited by project, task and event.	
Project	root relating all objects through Tasks and Events.	activity
Task	group of Events as section of technical publication.	activity
Event	atomic level of activity describing what data were produced, when and where they were produced, who contributed, and how they were obtained.	activity
File	file stored on computers, including text, photo, video, and binary files.	
EQMotion	particular file containing a time series of earthquake motion.	file
Configuration	the state of a collection of equipment, sensor, software and specimen before a task and/or event.	
Software	system and application software used for computer simulation, analysis, data processing, and data input.	
Equipment	equipment and parts as in a manufacturer inventory, described independently of activity.	
Sensor	particular Equipment measuring Specimen responses with calibration information.	equipment
Label	temporary label used to identify various objects in figures.	
Specimen	any subject under investigation by experiments and computer simulations.	

5.4.3. Summary of relationships in metadata model

Figure 5.3 and Table 5.2 summarize the relations between the objects of the metadata model, which were presented in previous section. The metadata model systematically reuses objects, and avoids the repetition and possible conflicts of information. Project, Task and Event re-use extensively activity-independent objects (i.e., Person, Equipment, Sensor, Organization, Software, Specimen, File, and EQMotion) and complete them with new attributes that depend on activity. Project is the root of the metadata model. It is the only object that invoke directly or indirectly all other objects. As shown in Figure 5.3, the information from various objects flows one way into Project, which rules out ambiguous relations and inconsistent query results. When metadata are properly entered, any object should relate to at least one Project. Metadata are incomplete when an object cannot be related to any project. In that case, the unrelated object is an orphan unreachable through queries and disconnected from data sets.

Table 5.2 List of all relations between objects of the metadata model

class calling	attributes	point to instances of
Project	*is_sponsored_by*	Organization
Project	*is_granted_to*	Organization
Project/Task	*is_subdivided_in*	Task/Event
Project/Task/Event	*originated_from*	Project/Task/Event
Project/Task/Event	*is_performed_at*	Organization
Project/Task/Event	*is_performed_by*	Worker
Task/Event/Configuration	*is_configured_by*	Configuration
Event	*produces*	File/Software/Specimen/Equipment/ Publication/Configuration
Organization	*is_workplace_of*	worker
Organization	*is_composed_of*	Organization
Worker	*is_identified_as*	Person
Equipment	*is_made_up_of*	Equipment
Equipment	*is_manufactured_by*	Organization/Person
Equipment	*is_owned_by*	Organization/Person
Specimen	*is_instrumented_with*	Sensor/Label
Sensor	*is_calibrated_in*	File
Configuration	*is_prepared_for*	Label/Specimen/Equipment/Sensor/Software
Configuration/Specimen/Software/	*is_described_in*	File/Publication

Equipment/Project/Task/Event/ Label	*is_label_of*	Specimen/Equipment/Sensor/Software
Publication/Software	*is_authored_by*	Person/Organization

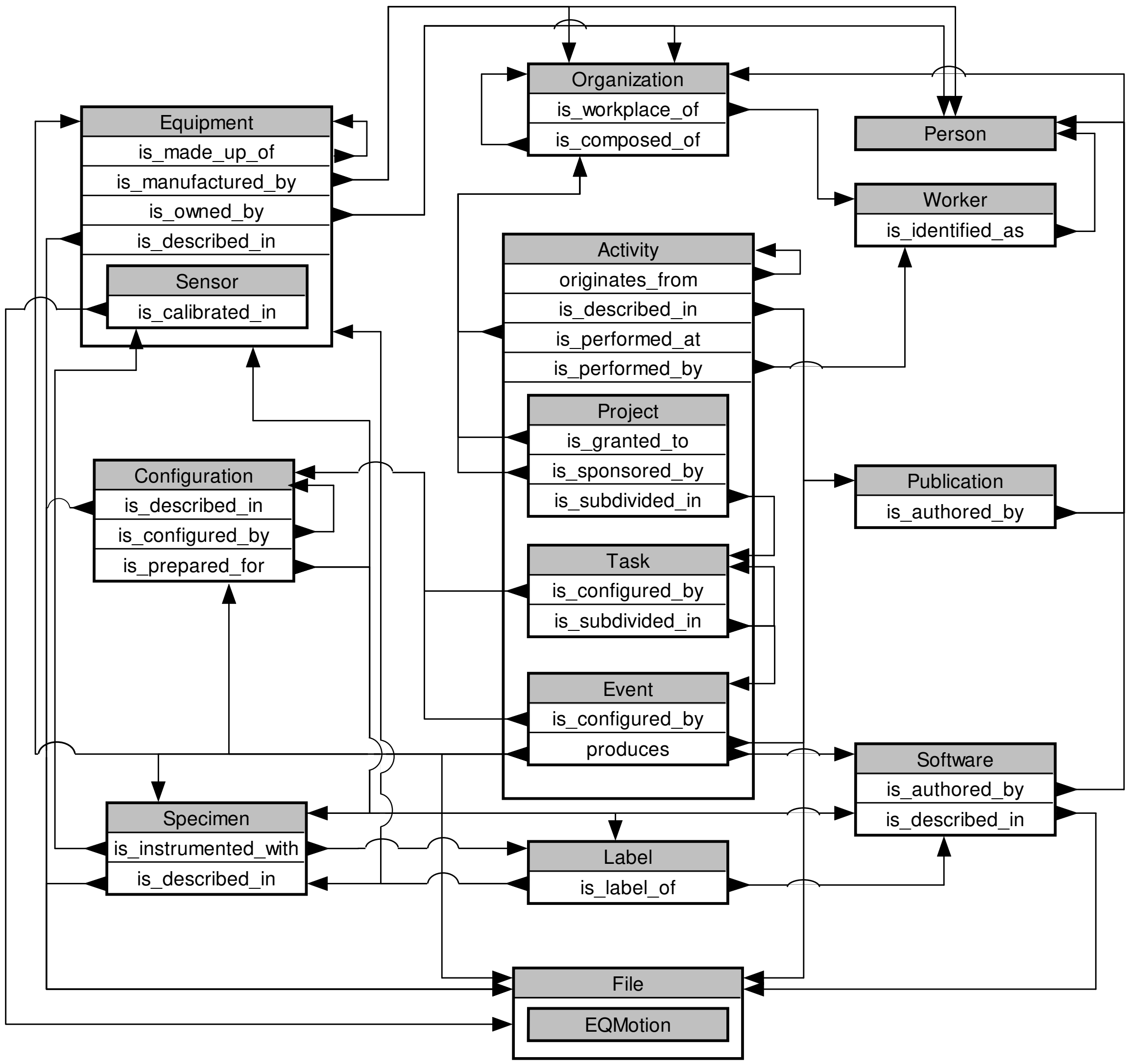


Figure 5.3 Graphical representation of all the relations in the metadata model

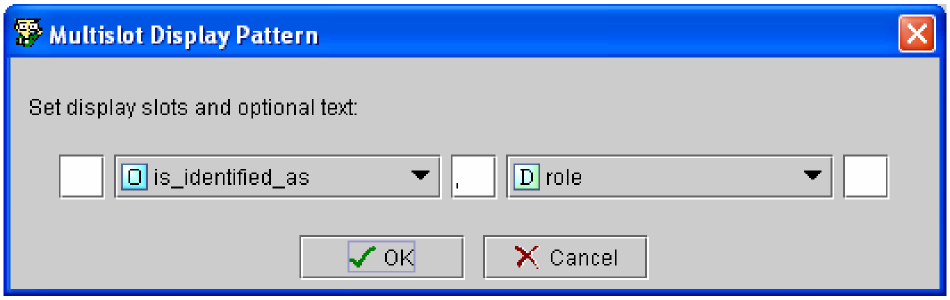
5.5. Applications of metadata model

A few practical considerations need to be mentioned before applying the metadata model, namely how to identify objects before relating objects and how to check the consistency of input metadata. Following these considerations, we summarize a few applications of the metadata model including: 1) documentation of data sets; 2) exchange of documented data sets; and 3) automatic generation of data reports. Detailed applications of the metadata model can be found in Bardet et al. 2004 b, c, and d.

5.5.1. Naming objects for building relationships

Most database programs (e.g. Protégé) identify objects by assigning them unique names, which are unfortunately inconvenient for constructing relations between objects. For instance, Protégé assigns internally to instances the name “filename_individual_i” where “filename” is the Protégé file name and *i* is the instance number internally defined by Protégé. Protégé also allows users to rename objects meaningfully for building relations between objects, provided that these new names are unique within a particular scope. For example, it is convenient to rename a Publication instance “filename_individual_103” and a Person instance “filename_individual_87” as “centrifuge_report” and “Kawata,” respectively, when one desires to indicate that “centrifuge_report” is authored by (is_authored_by) “Kawata.”

As shown in Figure 5.4, Protégé has a “Multislot Display Pattern” which renames objects automatically by combining several of their attribute values. For example, an instance of *Worker* can be conveniently named “Kutter, Bruce, principal investigator.” In this example, *is_identified_as* is named using the *lastName* and *firstName* of a Person ("Kutter, Bruce") and adding its *role* ("principal investigator"). This feature is useful for defining unique, yet meaningful names, and building relations between objects. This feature can also be used to sort, filter and select instances by their attributes before relating them. Table 5.3 lists a few useful combinations of attributes for naming different objects.



Set display slots and optional text:

is_identified_as , role

Figure 5.4 Multislot display using Proégé

Table 5.3 Naming objects from their attributes for building relations

object	slot 1	slot 2	slot 3
equipment	is_manufactured_by	partNumber	serialNumber
file	URI		
person	lastName	firstName	
worker	is_identified_as	Role	
publication	description		
label	name	is_label_of	
sensor	name	is_manufactured_by	sensorType
software	name	is_authored_by	

5.5.2. Metadata entry

In contrast to the reference data model (Bardet et. al, 2004a), the present metadata model is capable of tracing the interdependencies between all its objects. At the present, no complete series of test has been devised to verify that metadata are completely entered.

Four tests have however been implemented to check metadata in Protégé (Zand and Bardet, 2004). The first test consists of checking the data types of class attributes when they are entered into Protégé. This test is performed through entry slot widgets including a new ISO8601 slot widget for time and date. The second test performs the Protégé ontology tests, which checks for incorrect data types and cardinality constraints (e.g., insufficient or excessive number of data). A large number of object attributes are optional, which allows users to enter metadata with great flexibility. The third test verifies that all the File objects are related to at least another object. When a File cannot be related to any objects, its origin and usefulness become undetermined. The fourth test avoids circular references. Equipment, Sensor, Organization, Task, Configuration, and Label have attributes that points to objects of the same type, which is convenient to define a hierarchy of objects; however some hierarchies may inadvertently lead to circular references and infinite loops when instances are improperly defined. A new Protégé widget detects these circular references and displays error messages when it encounters them.

5.5.3. Documentation of data sets

The metadata model has been applied to a few particular projects for documenting data sets (Bardet et. al., 2004 b and c). For instance, Bardet et al., 2004 c, applied the metadata model to a complicated centrifuge data set, including hundreds of different objects and experimental steps. The metadata model is found capable of documenting experimental processes in centrifuge testing in much greater details than traditional data reports on websites. Figure 5.5 shows the distribution of all objects for a selected centrifuge test. In comparison with a traditional online data report posted by the Center for Geotechnical Modeling (CGM) at UC Davis, the metadata model accounts for many more objects.

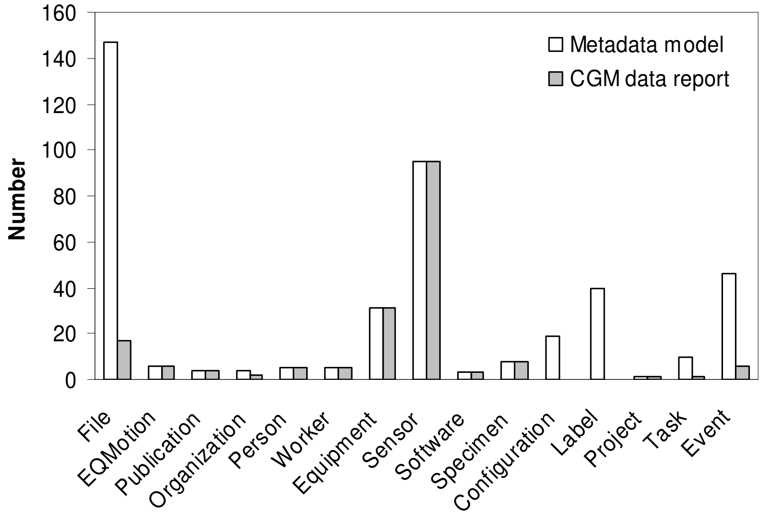


Figure 5.5 Number of objects used in metadata model and a traditional data report on websites

5.5.4. Exchange of data sets documented with XML metadata

The present metadata model can be represented using XML schemas as well. Figure 5.6 shows an example of XML graphical representation for the class Equipment. The XML schemas are made of 16 elements, which represent the 16 classes of the metadata model. Detailed definitions of all elements and element attributes can be found in Bardet et al., 2004 d. XML schemas are convenient to exchange metadata among researchers. When researchers distribute their data sets, they can document directories of data files (e.g., zip files) with a single XML metadata document.

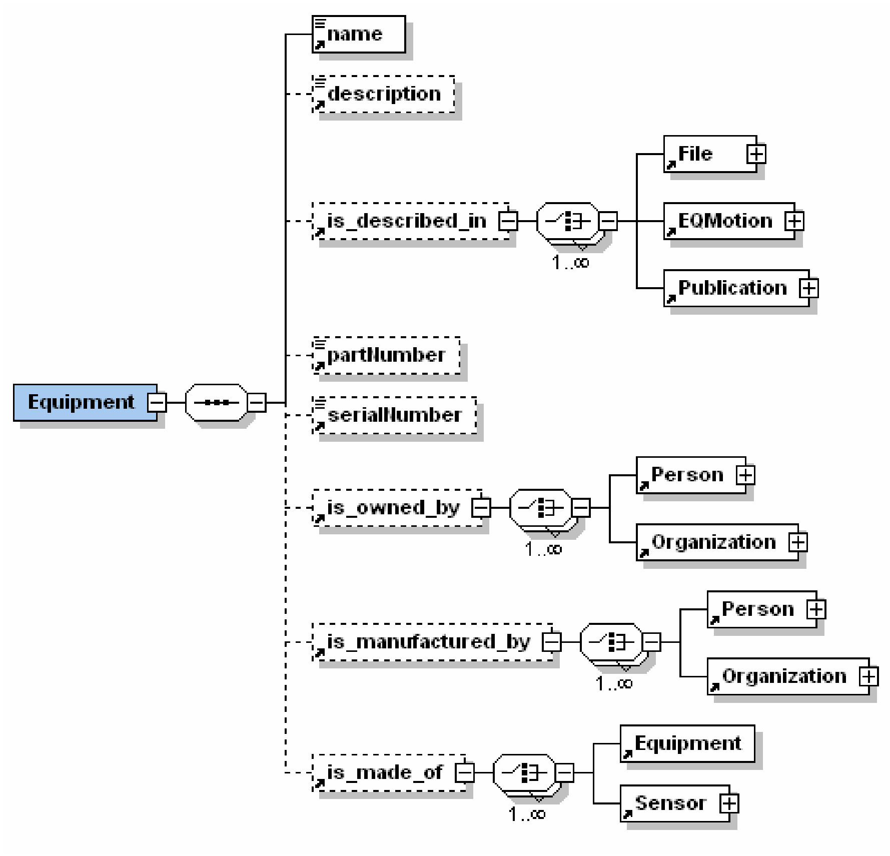
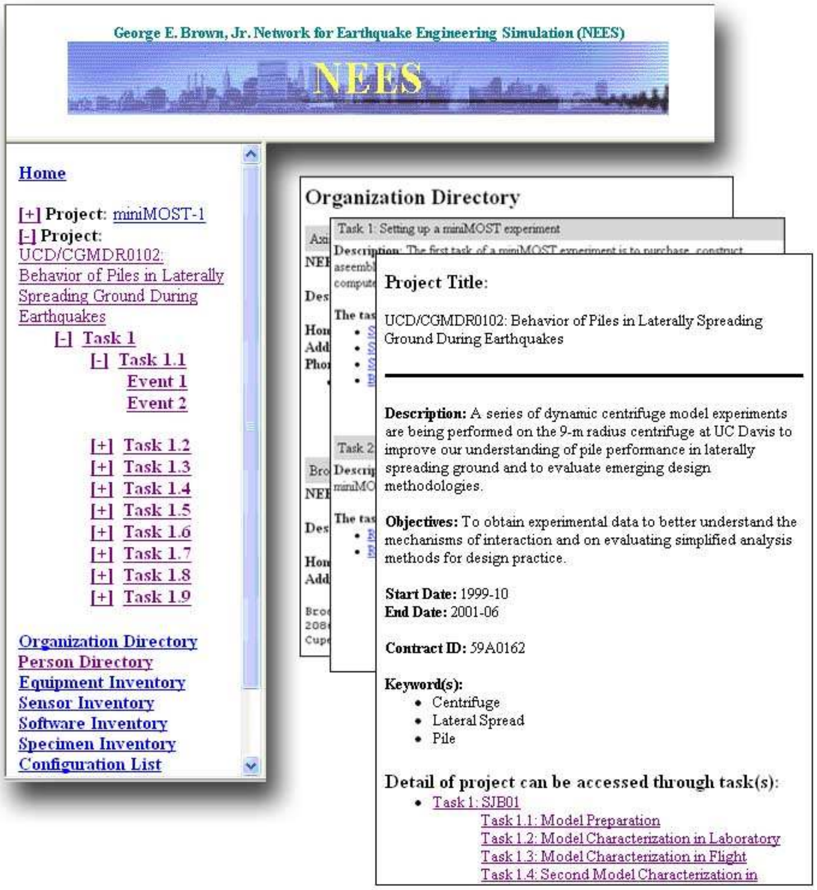


Figure 5.6 Diagram of XML element Equipment

5.5.5. Automatic generation of data reports

Bardet et al., 2004 d, illustrated the application of metadata to generate automatic data report for a particular experimental project. Figure 5.7 shows web pages of the detailed data report. The left frame displays the hierarchical structure of the complete project using a directory structure and points to the pages on Task, Event, inventories and directories. Inventory pages describe and bookmark all objects including Equipment, Sensor, Software, Specimen, Organization, Person, and Configuration. These objects are called by the Project, Task and Event pages through hyperlinks.



Organization Directory

Task 1: Setting up a miniMOST experiment

Project Title:

UCD/CGMDR0102: Behavior of Piles in Laterally Spreading Ground During Earthquakes

Description: A series of dynamic centrifuge model experiments are being performed on the 9-m radius centrifuge at UC Davis to improve our understanding of pile performance in laterally spreading ground and to evaluate emerging design methodologies.

Objectives: To obtain experimental data to better understand the mechanisms of interaction and on evaluating simplified analysis methods for design practice.

Start Date: 1999-10
End Date: 2001-06

Contract ID: 59A0162

Keyword(s):
- Centrifuge
- Lateral Spread
- Pile

Detail of project can be accessed through task(s):
- Task 1: SJB01
  Task 1.1: Model Preparation
  Task 1.2: Model Characterization in Laboratory
  Task 1.3: Model Characterization in Flight
  Task 1.4: Second Model Characterization in

Figure 5.7 Data report automatically generated by the metadata model

5.6. Discussion

The metadata model is an object-oriented model that borrows the concepts of URI from RDF for building relations between the data generated by experiments and simulations in geotechnical earthquake engineering. It re-uses information in efficient ways, describes relations using attributes and constrains relations only between selected classes. However it does not define inverse relations and does not impose the sophisticated relational constraints (e.g., symmetry and transitivity) implemented in OWL. More advanced relations could be the subject of future research.

Metadata models are not yet fully accepted in geotechnical earthquake engineering. Researchers are concerned with the additional work required to enter metadata on top of existing data. These concerns, which are legitimate at the present, may become lesser problems when the benefits of metadata models become more established through practical use, e.g., automatic creation of data report, and integration with electronic laboratory notebooks.

5.7. Summary

After defining an efficient data exchange format for the geotechnical community, the next step would be to preserve the understanding of how data sets are generated, specifically for not well-documented data. Therefore, metadata modeling approach is proposed for future work. As a guideline, metadata modeling for collaborative experimental research is studied.

Large projects in geotechnical earthquake engineering generate valuable data sets from experiments and computer simulations that need to be preserved in data repository and exchanged among many researchers. In order to facilitate the archival and exchange of information, a metadata model was proposed to document the data sets and to relate them to the persons, organizations, software, and equipment that contributed to data generation. The metadata model was conceived based on previous data models, which were unfortunately too complicated and could only be applied to a limited set of experiments. The present metadata model, although simpler than early data models, has more capabilities and applies to many kinds of experiments. The metadata model was developed in the object-oriented framework using Protégé. Its applications were illustrated with examples from centrifuge experiments. The metadata model can be used as a guideline to develop a similar metadata model for practical geotechnical engineering projects.

Chapter 6. Summary and Conclusions

6.1. Summary of Research

Geotechnical information is very broad and comprehensive and can contain anything from regular stratigraphy to finite element analysis element properties or GIS data. However, Chapter two of this dissertation showed that borehole data including in situ and laboratory tests are vital for almost any construction project. Borehole data have priority over other geotechnical data as all other data rest upon it. Although this information is obtained from rather costly drilling and laboratory operations, they are poorly documented and curated. This is mainly due to the lack of easily adoptable geotechnical community standards for data handling.

This study defined geotechnical community and how they create and use data. Previous geotechnical information release efforts were recognized and reviewed. A requirement list for data exchange format was developed based on the advantages and shortcomings of past efforts and the community specific needs.

This study analyzed AGS data format in more depth and concluded that the information contained in AGS can be modeled. Moreover, Chapter three reviewed existing conversion algorithms from relational to XML-based data organization and defined some principles for the transformation. The study then, proposed a preliminary model based on automatic conversion that was modified to present the hierarchical nature of AGS data with nested structure. It was concluded that this structure is not appropriate for exchange of data subsets. Consequently, another structure was proposed. This new structure preserves the relationships between objects by unique identifiers named key and keyref in XML schema. Therefore, it is selected for second version of XAGS. Different components of geotechnical data were illustrated in XASG 2.0 format with diagrams and XAGS data file examples. Chapter three also, addressed some of the deliberations for XAGS 2.0 decisions.

Chapter four demonstrated how borehole data in XAGS format can be viewed and tabulated with HTML on an Internet browser and the location of borehole can be displayed in interactive maps. Conversion to and from XAGS format were also addressed. With the help of XML editors and spreadsheets, XAGS data generation and modification were illustrated. Moreover, data validation properties of XAGS for different scenarios of structural inconsistencies, and corresponding error messages were shown. Data exchange between team and team members for data sets and data subsets were discussed. The chapter demonstrated archive and distribution of XAGS data over the web. Finally, XAGS appropriateness as a data exchange format was evaluated by comparing its capabilities with the requirement list developed early in the study. Chapter four concluded that even though XAGS is not yet satisfying all the requirements, it is a major leap for the geotechnical community from the past efforts.

In future as data exchange gets more comprehensive and more of less common procedures will be included, all data will not be covered by standards. Hence, additional data will be needed to understand how data is obtained. Chapter five proposed metadata modeling as an added layer over data exchange for future work. Metadata modeling will document the data needed to understand how data results are obtained. As a guideline a metadata model for collaborative geotechnical earthquake engineering experiments was introduced.

6.2. Major Contributions

The major contributions of this research consist of six parts.

As its first contribution, this dissertation described geotechnical community. It analyzed geotechnical data and identified the most critical data among geotechnical information. It analyzed the way data is used within the community and defined what data exchange is.

The second contribution is development of a requirement set for geotechnical community specific data exchange format. This set was created by analyzing some of the more recent efforts for developing information release standards such as AGS, GML and DIGGS. This requirement set can be used to evaluate almost all data exchange formats developed for different areas of geotechnical engineering.

The third contribution is to demonstarte systematic development of a data exchange format for geotechnical community. Already existing data modeling concepts in information technology arena are applied on geotechnical engineering information. Some of these ideas are implemented, for the first time, in the geotechnical community. This study analyzed a geotechnical information release system with the help of conceptual data modeling. It was concluded that the information system is a relational information system. Physical relationships between geotechnical information objects were defined. Eventually, conversion algorithms from information technology were introduced to transform relational data to XML format.

The fourth accomplishment is development of XAGS. XAGS is a data exchange format that can be easily adopted and used by geotechnical community for efficient data handling among team members, teams and over the web. This format can be integrated into the common and popular procedures of geotechnical community without use of special commercial software programs or extensive trainings. XAGS is utilizing the already established and well defined AGS data dictionary. XAGS is created based on XML which is proven to be the best data exchange format and could be processed by most major software programs. Moreover, an information system is created to archive and disseminate XAGS via web.

The fifth contribution of this research is documentation of implementation of XAGS format. This dissertation describes the procedures for implementing this exchange format for the most common in situ and laboratory tests in geotechnical projects. This text documents how XAGS data can be easily viewed, generated, and modified with one of the already available free or commercial XML editors or spreadsheets. It also shows how to exchange data between involved parties. Additionally, XAGS was validated by illustrating its ease of use in archival, retrieval, distribution and graphical display of geotechnical information between teams and team members as well as via World Wide Web. It was also shown that legacy data can be converted to XAGS and XAGS data can be converted to other defined formats such as AGS with the help of XSL Transformations.

As its sixth contribution, this research introduced metadata modeling as an additional layer for documenting geotechnical information. As the data exchange format becomes more comprehensive all data will not be covered by standards. Therefore, an additional layer of data will be needed to understand how data is obtained. The development of a metadata model for collaborative experimental research in geotechnical earthquake engineering was presented for future development of a metadata model for practical geotechnical engineering projects.

6.3. Possible Future Considerations for XAGS

Here are some of the areas that can be studied further or the future advancements in other areas such as information technologies might make them possible to consider for XAGS format:

- XAGS is currently based on AGS data dictionary that is mainly conforming to British Standards (BS). If an ASTM conforming format is developed, it will certainly make the adaption of the exchange format in United States and other countries following ASTM standard considerably more successful and relevant.

- At the current state, XAGS is implemented for the common in situ and laboratory tests. XAGS can be extended to serve other tests and events such as geophysical tests, dynamic laboratory tests or geoenvironmental events.

- Currently, KML format can not fully utilize all the capabilities available to XML-based data. If KML format becomes able to fully utilize all the characteristics of XML in the future, it is strongly recommended to develop a KML compatible version of XAGS. This will simplify the spatial presentation of data with interactive maps.

- It should be noted that XASG 2.0 implementation was based on the current code limitations, such as XSLT and web browser capabilities. In creating the most desirable format, software program limitations played a confining role. As the newer versions of codes are introduced and the limitations are lifted, XAGS can also be updated to a newer version.

- Since XAGS is an XML-based format. It has the potential to be used for web services to transfer the data between databases and client sides without the user intervention.

- Even though XAGS meets almost all the current needs of geotechnical engineers, a data exchange format is a work in progress. With the new codes, programs, needs, tests, and procedures data format might need to be updated over time. For the next versions of XAGS expansion and revision, community input needs to play a more important role. Approaching geotechnical community to survey more specific requirements for data exchange format is one the means to acquire the community input and feedback.

- In addition, by increase of software programs capabilities in newer versions, XAGS, as an XML based format, could be utilized in new creative ways. This will open totally new horizons for this format. More advanced presentation with commercial programs or XAGS tailored specific programs are possible.

- If GML format is simplified in the future and is picked up by a considerable number of programs and achieves acceptance by data users, the idea of having a GML-compliant format might be worth exploring. GML compatibility of XML language will provide background for future GIS based applications.

References

Afnor, 1991, Soils: investigation and testing – Standard Penetration Test, NF P 94-116, Oct.

AGS, 2004. Electronic transfer of geotechnical and geoenvironmental data, Edition 3.1. Association of Geotechnical and Geoenvironmental Specialists, Kent, United Kingdom, 150 pp.

AGS, 2005. Electronic transfer of geotechnical and geoenvironmental data using XML data format. Association of Geotechnical and Geoenvironmental Specialists, Kent, United Kingdom, 23 pp.

AGSML, 2005. AGSML – The combined power of AGS and XML. http://ags.org.uk/agsml, July 2010.

The American Heritage Dictionary of the English Language, 2000. 4th ed. Orlando: Houghton Mifflin Company.

Arlow, J., Neustadt, I., 2001. UML and the Unified Process: practical object-oriented analysis and design. Addison-Wesley Professional (December 31, 2001), 416 pp.

ASTM, 2005a. Annual book of ASTM Standards, Soil and Rock (I): D420 – D5611, Vol. 04.08, American Society for Testing and Materials. Philadelphia, PA, 1429 pp.

ASTM, 2005b. Annual book of ASTM Standards, Soil and Rock (II): D5714 - Latest, Vol. 04.09, American Society for Testing and Materials. Philadelphia, PA, 1610 pp.

Atzeni, P., Batini, C., De Antonellis, V., 1993. Relational Database Theory: A Comprehensive Introduction. Addison Wesley Longman, Jan 10, 1993, 350 pp.

Baise, L. G., Brankman, C.M., 2004. “Liquefaction Hazard Mapping in Boston, Massachusetts: Collaborative Research with William Lettis & Associates, Inc., and Tufts University.” Award No. 02HQGR0036 and 02HQGR0040, National Earthquake Hazard Reduction Program, U.S. Geological Survey, United States, 2004

Bardet, J. P.,1997. Experimental Soil Mechanics. Prentice-Hall, Upper Saddle River, New Jersey, p. 583.

Bardet, J. P., Hu, J., Tobita, T., and Mace, N., 1999. Database of case histories on liquefaction-induced ground deformation. Civil Engineering Department, University of Southern California, Los Angeles.

Bardet, J.P., Peng, J., Law, K., and Swift, J., 2004a. "Overview of the NEES data/metadata model." In: Proceedings of 13th World Conference in Earthquake Engineering, Vancouver, Canada, paper 4001.

Bardet, J. P, Liu, F., Mokaram, N., 2004 b. "Application of the NEES metadata model to the miniMOST experiment." Report, Civil Engineering Department, University of Southern California, Los Angeles.

Bardet, J.P., Liu, F., Mokarram, N., Brandenberg, S., Kutter, B., and Wilson, D. 2004c. Application of the NEES metadata model to centrifuge modeling. Report of Civil Engineering Dept, University of Southern California, Los Angeles.

Bardet, J. P., Liu, F., Mokaram, N., 2004 d. "Report and exchange of NEES data sets using metadata model." Report, Civil Engineering Department, University of Southern California, Los Angeles.

Bardet, J.P., Liu, F., and Mokarram, N., 2005. "Data and metadata modeling for the George E. Brown Jr. Network for Earthquake Engineering Simulation." In: EURODYN 2005 Proceedings of the 6th International Conference on Structural Dynamics, Paris, France, Sep. 4-7 2005. Millpress, Rotterdam.

Bardet, J. P., Zand, A. G., 2009. "Spatial Modeling of Geotechnical Information Using GML." Transaction is GIS, 2009, 13(1): 125-165.

Bardet, J.P., Liu, F., and Mokarram, N., 2010. A metadata model for collaborative experiments and simulations in earthquake engineering. Frontiers of Architecture and Civil Engineering in China, Higher Education Press, co-published with Springer-Verlag, Volume 4, Number 2, June, 2010, 133-153pp.

Benoît, J., Lutenegger, A.J. (Eds.), 2000. National Geotechnical Experimentation Sites, ASCE, Co-published with the GeoInstitute, Reston, VA, 397 pp.

Booch, G., 1993. "Object-Oriented Analysis and Design with Applications." 2nd ed. Boston: Addison-Wesley.

Botts, M., Percivall, G., Reed, C., Davidson, J., 2008. "OGC® sensor Web enablement: overview and high level architecture." GeoSensor Networks. Second International Conference, GSN 2006. Revised Selected and Invited Papers, p 175-90.

Boumphrey, F., and Tittel, E., 2000, XML for Dummies – 2nd Edition, IDG Books Worlwide.

Bray, C.J, Chandler, R.J., Walthall, S., Hoit, M. and Lefchik, T. 2008. “Extending the Geotechnical Dictionary: Best Practice for Customizing the International Framework for Geotechnical Data.” Proc. GeoCongress 2008 (eds. A.N. Alshawabkeh, K.R. Reddy & M.V. Khire), Geotechnical Special Publication No. 179, Reston: American Society of Civil Engineers, pp. 549-556.

Chandler, R.J., Quinn P.M., Beaumont A.J., Evans D.J. and Toll D.G. , 2006. “Combining the power of AGS and XML: AGSML the data format for the future,” in GeoCongress 2006: Geotechnical Engineering in the Information Technology Age, Atlanta, USA, Reston: ASCE, pp. 112-117.

Chandler, R.J., 2009. Feasibility Study into Implementing DIGGS in the UK. presented in DIGGS meeting, March 2009, Orlando, Florida.

Codd, E.F., 1990. The Relational Model for Database Management (Version 2 ed.). Addison Wesley Publishing Company

Coronna, S., 2009. “gINT Software and DIGGS ,” presented in DIGGS meeting, March 2009, Orlando, Florida.

COSMOS, 2003. “COSMOS geotechnical data model, version 0.5.” Civil Engineering Department, University of Southern California, Los Angeles, California. http://geodata.usc.edu:1100/geotech_schemas/SatelliteSchema.html

COSMOS, 2010. The Consortium of Organizations for Strong-Motion Observation Systems, a pilot XML-based exchange standard for Geotechnical Virtual Data Center. http://www.cosmos-eq.org/, July 2010

Cox, S., 2007. Observations and Measurements – Part 1 - Observation schema, Open Geopspatial Consortium Inc. Reference, 2007-12-08 , OGC 07-022r1 Version: 1.0, OpenGIS® Implementation Standard.

Craig, R. F., 2001. Soil Mechanics. Sixth edition, Spon Press, London, Great Britain, 485 pp.

CUAHSI HIS, 2010. CUAHSI Hydrologic Information System (HIS). http://his.cuahsi.org/, July 2010.

CUREE, 2002. “CUREE-caltech wood frame project. Consortium of Universities for Research in Earthquake Engineering (CUREE).” http://www.curee.org/projects/woodframe/

Das, B., 1994. Principles of Geotechnical Engineering. Third edition, PWS Publishing Company, USA, 672 pp.

Deaton, S.L., 2009. “Dataforensics DIGGS Review,” presented in DIGGS meeting, March 2009, Orlando, Florida.

DIGGS, 2009. Invitational Meeting – Report on Project Status and Development of a New Roadmap. March 25-26, 2009, Orlando, Florida.

DIGGSML, 2010. Data Interchange for Geotechnical and GeoEnvironmental Specialists, http://www.diggsml.org, July 2010.

Duineveld, A. J., Stoter, R., Weiden, M. R., 2000. “Kenepa B, Benjamins V R. WonderTools? A comparative study of ontological engineering tools.” International Journal of Human-Computer Studies, 2000, 52(6): 1111—1133.

Elmasri, R., and Navathe, S. B., 2003. Fundamentals of Database Systems, 4th edition. Pearson Education, Inc., Boston, MA, 1030 pp.

Gamma, E., Helm, R., Johnson, R., Vlissides, J., 1995. Design Patterns: Elements of Reusable Object Oriented Software. Boston: Addison-Wesley.

Gennari, J., Musen, M. A., Fergerson, R. W., Grosso, W. E., Crubézy, M., Eriksson, H., Noy, N. F., Tu, S. W., 2002. “The Evolution of Protégé: An environment for knowledge-based systems development.” Technical Report SMI-2002-0943.

GGSD, 2010, Geotechnical and Geoenvironmental Software Directory, http://www.ggsd.com

Gibson, R., and Erle, S., (2006). Google Maps Hacks. O’Reilly.

GML, 2004. Geography Markup Language, http://www.opengis.net/gml/, July 2010.

Google Earth, 2010. http://earth.google.com/. July 2010.

HTML, 2010. HTML Working Group. http://www.w3.org/html/wg/, July 2010.

Hu, J. (2003). "Modeling of ground deformation using spatial analysis and elasto-plastic finite strain theory," PhD, University of Southern California, Los Angeles, CA.

IEEE 754-1985, 2006, IEEE Standard for Binary Floating-Point Arithmetic. http://standards.ieee.org/reading/ieee/std_public/description/busarch/754-1985_desc.html, July 2010.

ISO 8879, 1986. Overview of SGML Resources. http://www.w3.org/MarkUp/SGML/, July 2010.

Jacobsen, I., 1994. “Object-Oriented Software Engineering: A Use Case-Driven Approach.” Boston: Addison-Wesley.

Jennings, R., 2007. Special Edition Using Microsoft Office Access 2007. Que, 1488 pp.

Kapuskar, M., 1993, Seismic Site Response Analysis of the Marina District, San Francisco, University of Southern California, Department of Civil Engineering.

KML, 2010. Google code: What is KML? http://code.google.com/apis/kml/, July 2010.

Kramer, S. L., 1996, Geotechnical Earthquake Engineering, Prentice Hall.

Lee, D., Mani, M., Chiu, F. and Chu, W.W., 2001. Nesting-based relational-to-XML schema translation,” In Proc. of the WebDB, pages 61–66.

Lee, D., Mani, M., and Chu, W.W., in press. Schema conversion methods between XML and relational models.

Lee, Y., 2006. Wireless Instrumentation and Metadata for Geotechnical Centrifuges. PhD, Universoty of Southern California.

Liu, C., Vincent, M., Liu, J., and Guo, M., 2003. “A virtual XML database engine for relational databases.” In: Proceedings of XSYM, pp. 37–51.

Liu, C., Vincent, M. W. and Liu, J., 2006. “Constraint preserving transformation from relational schema to XML schema.” World Wide Web Journal, vol. 9, no. 1, pp. 93-110.

Liu, F. (2008). "Data discovery on liquefaction-induced lateral ground deformations," PhD, University of Southern California, Los Angeles, CA.

massDOT, 2010. The Central Artery/Tunnel Project - The Big Dig, http://www.massdot.state.ma.us/Highway/bigdig/bigdigmain.aspx, July 2010.

Meyer, B., 1997. “Object-Oriented Software Construction.” 2nd ed. Englewood Cliffs: Prentice Hall.

NEES, 2010. George E. Brown, Jr. Network for Earthquake Engineering Simulation. http://www.nees.org/, July 2010.

NGES, 2000. National Geotechnical Experimentation Sites. http://www.unh.edu/nges/, July 2010.

OGC (Open Geospatial Consortium), 2008. Geography Markup Language, Encoding Standards. WWW document, http://www.opengeospatial.org/standards/gml

Patterson, D., 2009. DIGGS Implementation efforts –UK Transportation Agency Perspective, Highways Agency, UK, presented at: DIGGS Invitational Meeting – Orlando, Florida, March 25-26, 2009.

Peng, J., and Law, K., 2004. Reference NEESgrid data model. Technical Report TR-2004-40, NCSA, University of Illinois at Urbana-Champaign, Illinois.

Protégé, 2004. "The Protégé project." Stanford University. http://protege.stanford.edu

QuakeML, 2010, QuakeML, https://quake.ethz.ch/quakeml, July 2010.

Road Traffic, 2010. Industry Projects: Big Dig, Central Artery / Tunnel Project Specifications. http://www.roadtraffic-technology.com/projects/big_dig/specs.html, July 2010.

ROSRINE, 2008. Resolution of Site Response Issues from the Northridge Earthquake. http://gees.usc.edu/ROSRINE, July 2010.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W., 1991. "Object-Oriented Modeling and Design." Englewood Cliffs: Prentice Hall.

Saake, G., Conrad, S., Schmitt, I. and Turker, C., 1995. Object-oriented database design: what is the difference with relational database design? Proceedings of Object World Frankfurt '95, Frankfurt, Germany, 1995.

SCEC, 2010. Southern Californica Earthquake Center, http://www.scec.org/, July 2010.

Schutzberg, A., 2005. "The Technology Behind Google Maps." Directions Magazine, http://www.directionsmag.com/article.php?article_id=760&trv=1, July 2010.

Styler, M. Hoit, M. and McVay, M., 2007. Deep Foundation Data Capabilities of the Data Interchange for Geotechnical and Geoenvironmental Specialists (DIGGS) Mark-up Language. http://www.ejge.com/2007/JTC2_Volume/JTC2_Volume_TOC.htm .

Swift, J., Peng, J., Bardet, J.P., Liu, F., Mokarram, N., and Pekcan, G.,2004. Using the NEES reference data model and the NEES metadata browser for centrifuge experiments, Version 1.1. Report of Civil Engineering Dept, University of Southern California, Los Angeles.

Toll, D. G., Shields, R., 2003. A web-based Data Format for Ground Investigation Data. Electronic Journal of Geotechnical Engineering, http://www.ejge.com/2003/JourTOC8D.htm.

Toll, D. G., Oliver, A. J., 1995. Structuring Soil and Rock Descriptions for Storage in Geotechnical Databases. Geological Data Management (ed. J. R. A. Giles), Geol. Soc. Special Publ. No. 97, Bath: Geological Society Publishing House, pp 65-71.

Tsuboi, S., Morino, S., 2003. “Conversion of SEED format to XML representation for a new standard of seismic waveform exchange,” 3rd International Workshop on Scientific Use of Submarine Cables and Related Technologies (Cat. No.03EX660), p 285-7.

USGS. (1996). "USGS Response to an Urban Earthquake:Northridge '94." Open-File Report 96-263.

USGS, 2010. U.S. Geological Survey, http://www.usgs.gov/, July 2010.

VELACS, 1997. “Centrifuge and laboratory data sets of the NSF collaborative project verification of liquefaction analysis using centrifuge studies.” University of Southern California, Civil Engineering Department, Los Angeles, California. http://gees.usc.edu/velacs

Vlist, E., 2002. XML Schema. O'Reilly Media; 1 edition (June 15, 2002), 400 pp.

Wang, H., Azuaje, F., Clifford, G., Jung, B., Black, N., 2004. “Methods and tools for generating and managing ecgML-based information,” Computers in Cardiology 2004 (IEEE Cat. No.04CH37641), p 573-6.

Weaver, S.D., Lefchik, T., Hoit, M. and Beach, K. 2008. Geoenvironmental and Geotechnical Data Exchange: Setting the Standard. Proc. GeoCongress 2008 (eds. A.N. Alshawabkeh, K.R. Reddy & M.V. Khire), Geotechnical Special Publication No. 179, Reston: American Society of Civil Engineers, pp. 557-564.

Wusteman, J., and O'hlceadha, P. (2006). "Using AJAX to empower dynamic searching." Information Technology and Libraries, 25(2), 57-64.

XML, 2003. Extensible Markup Languages (XML), http://www.w3.org/XML/, July 2010.

XML Notepad, 2007. XML Notepad. http://xmlnotepad.codeplex.com/, July 2010.

XML schema, 2000, W3C XML Schema, http://www.w3.org/XML/Schema, July 2010.

XML-SEED. http://www.jamstec.go.jp/pacific21/xmlninja/, July 2010

XMLSPY, 2010, Altova XML Editor. http://www.altova.com/xml-editor/, July 2010.

XSL, 2009. The Extensible Stylesheet Language Family (XSL). http://www.w3.org/Style/XSL/, July 2010.

Yank, K., 2001, "Get XSL To Do Your Dirty Work". XML, XSLT & web service tutorial online. http://www.sitepoint.com/article/get-xsl-dirty-work, July 2010.

Yourdon, E., 2006. "Just Enough Structured Analysis," available in wiki format at: http://yourdon.com/strucanalysis/wiki/index.php?title=Introduction.

Zand, A. G., Bardet, J. P., 2004. "Protégé plug-in extensions for NEES metadata model." Report, Civil Engineering Department, University of Southern California, Los Angeles.

Zand, A., 2005. "Application of Spatial Databases and web services in geotechnical earthquake engineering." PhD Dissertation Proposal, Department of civil engineering university of southern California, August 2005, 44pp.

Zheng, H., Wanga, H.Y., Black, N.D., Winder, R.J., 2010. "Medical Informatics: Data structures, coding and classification," Technology and Health Care 18 (2010) 71–87, DOI 10.3233/THC-2010-0568

Zimmermann, R., Ku, W.S., Wang, H., Zand, A., and Bardet, J.P., 2006. "A Distributed Geotechnical Information Management and Exchange Architecture." Internet Computing, IEEE, Vol. 10, Issue 5, 26-33.

Appendix A XAGS 2.0 Schema Files

XAGS2.xsd:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns="http://gees.usc.edu/XAGS" xmlns:xags="http://gees.usc.edu/XAGS"
targetNamespace="http://gees.usc.edu/XAGS" elementFormDefault="qualified" xml:lang="English">
	<xs:annotation>
		<xs:documentation>
	This is the schema for XAGS version 2.0 containing Project, Hole and Sample and including ISPT, STCN and
LabTests.
		</xs:documentation>
	</xs:annotation>
	<!--========================Imports and Includes=========================-->
	<xs:import namespace="http://www.w3.org/1999/xlink" schemaLocation="xlink.xsd"/>
	<xs:include schemaLocation="ISPT2.xsd"/>
	<xs:include schemaLocation="STCN2.xsd"/>
	<xs:include schemaLocation="LABTests2.xsd"/>
	<!--==========================Elements===============================-->
	<xs:element name="XAGS">
		<xs:complexType>
			<xs:sequence>
				<xs:element ref="PROJ" minOccurs="0"/>
				<xs:element ref="HOLE" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="ISPT" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="STCN" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="SAMP" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="CBRG" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="CBRT" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="CHLK" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="CLSS" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="CMPG" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="CMPT" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="CNMT" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="CONG" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="CONS" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="FRST" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="GRAD" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="MCVG" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="MCVT" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="PTST" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="RELD" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="ROCK" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="SHBG" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="SHBT" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="SUCT" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="TNPC" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="TRIG" minOccurs="0" maxOccurs="unbounded"/>
				<xs:element ref="TRIX" minOccurs="0" maxOccurs="unbounded"/>
			</xs:sequence>
		</xs:complexType>
		<!--====Definition of Keys and Keyrefs====-->
		<xs:key name="ProjID">
			<xs:selector xpath="./xags:PROJ"/>
			<xs:field xpath="@PROJ_ID"/>
		</xs:key>
```

```
    <xs:keyref name="ProjIDRef" refer="ProjID">
        <xs:selector xpath="./xags:HOLE"/>
        <xs:field xpath="@PROJ_IDREF"/>
    </xs:keyref>
    <xs:key name="HoleID">
        <xs:selector xpath="./xags:HOLE"/>
        <xs:field xpath="@HOLE_ID"/>
    </xs:key>
    <xs:keyref name="HoleIDRef" refer="HoleID">
        <xs:selector xpath="./xags:ISPT|./xags:STCN|./xags:SAMP"/>
        <xs:field xpath="@HOLE_IDREF"/>
    </xs:keyref>
    <xs:key name="SptID">
        <xs:selector xpath="./xags:ISPT"/>
        <xs:field xpath="@ISPT_ID"/>
    </xs:key>
    <xs:key name="CptID">
        <xs:selector xpath="./xags:STCN"/>
        <xs:field xpath="@STCN_ID"/>
    </xs:key>
    <xs:key name="SampID">
        <xs:selector xpath="./xags:SAMP"/>
        <xs:field xpath="@SAMP_ID"/>
    </xs:key>
    <xs:key name="CbrgID">
        <xs:selector xpath="./xags:CBRG"/>
        <xs:field xpath="@CBRG_ID"/>
    </xs:key>
    <xs:keyref name="SampIDRef" refer="SampID">
        <xs:selector
xpath="./xags:CBRG|./xags:CHLK|./xags:CLSS|./xags:CMPG|./xags:CNMT|./xags:CONG|./xags:FRST|./xags:GRAD|./xag
s:MCVG|./xags:FRST|./xags:PTST|./xags:RELD|./xags:ROCK|./xags:SHBG|./xags:SUCT|./xags:TNPC|./xags:TRIG"/>
        <xs:field xpath="@SAMP_IDREF"/>
    </xs:keyref>
    <xs:key name="CbrtID">
        <xs:selector xpath="./xags:CBRT"/>
        <xs:field xpath="@CBRT_ID"/>
    </xs:key>
    <xs:keyref name="CbrgIDRef" refer="CbrgID">
        <xs:selector xpath="./xags:CBRT"/>
        <xs:field xpath="@CBRG_IDREF"/>
    </xs:keyref>
    <xs:key name="ChlkID">
        <xs:selector xpath="./xags:CHLK"/>
        <xs:field xpath="@CHLK_ID"/>
    </xs:key>
    <xs:key name="ClssID">
        <xs:selector xpath="./xags:CLSS"/>
        <xs:field xpath="@CLSS_ID"/>
    </xs:key>
    <xs:key name="CmpgID">
        <xs:selector xpath="./xags:CMPG"/>
        <xs:field xpath="@CMPG_ID"/>
    </xs:key>
    <xs:key name="CmptID">
        <xs:selector xpath="./xags:CMPT"/>
        <xs:field xpath="@CMPT_ID"/>
    </xs:key>
    <xs:keyref name="CmpgIDRef" refer="CmpgID">
        <xs:selector xpath="./xags:CMPT"/>
        <xs:field xpath="@CMPG_IDREF"/>
    </xs:keyref>
    <xs:key name="CnmtID">
        <xs:selector xpath="./xags:CNMT"/>
        <xs:field xpath="@CNMT_ID"/>
```

```
</xs:key>
<xs:key name="CongID">
        <xs:selector xpath="./xags:CONG"/>
        <xs:field xpath="@CONG_ID"/>
</xs:key>
<xs:key name="ConsID">
        <xs:selector xpath="./xags:CONS"/>
        <xs:field xpath="@CONS_ID"/>
</xs:key>
<xs:keyref name="CongIDRef" refer="CongID">
        <xs:selector xpath="./xags:CONS"/>
        <xs:field xpath="@CONG_IDREF"/>
</xs:keyref>
<xs:key name="GradID">
        <xs:selector xpath="./xags:GRAD"/>
        <xs:field xpath="@GRAD_ID"/>
</xs:key>
<xs:key name="McvgID">
        <xs:selector xpath="./xags:MCVG"/>
        <xs:field xpath="@MCVG_ID"/>
</xs:key>
<xs:key name="McvtID">
        <xs:selector xpath="./xags:MCVT"/>
        <xs:field xpath="@MCVT_ID"/>
</xs:key>
<xs:keyref name="McvgIDRef" refer="McvgID">
        <xs:selector xpath="./xags:MCVT"/>
        <xs:field xpath="@MCVG_IDREF"/>
</xs:keyref>
<xs:key name="FrstID">
        <xs:selector xpath="./xags:FRST"/>
        <xs:field xpath="@FRST_ID"/>
</xs:key>
<xs:key name="PtstID">
        <xs:selector xpath="./xags:PTST"/>
        <xs:field xpath="@PTST_ID"/>
</xs:key>
<xs:key name="ReldID">
        <xs:selector xpath="./xags:RELD"/>
        <xs:field xpath="@RELD_ID"/>
</xs:key>
<xs:key name="RockID">
        <xs:selector xpath="./xags:ROCK"/>
        <xs:field xpath="@ROCK_ID"/>
</xs:key>
<xs:key name="ShbgID">
        <xs:selector xpath="./xags:SHBG"/>
        <xs:field xpath="@SHBG_ID"/>
</xs:key>
<xs:key name="ShbtID">
        <xs:selector xpath="./xags:SHBT"/>
        <xs:field xpath="@SHBT_ID"/>
</xs:key>
<xs:keyref name="ShbgIDRef" refer="ShbgID">
        <xs:selector xpath="./xags:SHBT"/>
        <xs:field xpath="@SHBG_IDREF"/>
</xs:keyref>
<xs:key name="SuctID">
        <xs:selector xpath="./xags:SUCT"/>
        <xs:field xpath="@SUCT_ID"/>
</xs:key>
<xs:key name="TncpID">
        <xs:selector xpath="./xags:TNPC"/>
        <xs:field xpath="@TNPC_ID"/>
</xs:key>
```

```
    <xs:key name="TrigID">
        <xs:selector xpath="./xags:TRIG"/>
        <xs:field xpath="@TRIG_ID"/>
    </xs:key>
    <xs:key name="TrixID">
        <xs:selector xpath="./xags:TRIX"/>
        <xs:field xpath="@TRIX_ID"/>
    </xs:key>
    <xs:keyref name="TrigIDRef" refer="TrigID">
        <xs:selector xpath="./xags:TRIX"/>
        <xs:field xpath="@TRIG_IDREF"/>
    </xs:keyref>
</xs:element>
<xs:element name="PROJ">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="PROJ_NAME" type="xs:string" minOccurs="0"/>
            <xs:element name="PROJ_LOC" type="xs:string" minOccurs="0"/>
            <xs:element name="PROJ_CLNT" type="xs:string" minOccurs="0"/>
            <xs:element name="PROJ_CONT" type="xs:string" minOccurs="0"/>
            <xs:element name="PROJ_ENG" type="xs:string" minOccurs="0"/>
            <xs:element name="PROJ_MEMO" type="xs:string" minOccurs="0"/>
            <xs:element name="PROJ_DATE" type="xs:date" minOccurs="0"/>
            <xs:element name="PROJ_CID" type="xs:string" minOccurs="0"/>
            <xs:element name="PROJ_PROD" type="xs:string" minOccurs="0"/>
            <xs:element name="PROJ_RECV" type="xs:string" minOccurs="0"/>
            <xs:element name="PROJ_ISNO" type="xs:float" minOccurs="0"/>
            <xs:element name="PROJ_STAT" type="xs:string" minOccurs="0"/>
            <xs:element name="PROJ_AGS" type="xs:float" minOccurs="0"/>
            <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="PROJ_ID" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="HOLE">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="HOLE_NAME" type="xs:string" minOccurs="0"/>
            <xs:element name="HOLE_TYPE" type="xs:string" minOccurs="0"/>
            <xs:element name="HOLE_NATE" type="MiscType" minOccurs="0"/>
            <xs:element name="HOLE_NATN" type="MiscType" minOccurs="0"/>
            <xs:element name="HOLE_GL" type="LengthType" minOccurs="0"/>
            <xs:element name="HOLE_FDEP" type="LengthType" minOccurs="0"/>
            <xs:element name="HOLE_STAR" type="xs:date" minOccurs="0"/>
            <xs:element name="HOLE_LOG" type="xs:string" minOccurs="0"/>
            <xs:element name="HOLE_REM" type="xs:string" minOccurs="0"/>
            <xs:element name="HOLE_ENDD" type="xs:date" minOccurs="0"/>
            <xs:element name="HOLE_BACD" type="xs:date" minOccurs="0"/>
            <xs:element name="HOLE_CREW" type="xs:string" minOccurs="0"/>
            <xs:element name="HOLE_ORNT" type="MiscType" minOccurs="0"/>
            <xs:element name="HOLE_INCL" type="MiscType" minOccurs="0"/>
            <xs:element name="HOLE_EXC" type="xs:string" minOccurs="0"/>
            <xs:element name="HOLE_SHOR" type="xs:string" minOccurs="0"/>
            <xs:element name="HOLE_STAB" type="xs:string" minOccurs="0"/>
            <xs:element name="HOLE_DIML" type="LengthType" minOccurs="0"/>
            <xs:element name="HOLE_DIMW" type="LengthType" minOccurs="0"/>
            <xs:element name="HOLE_LOCM" type="xs:string" minOccurs="0"/>
            <xs:element name="HOLE_LOCA" type="xs:string" minOccurs="0"/>
            <xs:element name="HOLE_CLST" type="xs:string" minOccurs="0"/>
            <xs:element name="HOLE_STAT" type="xs:string" minOccurs="0"/>
            <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="HOLE_ID" type="xs:string" use="required"/>
        <xs:attribute name="PROJ_IDREF" type="xs:string" use="required"/>
    </xs:complexType>
    </xs:element>
    <xs:element name="SAMP">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SAMP_TOP" type="LengthType" minOccurs="0"/>
                <xs:element name="SAMP_REF" type="xs:string" minOccurs="0"/>
                <xs:element name="SAMP_TYPE" type="xs:string" minOccurs="0"/>
                <xs:element name="SAMP_DIA" type="LengthType" minOccurs="0"/>
                <xs:element name="SAMP_BASE" type="LengthType" minOccurs="0"/>
                <xs:element name="SAMP_DESC" type="xs:string" minOccurs="0"/>
                <xs:element name="SAMP_UBLO" type="xs:integer" minOccurs="0"/>
                <xs:element name="SAMP_REM" type="xs:string" minOccurs="0"/>
                <xs:element name="SAMP_DATE" type="xs:date" minOccurs="0"/>
                <xs:element name="SAMP_TIME" type="xs:time" minOccurs="0"/>
                <xs:element name="SAMP_BAR" type="PressureType" minOccurs="0"/>
                <xs:element name="SAMP_WDEP" type="LengthType" minOccurs="0"/>
                <xs:element name="SAMP_TEMP" type="TemperatureType" minOccurs="0"/>
                <xs:element name="SAMP_PRES" type="PressureType" minOccurs="0"/>
                <xs:element name="SAMP_FLOW" type="FlowType" minOccurs="0"/>
                <xs:element name="SAMP_PREP" type="xs:string" minOccurs="0"/>
                <xs:element name="GEOL_STAT" type="xs:string" minOccurs="0"/>
                <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="SAMP_ID" type="xs:string" use="required"/>
            <xs:attribute name="HOLE_IDREF" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="FILE_FSET" type="FileType"/>
    <!--=======================Definition of Association Types=========================-->
    <xs:complexType name="MiscType">
        <xs:simpleContent>
            <xs:extension base="xs:float">
                <xs:attribute name="uom" type="MiscUnit" use="required"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:simpleType name="MiscUnit">
        <xs:annotation>
            <xs:documentation>
     deg:degrees, m2/MN:square meters per megaNewton, ft2/t:square feet per ton, m2/yr:square meters per year,
ft2/yr:square feet per year. ft2/day:square feet per day, Nm:Newton meter, uV:microVolt, mV:milliVolt, ohm:Ohm,
ohmc:Ohm centimeter, uS/cm:microSiemens per centimeter, kJ/kg:kiloJoules per kilogram, counts/s:counts per second
  </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
            <xs:enumeration value="deg"/>
            <xs:enumeration value="m2/MN"/>
            <xs:enumeration value="ft2/t"/>
            <xs:enumeration value="m2/yr"/>
            <xs:enumeration value="ft2/yr"/>
            <xs:enumeration value="ft2/day"/>
            <xs:enumeration value="Nm"/>
            <xs:enumeration value="uV"/>
            <xs:enumeration value="mV"/>
            <xs:enumeration value="ohm"/>
            <xs:enumeration value="ohmcm"/>
            <xs:enumeration value="uS/cm"/>
            <xs:enumeration value="kJ/kg"/>
            <xs:enumeration value="counts/s"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="LengthType">
        <xs:simpleContent>
            <xs:extension base="xs:float">
                <xs:attribute name="uom" type="LengthUnit" use="required"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:simpleType name="LengthUnit">
        <xs:annotation>
            <xs:documentation>
     m:meter, ft:foot, mm:millimeter, cm:centimeter, km:kilometer, in:inch, yd:yard, mi:mile
   </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
            <xs:enumeration value="m"/>
            <xs:enumeration value="ft"/>
            <xs:enumeration value="mm"/>
            <xs:enumeration value="cm"/>
            <xs:enumeration value="km"/>
            <xs:enumeration value="in"/>
            <xs:enumeration value="yd"/>
            <xs:enumeration value="mi"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="PressureType">
        <xs:simpleContent>
            <xs:extension base="xs:float">
                <xs:attribute name="uom" type="PressureUnit" use="required"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:simpleType name="PressureUnit">
        <xs:annotation>
            <xs:documentation>
     kM/m2:kiloNewtons per square meter, kPa:kiloPascal, MN/m2:megaNewtons per square meter, MPa:megaPascal,
GPa: gigaPascal, psi:pounds per square inch, psf:pounds per square foot, ksi:kips per square inch, ksf:kips per square
foot, tsf:tons per square foot, kg.cm2:kilograms per square centimeter, bar:bar
   </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
            <xs:enumeration value="kN/m2"/>
            <xs:enumeration value="kPa"/>
            <xs:enumeration value="MN/m2"/>
            <xs:enumeration value="MPa"/>
            <xs:enumeration value="GPa"/>
            <xs:enumeration value="psi"/>
            <xs:enumeration value="psf"/>
            <xs:enumeration value="ksi"/>
            <xs:enumeration value="ksf"/>
            <xs:enumeration value="tsf"/>
            <xs:enumeration value="kg/cm2"/>
            <xs:enumeration value="bar"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="TemperatureType">
        <xs:simpleContent>
            <xs:extension base="xs:float">
                <xs:attribute name="uom" type="TemperatureUnit" use="required"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:simpleType name="TemperatureUnit">
        <xs:annotation>
            <xs:documentation>
    DegC:degree Celsius, DegF:degree Fahrenheit
   </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
            <xs:enumeration value="DegC"/>
```

```
               <xs:enumeration value="DegF"/>
          </xs:restriction>
     </xs:simpleType>
     <xs:complexType name="FlowType">
          <xs:simpleContent>
               <xs:extension base="xs:float">
                    <xs:attribute name="uom" type="FlowUnit" use="required"/>
               </xs:extension>
          </xs:simpleContent>
     </xs:complexType>
     <xs:simpleType name="FlowUnit">
          <xs:annotation>
               <xs:documentation>
     l/s:liters per second, l/min:liters per minute, m3/s:cubic meters per second, gpm:gallons per minute, mgd:million
gallons per day, cfs:cubic feet per second
   </xs:documentation>
          </xs:annotation>
          <xs:restriction base="xs:string">
               <xs:enumeration value="l/s"/>
               <xs:enumeration value="l/min"/>
               <xs:enumeration value="m3/s"/>
               <xs:enumeration value="gpm"/>
               <xs:enumeration value="mgd"/>
               <xs:enumeration value="cfs"/>
          </xs:restriction>
     </xs:simpleType>
     <xs:complexType name="DensityType">
          <xs:simpleContent>
               <xs:extension base="xs:float">
                    <xs:attribute name="uom" type="DensityUnit" use="required"/>
               </xs:extension>
          </xs:simpleContent>
     </xs:complexType>
     <xs:simpleType name="DensityUnit">
          <xs:annotation>
               <xs:documentation>
     kN/m3:kiloNewtons per cubic meter, Mg/m3:megagrams per cubic meter, pcf:pounds per cubic foot, g/cm3:grams
per cubic centimeter, kg/m3:kilograms per cubic meter, kg/m:kilograms per meter run
   </xs:documentation>
          </xs:annotation>
          <xs:restriction base="xs:string">
               <xs:enumeration value="kN/m3"/>
               <xs:enumeration value="Mg/m3"/>
               <xs:enumeration value="pcf"/>
               <xs:enumeration value="g/cm3"/>
               <xs:enumeration value="kg/m3"/>
               <xs:enumeration value="kg/m"/>
          </xs:restriction>
     </xs:simpleType>
     <xs:complexType name="VelocityType">
          <xs:simpleContent>
               <xs:extension base="xs:float">
                    <xs:attribute name="uom" type="VelocityUnit" use="required"/>
               </xs:extension>
          </xs:simpleContent>
     </xs:complexType>
     <xs:simpleType name="VelocityUnit">
          <xs:annotation>
               <xs:documentation>
     mm/min:millimeter per minute, mm/s:millimeter per second, cm/s:centimeter per second, m/s:meter per second,
km/hr:kilometers per hour, ft/min:feet per minute, mph:miles per hour
   </xs:documentation>
          </xs:annotation>
          <xs:restriction base="xs:string">
               <xs:enumeration value="mm/min"/>
```

```
            <xs:enumeration value="mm/s"/>
            <xs:enumeration value="cm/s"/>
            <xs:enumeration value="m/s"/>
            <xs:enumeration value="km/hr"/>
            <xs:enumeration value="ft/min"/>
            <xs:enumeration value="mph"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="ForceType">
        <xs:simpleContent>
            <xs:extension base="xs:float">
                <xs:attribute name="uom" type="ForceUnit" use="required"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:simpleType name="ForceUnit">
        <xs:annotation>
            <xs:documentation>
   N:Newton, kN:kiloNewton, MN:megaNewton, lbf:pounds force, tonf:tons force, kgf:kilograms force
  </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
            <xs:enumeration value="N"/>
            <xs:enumeration value="kN"/>
            <xs:enumeration value="MN"/>
            <xs:enumeration value="lbf"/>
            <xs:enumeration value="tonf"/>
            <xs:enumeration value="kgf"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="PercentageType">
        <xs:simpleContent>
            <xs:extension base="xs:integer">
                <xs:attribute name="uom" type="PercentageUnit" use="required"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:simpleType name="PercentageUnit">
        <xs:annotation>
            <xs:documentation>
   Percent value in the range [0, 100].
  </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:unsignedShort">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="100"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="FileType">
        <xs:annotation>
            <xs:documentation>This type is used by all types needing to attach files</xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="Name" type="xs:string"/>
        </xs:sequence>
        <xs:attribute ref="xlink:href" use="required"/>
    </xs:complexType>
</xs:schema>
```

ISPT2.xsd:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns="http://gees.usc.edu/XAGS" targetNamespace="http://gees.usc.edu/XAGS" elementFormDefault="qualified">
    <xs:element name="ISPT">
```

```
        <xs:complexType>
            <xs:sequence>
                <xs:element name="uom">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="ISPT_TOP_UNIT" type="LengthUnit"/>
                            <xs:element name="ISPT_NPEN_UNIT" type="LengthUnit" minOccurs="0"/>
                            <xs:element name="ISPT_CAS_UNIT" type="LengthUnit" minOccurs="0"/>
                            <xs:element name="ISPT_WAT_UNIT" type="LengthUnit" minOccurs="0"/>
                            <xs:element name="ISPT_SWP_UNIT" type="LengthUnit" minOccurs="0"/>
                            <xs:element name="ISPT_PEN1_UNIT" type="LengthUnit" minOccurs="0"/>
                            <xs:element name="ISPT_PEN2_UNIT" type="LengthUnit" minOccurs="0"/>
                            <xs:element name="ISPT_PEN3_UNIT" type="LengthUnit" minOccurs="0"/>
                            <xs:element name="ISPT_PEN4_UNIT" type="LengthUnit" minOccurs="0"/>
                            <xs:element name="ISPT_PEN5_UNIT" type="LengthUnit" minOccurs="0"/>
                            <xs:element name="ISPT_PEN6_UNIT" type="LengthUnit" minOccurs="0"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="row" minOccurs="0" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="ISPT_TOP" type="xs:float"/>
                            <xs:element name="ISPT_SEAT" type="xs:integer" minOccurs="0"/>
                            <xs:element name="ISPT_MAIN" type="xs:integer" minOccurs="0"/>
                            <xs:element name="ISPT_NPEN" type="xs:float" minOccurs="0"/>
                            <xs:element name="ISPT_NVAL" type="xs:integer" minOccurs="0"/>
                            <xs:element name="ISPT_REP" type="xs:string" minOccurs="0"/>
                            <xs:element name="ISPT_CAS" type="xs:float" minOccurs="0"/>
                            <xs:element name="ISPT_WAT" type="xs:float" minOccurs="0"/>
                            <xs:element name="ISPT_TYPE" type="xs:string" minOccurs="0"/>
                            <xs:element name="ISPT_SWP" type="xs:float" minOccurs="0"/>
                            <xs:element name="ISPT_REM" type="xs:string" minOccurs="0"/>
                            <xs:element name="ISPT_INC1" type="xs:integer" minOccurs="0"/>
                            <xs:element name="ISPT_INC2" type="xs:integer" minOccurs="0"/>
                            <xs:element name="ISPT_INC3" type="xs:integer" minOccurs="0"/>
                            <xs:element name="ISPT_INC4" type="xs:integer" minOccurs="0"/>
                            <xs:element name="ISPT_INC5" type="xs:integer" minOccurs="0"/>
                            <xs:element name="ISPT_INC6" type="xs:integer" minOccurs="0"/>
                            <xs:element name="ISPT_PEN1" type="xs:float" minOccurs="0"/>
                            <xs:element name="ISPT_PEN2" type="xs:float" minOccurs="0"/>
                            <xs:element name="ISPT_PEN3" type="xs:float" minOccurs="0"/>
                            <xs:element name="ISPT_PEN4" type="xs:float" minOccurs="0"/>
                            <xs:element name="ISPT_PEN5" type="xs:float" minOccurs="0"/>
                            <xs:element name="ISPT_PEN6" type="xs:float" minOccurs="0"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
            <xs:attribute name="ISPT_ID" type="xs:string" use="required"/>
            <xs:attribute name="HOLE_IDREF" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

STCN2.xsd:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns="http://gees.usc.edu/XAGS" targetNamespace="http://gees.usc.edu/XAGS" elementFormDefault="qualified">
    <!--==========================Elements================================-->
    <xs:element name="STCN">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="STCN_TYPE" type="xs:string" minOccurs="0"/>
                <xs:element name="STCN_REF" type="xs:string" minOccurs="0"/>
                <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="uom">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="STCN_DPTH_UNIT" type="LengthUnit"/>
                            <xs:element name="STCN_RES_UNIT" type="PressureUnit" minOccurs="0"/>
                            <xs:element name="STCN_FRES_UNIT" type="PressureUnit"
minOccurs="0"/>
                            <xs:element name="STCN_PWP1_UNIT" type="PressureUnit"
minOccurs="0"/>
                            <xs:element name="STCN_PWP2_UNIT" type="PressureUnit"
minOccurs="0"/>
                            <xs:element name="STCN_PWP3_UNIT" type="PressureUnit"
minOccurs="0"/>
                            <xs:element name="STCN_CON_UNIT" type="MiscUnit" minOccurs="0"/>
                            <xs:element name="STCN_TEMP_UNIT" type="TemperatureUnit"
minOccurs="0"/>
                            <xs:element name="STCN_SLP1_UNIT" type="MiscUnit" minOccurs="0"/>
                            <xs:element name="STCN_SLP2_UNIT" type="MiscUnit" minOccurs="0"/>
                            <xs:element name="STCN_REDX_UNIT" type="MiscUnit" minOccurs="0"/>
                            <xs:element name="STCN_FFD_UNIT" type="PercentageUnit"
minOccurs="0"/>
                            <xs:element name="STCN_PMT_UNIT" type="MiscUnit" minOccurs="0"/>
                            <xs:element name="STCN_PID_UNIT" type="MiscUnit" minOccurs="0"/>
                            <xs:element name="STCN_FID_UNIT" type="MiscUnit" minOccurs="0"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="row" minOccurs="0" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="STCN_DPTH" type="xs:float"/>
                            <xs:element name="STCN_RES" type="xs:float" minOccurs="0"/>
                            <xs:element name="STCN_FRES" type="xs:float" minOccurs="0"/>
                            <xs:element name="STCN_PWP1" type="UnionType" minOccurs="0"/>
                            <xs:element name="STCN_PWP2" type="UnionType" minOccurs="0"/>
                            <xs:element name="STCN_PWP3" type="UnionType" minOccurs="0"/>
                            <xs:element name="STCN_CON" type="UnionType" minOccurs="0"/>
                            <xs:element name="STCN_TEMP" type="UnionType" minOccurs="0"/>
                            <xs:element name="STCN_PH" type="UnionType" minOccurs="0"/>
                            <xs:element name="STCN_SLP1" type="UnionType" minOccurs="0"/>
                            <xs:element name="STCN_SLP2" type="UnionType" minOccurs="0"/>
                            <xs:element name="STCN_REDX" type="UnionType" minOccurs="0"/>
                            <xs:element name="STCN_FFD" type="UnionType" minOccurs="0"/>
                            <xs:element name="STCN_PMT" type="UnionType" minOccurs="0"/>
                            <xs:element name="STCN_PID" type="UnionType" minOccurs="0"/>
                            <xs:element name="STCN_FID" type="UnionType" minOccurs="0"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
            <xs:attribute name="STCN_ID" type="xs:string" use="required"/>
            <xs:attribute name="HOLE_IDREF" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:simpleType name="UnionType">
        <xs:union>
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="NA"/>
                </xs:restriction>
            </xs:simpleType>
            <xs:simpleType>
                <xs:restriction base="xs:float"/>
            </xs:simpleType>
        </xs:union>
    </xs:simpleType>
</xs:schema>
```

LabTests2.xsd:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <!--=========================Elements==============================-->
    <xs:element name="CBRG">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SPEC_REF" type="xs:integer"/>
                <xs:element name="SPEC_DPTH" type="LengthType"/>
                <xs:element name="CBRG_COND" type="xs:string" minOccurs="0"/>
                <xs:element name="CBRG_METH" type="xs:string" minOccurs="0"/>
                <xs:element name="CBRG_REM" type="xs:string" minOccurs="0"/>
                <xs:element name="CBRG_NMC" type="PercentageType" minOccurs="0"/>
                <xs:element name="CBRG_IMC" type="PercentageType" minOccurs="0"/>
                <xs:element name="CBRG_200" type="PercentageType" minOccurs="0"/>
                <xs:element name="CBRG_SWEL" type="LengthType" minOccurs="0"/>
                <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
                <!-- <xs:element ref="CBRT" minOccurs="0" maxOccurs="unbounded"/>  -->
            </xs:sequence>
            <xs:attribute name="CBRG_ID" type="xs:string" use="required"/>
            <xs:attribute name="SAMP_
" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="CBRT">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="CBRT_TESN" type="xs:integer"/>
                <xs:element name="CBRT_TOP" type="PercentageType" minOccurs="0"/>
                <xs:element name="CBRT_BOT" type="PercentageType" minOccurs="0"/>
                <xs:element name="CBRT_MCT" type="PercentageType" minOccurs="0"/>
                <xs:element name="CBRT_MCBT" type="PercentageType" minOccurs="0"/>
                <xs:element name="CBRT_BDEN" type="DensityType" minOccurs="0"/>
                <xs:element name="CBRT_DDEN" type="DensityType" minOccurs="0"/>
                <xs:element name="CBRT_SWEL" type="LengthType" minOccurs="0"/>
                <xs:element name="CBRT_REM" type="xs:string" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="CBRT_ID" type="xs:string" use="required"/>
            <xs:attribute name="CBRG_IDREF" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="CHLK">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SPEC_DPTH" type="LengthType"/>
                <xs:element name="SPEC_REF" type="xs:integer"/>
                <xs:element name="CHLK_TESN" type="xs:integer"/>
                <xs:element name="CHLK_CCV" type="xs:float" minOccurs="0"/>
                <xs:element name="CHLK_MC" type="PercentageType" minOccurs="0"/>
                <xs:element name="CHLK_SMC" type="PercentageType" minOccurs="0"/>
                <xs:element name="CHLK_010" type="PercentageType" minOccurs="0"/>
                <xs:element name="CHLK_REM" type="xs:string" minOccurs="0"/>
                <xs:element name="CHLK_CARB" type="PercentageType" minOccurs="0"/>
                <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="CHLK_ID" type="xs:string" use="required"/>
            <xs:attribute name="SAMP_IDREF" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
```

```
<xs:element name="CLSS">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="SPEC_DPTH" type="LengthType"/>
            <xs:element name="SPEC_REF" type="xs:integer"/>
            <xs:element name="CLSS_NMC" type="PercentageType" minOccurs="0"/>
            <xs:element name="CLSS_LL" type="PercentageType" minOccurs="0"/>
            <xs:element name="CLSS_PL" type="PercentageType" minOccurs="0"/>
            <xs:element name="CLSS_BDEN" type="DensityType" minOccurs="0"/>
            <xs:element name="CLSS_DDEN" type="DensityType" minOccurs="0"/>
            <xs:element name="CLSS_PD" type="DensityType" minOccurs="0"/>
            <xs:element name="CLSS_425" type="PercentageType" minOccurs="0"/>
            <xs:element name="CLSS_PREP" type="xs:string" minOccurs="0"/>
            <xs:element name="CLSS_SLIM" type="PercentageType" minOccurs="0"/>
            <xs:element name="CLSS_LS" type="PercentageType" minOccurs="0"/>
            <xs:element name="CLSS_HVP" type="PressureType" minOccurs="0"/>
            <xs:element name="CLSS_HVR" type="PressureType" minOccurs="0"/>
            <xs:element name="CLSS_PPEN" type="PressureType" minOccurs="0"/>
            <xs:element name="CLSS_VNPK" type="PressureType" minOccurs="0"/>
            <xs:element name="CLSS_VNRM" type="PressureType" minOccurs="0"/>
            <xs:element name="CLSS_REM" type="xs:string" minOccurs="0"/>
            <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="CLSS_ID" type="xs:string" use="required"/>
        <xs:attribute name="SAMP_IDREF" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="CMPG">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="SPEC_REF" type="xs:integer"/>
            <xs:element name="SPEC_DPTH" type="LengthType"/>
            <xs:element name="CMPG_TYPE" type="xs:string" minOccurs="0"/>
            <xs:element name="CMPG_MOLD" type="xs:string" minOccurs="0"/>
            <xs:element name="CMPG_375" type="PercentageType" minOccurs="0"/>
            <xs:element name="CMPG_200" type="PercentageType" minOccurs="0"/>
            <xs:element name="CMPG_PDEN" type="DensityType" minOccurs="0"/>
            <xs:element name="CMPG_MAXD" type="DensityType" minOccurs="0"/>
            <xs:element name="CMPG_MCOP" type="PercentageType" minOccurs="0"/>
            <xs:element name="CMPG_REM" type="xs:string" minOccurs="0"/>
            <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
            <!-- <xs:element ref="CMPT" minOccurs="0" maxOccurs="unbounded"/>  -->
        </xs:sequence>
        <xs:attribute name="CMPG_ID" type="xs:string" use="required"/>
        <xs:attribute name="SAMP_IDREF" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="CMPT">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="CMPT_TESN" type="xs:integer"/>
            <xs:element name="CMPT_MC" type="PercentageType" minOccurs="0"/>
            <xs:element name="CMPT_DDEN" type="DensityType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="CMPT_ID" type="xs:string" use="required"/>
        <xs:attribute name="CMPG_IDREF" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="CNMT">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="SPEC_REF" type="xs:integer"/>
            <xs:element name="SPEC_DPTH" type="LengthType"/>
            <xs:element name="CNMT_TYPE" type="xs:string"/>
            <xs:element name="CNMT_TTYP" type="xs:string"/>
            <xs:element name="CNMT_RESL" type="xs:string" minOccurs="0"/>
            <xs:element name="CNMT_UNIT" type="xs:string" minOccurs="0"/>
            <xs:element name="CNMT_CAS" type="xs:string" minOccurs="0"/>
            <xs:element name="CNMT_METH" type="xs:string" minOccurs="0"/>
            <xs:element name="CNMT_PREP" type="xs:string" minOccurs="0"/>
            <xs:element name="CNMT_REM" type="xs:string" minOccurs="0"/>
            <xs:element name="CNMT_LIM" type="xs:string" minOccurs="0"/>
            <xs:element name="CNMT_ULIM" type="xs:string" minOccurs="0"/>
            <xs:element name="CNMT_NAME" type="xs:string" minOccurs="0"/>
            <xs:element name="CNMT_LAB" type="xs:string" minOccurs="0"/>
            <xs:element name="CNMT_CRED" type="xs:string" minOccurs="0"/>
            <xs:element name="CNMT_LBID" type="xs:string" minOccurs="0"/>
            <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="CNMT_ID" type="xs:string" use="required"/>
        <xs:attribute name="SAMP_IDREF" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="CONG">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="SPEC_REF" type="xs:float"/>
            <xs:element name="SPEC_DPTH" type="LengthType"/>
            <xs:element name="CONG_TYPE" type="xs:string" minOccurs="0"/>
            <xs:element name="CONG_COND" type="xs:string" minOccurs="0"/>
            <xs:element name="CONG_REM" type="xs:string" minOccurs="0"/>
            <xs:element name="CONG_INCM" type="MiscType" minOccurs="0"/>
            <xs:element name="CONG_INCD" type="PressureType" minOccurs="0"/>
            <xs:element name="CONG_DIA" type="LengthType" minOccurs="0"/>
            <xs:element name="CONG_HIGT" type="LengthType" minOccurs="0"/>
            <xs:element name="CONG_MCI" type="PercentageType" minOccurs="0"/>
            <xs:element name="CONG_MCF" type="PercentageType" minOccurs="0"/>
            <xs:element name="CONG_BDEN" type="DensityType" minOccurs="0"/>
            <xs:element name="CONG_DDEN" type="DensityType" minOccurs="0"/>
            <xs:element name="CONG_PDEN" type="DensityType" minOccurs="0"/>
            <xs:element name="CONG_SATR" type="PercentageType" minOccurs="0"/>
            <xs:element name="CONG_SPRS" type="PressureType" minOccurs="0"/>
            <xs:element name="CONG_SATH" type="PercentageType" minOccurs="0"/>
            <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="CONG_IVR" type="xs:float" minOccurs="0"/>
            <!-- <xs:element ref="CONS" minOccurs="0" maxOccurs="unbounded"/>  -->
        </xs:sequence>
        <xs:attribute name="CONG_ID" type="xs:string" use="required"/>
        <xs:attribute name="SAMP_IDREF" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="CONS">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="CONS_INCN" type="xs:float"/>
            <xs:element name="CONS_IVR" type="xs:float" minOccurs="0"/>
            <xs:element name="CONS_INCF" type="PressureType" minOccurs="0"/>
            <xs:element name="CONS_INCE" type="xs:float" minOccurs="0"/>
            <xs:element name="CONS_INMV" type="MiscType" minOccurs="0"/>
            <xs:element name="CONS_INCV" type="MiscType" minOccurs="0"/>
            <xs:element name="CONS_INSC" type="xs:float" minOccurs="0"/>
            <xs:element name="CONS_CVRT" type="MiscType" minOccurs="0"/>
            <xs:element name="CONS_CVLG" type="MiscType" minOccurs="0"/>
            <xs:element name="CONS_REM" type="xs:string" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="CONS_ID" type="xs:string" use="required"/>
        <xs:attribute name="CONG_IDREF" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="GRAD">
```

```
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SPEC_REF" type="xs:float"/>
                <xs:element name="SPEC_DPTH" type="LengthType"/>
                <xs:element name="GRAD_SIZE" type="LengthType"/>
                <xs:element name="GRAD_PERP" type="PercentageType" minOccurs="0"/>
                <xs:element name="GRAD_TYPE" type="xs:string" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="GRAD_ID" type="xs:string" use="required"/>
            <xs:attribute name="SAMP_IDREF" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="MCVG">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SPEC_REF" type="xs:float"/>
                <xs:element name="SPEC_DPTH" type="LengthType"/>
                <xs:element name="MCVG_REM" type="xs:string" minOccurs="0"/>
                <xs:element name="MCVG_200" type="PercentageType" minOccurs="0"/>
                <xs:element name="MCVG_NMC" type="PercentageType" minOccurs="0"/>
                <xs:element name="MCVG_PRCL" type="xs:string" minOccurs="0"/>
                <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
                <!-- <xs:element ref="MCVT" minOccurs="0" maxOccurs="unbounded"/>  -->
            </xs:sequence>
            <xs:attribute name="MCVG_ID" type="xs:string" use="required"/>
            <xs:attribute name="SAMP_IDREF" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="MCVT">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="MCVT_TESN" type="xs:integer"/>
                <xs:element name="MCVT_MC" type="PercentageType" minOccurs="0"/>
                <xs:element name="MCVT_RELK" type="xs:float" minOccurs="0"/>
                <xs:element name="MCVT_BDEN" type="DensityType" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="MCVT_ID" type="xs:string" use="required"/>
            <xs:attribute name="MCVG_IDREF" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="FRST">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SPEC_REF" type="xs:integer"/>
                <xs:element name="SPEC_DPTH" type="LengthType"/>
                <xs:element name="FRST_COND" type="xs:string" minOccurs="0"/>
                <xs:element name="FRST_REM" type="xs:string" minOccurs="0"/>
                <xs:element name="FRST_DDEN" type="DensityType" minOccurs="0"/>
                <xs:element name="FRST_MC" type="PercentageType" minOccurs="0"/>
                <xs:element name="FRST_HVE1" type="PercentageType" minOccurs="0"/>
                <xs:element name="FRST_HVE2" type="PercentageType" minOccurs="0"/>
                <xs:element name="FRST_HVE3" type="PercentageType" minOccurs="0"/>
                <xs:element name="FRST_HVE" type="PercentageType" minOccurs="0"/>
                <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="FRST_ID" type="xs:string" use="required"/>
            <xs:attribute name="SAMP_IDREF" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="PTST">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SPEC_REF" type="xs:integer"/>
                <xs:element name="SPEC_DPTH" type="LengthType"/>
                <xs:element name="PTST_TESN" type="xs:integer"/>
                <xs:element name="PTST_REM" type="xs:string" minOccurs="0"/>
                <xs:element name="PTST_COND" type="xs:string" minOccurs="0"/>
                <xs:element name="PTST_SZUN" type="LengthType" minOccurs="0"/>
                <xs:element name="PTST_UNS" type="PercentageType" minOccurs="0"/>
                <xs:element name="PTST_DIA" type="LengthType" minOccurs="0"/>
                <xs:element name="PTST_LEN" type="LengthType" minOccurs="0"/>
                <xs:element name="PTST_MC" type="PercentageType" minOccurs="0"/>
                <xs:element name="PTST_BDEN" type="DensityType" minOccurs="0"/>
                <xs:element name="PTST_DDEN" type="DensityType" minOccurs="0"/>
                <xs:element name="PTST_VOID" type="xs:float" minOccurs="0"/>
                <xs:element name="PTST_K" type="VelocityType" minOccurs="0"/>
                <xs:element name="PTST_TSTR" type="PressureType" minOccurs="0"/>
                <xs:element name="PTST_ISAT" type="PercentageType" minOccurs="0"/>
                <xs:element name="PTST_FSAT" type="PercentageType" minOccurs="0"/>
                <xs:element name="PTST_PDEN" type="DensityType" minOccurs="0"/>
                <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="PTST_ID" type="xs:string" use="required"/>
            <xs:attribute name="SAMP_IDREF" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="RELD">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SPEC_REF" type="xs:integer"/>
                <xs:element name="SPEC_DPTH" type="LengthType"/>
                <xs:element name="RELD_REM" type="xs:string" minOccurs="0"/>
                <xs:element name="RELD_DMAX" type="DensityType" minOccurs="0"/>
                <xs:element name="RELD_375" type="PercentageType" minOccurs="0"/>
                <xs:element name="RELD_063" type="PercentageType" minOccurs="0"/>
                <xs:element name="RELD_020" type="PercentageType" minOccurs="0"/>
                <xs:element name="RELD_DMIN" type="DensityType" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="RELD_ID" type="xs:string" use="required"/>
            <xs:attribute name="SAMP_IDREF" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="ROCK">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SPEC_REF" type="xs:integer"/>
                <xs:element name="SPEC_DPTH" type="LengthType"/>
                <xs:element name="ROCK_PLS" type="PressureType" minOccurs="0"/>
                <xs:element name="ROCK_PLSI" type="PressureType" minOccurs="0"/>
                <xs:element name="ROCK_PLTF" type="xs:string" minOccurs="0"/>
                <xs:element name="ROCK_UCS" type="PressureType" minOccurs="0"/>
                <xs:element name="ROCK_REM" type="xs:string" minOccurs="0"/>
                <xs:element name="ROCK_PREM" type="xs:string" minOccurs="0"/>
                <xs:element name="ROCK_UREMI" type="xs:string" minOccurs="0"/>
                <xs:element name="ROCK_E" type="PressureType" minOccurs="0"/>
                <xs:element name="ROCK_MU" type="xs:float" minOccurs="0"/>
                <xs:element name="ROCK_BRAZ" type="PressureType" minOccurs="0"/>
                <xs:element name="ROCK_BREM" type="xs:string" minOccurs="0"/>
                <xs:element name="ROCK_PORO" type="PercentageType" minOccurs="0"/>
                <xs:element name="ROCK_PORE" type="xs:string" minOccurs="0"/>
                <xs:element name="ROCK_MC" type="PercentageType" minOccurs="0"/>
                <xs:element name="ROCK_BDEN" type="DensityType" minOccurs="0"/>
                <xs:element name="ROCK_DDEN" type="DensityType" minOccurs="0"/>
                <xs:element name="ROCK_PDEN" type="DensityType" minOccurs="0"/>
                <xs:element name="ROCK_DREM" type="xs:string" minOccurs="0"/>
                <xs:element name="ROCK_WTAB" type="PercentageType" minOccurs="0"/>
                <xs:element name="ROCK_WREM" type="xs:string" minOccurs="0"/>
                <xs:element name="ROCK_SDI" type="PercentageType" minOccurs="0"/>
                <xs:element name="ROCK_SREM" type="xs:string" minOccurs="0"/>
                <xs:element name="ROCK_SOUN" type="PercentageType" minOccurs="0"/>
                <xs:element name="ROCK_MREM" type="xs:string" minOccurs="0"/>
                <xs:element name="ROCK_ACV" type="PercentageType" minOccurs="0"/>
                <xs:element name="ROCK_CREM" type="xs:string" minOccurs="0"/>
                <xs:element name="ROCK_AIV" type="PercentageType" minOccurs="0"/>
                <xs:element name="ROCK_IREM" type="xs:string" minOccurs="0"/>
                <xs:element name="ROCK_LOSA" type="PercentageType" minOccurs="0"/>
                <xs:element name="ROCK_LREM" type="xs:string" minOccurs="0"/>
                <xs:element name="ROCK_AAV" type="xs:float" minOccurs="0"/>
                <xs:element name="ROCK_PSV" type="xs:float" minOccurs="0"/>
                <xs:element name="ROCK_FI" type="PercentageType" minOccurs="0"/>
                <xs:element name="ROCK_EI" type="PercentageType" minOccurs="0"/>
                <xs:element name="ROCK_DESC" type="xs:string" minOccurs="0"/>
                <xs:element name="ROCK_SHOR" type="xs:float" minOccurs="0"/>
                <xs:element name="ROCK_PWAV" type="VelocityType" minOccurs="0"/>
                <xs:element name="ROCK_SWAV" type="VelocityType" minOccurs="0"/>
                <xs:element name="ROCK_EMOD" type="PressureType" minOccurs="0"/>
                <xs:element name="ROCK_SG" type="PressureType" minOccurs="0"/>
                <xs:element name="ROCK_SWEL" type="PressureType" minOccurs="0"/>
                <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="ROCK_ID" type="xs:string" use="required"/>
            <xs:attribute name="SAMP_IDREF" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="SHBG">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SPEC_REF" type="xs:integer"/>
                <xs:element name="SPEC_DPTH" type="LengthType"/>
                <xs:element name="SHBG_TYPE" type="xs:string" minOccurs="0"/>
                <xs:element name="SHBG_REM" type="xs:string" minOccurs="0"/>
                <xs:element name="SHBG_PCOH" type="PressureType" minOccurs="0"/>
                <xs:element name="SHBG_PHI" type="MiscType" minOccurs="0"/>
                <xs:element name="SHBG_RCOH" type="PressureType" minOccurs="0"/>
                <xs:element name="SHBG_RPHI" type="MiscType" minOccurs="0"/>
                <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
                <!-- <xs:element ref="SHBT" minOccurs="0" maxOccurs="unbounded"/> -->
            </xs:sequence>
            <xs:attribute name="SHBG_ID" type="xs:string" use="required"/>
            <xs:attribute name="SAMP_IDREF" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="SHBT">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SHBT_TESN" type="xs:integer"/>
                <xs:element name="SHBT_BDEN" type="DensityType" minOccurs="0"/>
                <xs:element name="SHBT_DDEN" type="DensityType" minOccurs="0"/>
                <xs:element name="SHBT_NORM" type="DensityType" minOccurs="0"/>
                <xs:element name="SHBT_DISP" type="VelocityType" minOccurs="0"/>
                <xs:element name="SHBT_PEAK" type="PressureType" minOccurs="0"/>
                <xs:element name="SHBT_RES" type="PressureType" minOccurs="0"/>
                <xs:element name="SHBT_PDIS" type="LengthType" minOccurs="0"/>
                <xs:element name="SHBT_RDIS" type="LengthType" minOccurs="0"/>
                <xs:element name="SHBT_PDEN" type="DensityType" minOccurs="0"/>
                <xs:element name="SHBT_IVR" type="xs:float" minOccurs="0"/>
                <xs:element name="SHBT_MCI" type="PercentageType" minOccurs="0"/>
                <xs:element name="SHBT_MCF" type="PercentageType" minOccurs="0"/>
                <xs:element name="SHBT_REM" type="xs:string" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="SHBT_ID" type="xs:string" use="required"/>
            <xs:attribute name="SHBG_IDREF" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="SUCT">
```

```
    <xs:complexType>
        <xs:sequence>
            <xs:element name="SPEC_REF" type="xs:integer"/>
            <xs:element name="SPEC_DPTH" type="LengthType"/>
            <xs:element name="SUCT_METH" type="xs:string" minOccurs="0"/>
            <xs:element name="SUCT_VAL" type="PressureType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="SUCT_ID" type="xs:string" use="required"/>
        <xs:attribute name="SAMP_IDREF" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="TNPC">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="SPEC_REF" type="xs:integer"/>
            <xs:element name="SPEC_DPTH" type="LengthType"/>
            <xs:element name="TNPC_TESN" type="xs:integer" minOccurs="0"/>
            <xs:element name="TNPC_REM" type="xs:string" minOccurs="0"/>
            <xs:element name="TNPC_DRY" type="ForceType" minOccurs="0"/>
            <xs:element name="TNPC_WET" type="ForceType" minOccurs="0"/>
            <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="TNPC_ID" type="xs:string" use="required"/>
        <xs:attribute name="SAMP_IDREF" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="TRIG">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="SPEC_REF" type="xs:integer"/>
            <xs:element name="SPEC_DPTH" type="LengthType"/>
            <xs:element name="TRIG_TYPE" type="xs:string" minOccurs="0"/>
            <xs:element name="TRIG_COND" type="xs:string" minOccurs="0"/>
            <xs:element name="TRIG_REM" type="xs:string" minOccurs="0"/>
            <xs:element name="TRIG_CU" type="PressureType" minOccurs="0"/>
            <xs:element name="TRIG_COH" type="PressureType" minOccurs="0"/>
            <xs:element name="TRIG_PHI" type="MiscType" minOccurs="0"/>
            <xs:element ref="FILE_FSET" minOccurs="0" maxOccurs="unbounded"/>
            <!-- <xs:element ref="TRIX" minOccurs="0" maxOccurs="unbounded"/>  -->
        </xs:sequence>
        <xs:attribute name="TRIG_ID" type="xs:string" use="required"/>
        <xs:attribute name="SAMP_IDREF" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="TRIX">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="TRIX_TESN" type="xs:integer"/>
            <xs:element name="TRIX_SDIA" type="LengthType" minOccurs="0"/>
            <xs:element name="TRIX_MC" type="PercentageType" minOccurs="0"/>
            <xs:element name="TRIX_CELL" type="PressureType" minOccurs="0"/>
            <xs:element name="TRIX_DEVF" type="PressureType" minOccurs="0"/>
            <xs:element name="TRIX_SLEN" type="LengthType" minOccurs="0"/>
            <xs:element name="TRIX_BDEN" type="DensityType" minOccurs="0"/>
            <xs:element name="TRIX_DDEN" type="DensityType" minOccurs="0"/>
            <xs:element name="TRIX_PWPF" type="PressureType" minOccurs="0"/>
            <xs:element name="TRIX_PWPI" type="PressureType" minOccurs="0"/>
            <xs:element name="TRIX_CU" type="PressureType" minOccurs="0"/>
            <xs:element name="TRIX_STRN" type="PercentageType" minOccurs="0"/>
            <xs:element name="TRIX_MODE" type="xs:string" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="TRIX_ID" type="xs:string" use="required"/>
        <xs:attribute name="TRIG_IDREF" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>
```

```
</xs:schema>
```

Appendix B   XAGS 2.0 Sample XSL File

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="UTF-8"/>
<xsl:template match="/">
<html>
<head><title>Liquefaction Database Data</title>
<script language="JavaScript"><xsl:comment>
// creates and initializes an array of product descriptions
var urls = new Array()
<xsl:for-each select="//FILE_FSET">
urls[<xsl:value-of select="position()"/>] = "http://gees.usc.edu/LiqDB/VNC/<xsl:value-of select="./Name"/>"
</xsl:for-each>
// user function
function doSelect(i)
{
  open(urls[i])
}
      // </xsl:comment></script>
</head>
<body>
   <h3>  <center>
   <xsl:value-of select="//PROJ_NAME"/>  Project
   </center> </h3>
   <h3>  <center>
   <xsl:value-of select="//PROJ_LOC"/>
   </center> </h3>
       <xsl:for-each select="XAGS/HOLE">
    <hr/>  <p>
    <b> Borehole Name: </b>
    <xsl:variable name="name" select="./HOLE_NAME"/>
    <xsl:value-of select="./HOLE_NAME"/>
    <!-- <xsl:value-of select="$name"/> -->
    </p>
    <b> Longitude (degree): </b>
    <xsl:variable name="long" select="./HOLE_NATE"/>
    <i>   <xsl:value-of select="$long"/> </i>,
    <b> Latitude (degree): </b>
    <xsl:variable name="lat" select="./HOLE_NATN"/>
    <i>   <xsl:value-of select="$lat"/> </i>,
    <b> Ground Elevation (m): </b>
    <i>  <xsl:value-of select="./HOLE_GL"/> </i> <br/> <br/>
    <b>
    <a href="http://maps.google.com/maps?q={$lat},{$long}" target="_blank">
           Click here to locate this borehole on Google Map </a></b>
       <p> <b> Drilling Date:         </b>  <xsl:value-of select="./HOLE_STAR"/> </p>
    <p> <b> Owner:         </b> <xsl:value-of select="./HOLE_LOG"/> </p>
    <p> <b> Final Depth:         </b>  <xsl:value-of select="./HOLE_FDEP"/> </p>
    <p> <b> Operator:         </b>  <xsl:value-of select="./HOLE_CREW"/>  </p>
    <p>  <b> Comments:         </b>  <xsl:value-of select="./HOLE_REM"/> </p>
    <b> Link to the original files:     </b>
      <xsl:for-each select="./FILE_FSET">
       <li><a href="javascript:doSelect({position()})">
       <xsl:value-of select="./Name"/></a></li>
  </xsl:for-each><br/>
      <xsl:variable name="hole-id" select="@HOLE_ID"/>
  <xsl:for-each select="//ISPT">
   <xsl:variable name="hole-idref" select="@HOLE_IDREF"/>
  <xsl:if test="$hole-idref=$hole-id"><br/>
```

```
<table width="100%" border="1">
<caption><b>SPT Readings </b> </caption>
 <THEAD>
     <TR>
      <TD width="20%"><B>SPT Top (ft)</B></TD>
      <TD width="60%"><B>Reported Blow Count</B></TD>
      <TD width="20%"><B>SPT Type</B></TD>
     </TR>
 </THEAD>
 <TBODY>
     <xsl:for-each select="//row">
     <TR>
      <TD width="20%"><xsl:value-of select="ISPT_TOP" /></TD>
      <TD width="60%"><xsl:value-of select="ISPT_REP" /></TD>
      <TD width="20%"><xsl:value-of select="ISPT_TYPE" /></TD>
     </TR>
     </xsl:for-each>
 </TBODY>
</table>
   </xsl:if>
  </xsl:for-each>
 <xsl:for-each select="//STCN">
   <xsl:variable name="hole-idref2" select="@HOLE_IDREF"/>
    <xsl:if test="$hole-idref2=$hole-id"><br/>
<table width="100%" border="1">
<caption><b>CPT Readings </b> </caption>
 <THEAD>
     <TR>
      <TD width="20%"><B>Depth of Result (ft)</B></TD>
      <TD width="30%"><B>Cone Resistance (tsf)</B></TD>
      <TD width="30%"><B>Side Friction Resistance (tsf)</B></TD>
      <TD width="20%"><B>Porewater Pressure (psi)</B></TD>
     </TR>
 </THEAD>
 <TBODY>
     <xsl:for-each select="//row">
     <TR>
      <TD width="20%"><xsl:value-of select="STCN_DPTH" /></TD>
      <TD width="30%"><xsl:value-of select="STCN_RES" /></TD>
      <TD width="30%"><xsl:value-of select="STCN_FRES" /></TD>
      <TD width="20%"><xsl:value-of select="STCN_PWP1" /></TD>
     </TR>
     </xsl:for-each>
 </TBODY>
</table>
   </xsl:if>
  </xsl:for-each>
</xsl:for-each>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```