Geo-Engineering Data:

Representation and Standardisation

9 September 2006
University of Nottingham, United Kingdom

IMPLEMENTING XML FOR GEOTECHNICAL DATABASES

Salvatore Caronna
gINT Software, Inc.
Windsor, California, USA

INTRODUCTION

Dissemination of geotechnical data has taken many forms since the introduction of computers. The number of formats appears to have grown world-wide over the years. This trend has made the business of data interchange difficult when it should have gotten simpler. Where standardization has taken root, such as in the UK with the AGS format, need for expansion and maintenance of the standards have brought to light limitations of the existing formats.

XML appears to be the way forward for the industry. Not only is it well established across many other industries but it provides proven structures, support, and pre-made facilities to handle all geotechnical and geoenvironmental requirements with built-in validation and expansion capabilities.

However, data interchange standards, whether based on XML or not, are not standards for the design of databases that are used to work with the data. Data interchange standards have their own design considerations for perform their functions optimally and usage databases have their own considerations.

This paper examines why XML is the way forward and practical considerations in implementation of an XML standard. Examples used will be from the upcoming DIGGS (Data Interchange for Geotechnical and Geoenvironmental Specialists) standard.

WHY XML?

Widely Supported: XML is the endorsed industry standard of the World Wide Web Consortium (W3C) and is supported by all leading software providers in many industries.

Self-Describing Data: XML data does not require the extensive metadata used by traditional data structures to define relationships, describe tables, and data types, etc. because the files themselves contain this information.

Support of all Data Types: XML documents can contain text and numbers, graphics, multimedia objects such as sounds, and active formats like Java applets or ActiveX components.

Easily Verifiable: Tools are available to guarantee that a given XML document is conforms to the basic XML grammar and is valid according to a particular file definition (schema).

Style Sheets: XSL Style Sheets can be created to present the XML in different formats without having to change the original data in any way.

Support of Distributed Data: XML documents can consist of data stored in multiple servers located anywhere on the Web.

Internationalization: XML supports multilingual documents and the Unicode standard.

Automated Conversions: Unit- and coordinate-dependent data can be automatically converted from one system to another if the units and coordinate systems are properly defined.

Platform Independent: XML documents can be transferred between computing platforms. Their self-describing nature minimizes confusion over meaning, values, etc.

Archival Advantages: Storage technology changes constantly as do binary data formats. Since XML is text based, the structure and meaning of the data can be stored and transformed more easily as storage technologies change in the future.

IMPLEMENTATION CONSIDERATIONS

Data Interchange Standards Are Not Working Databases Structures

Much of the DIGGS standard can be adopted as a usage standard, but there are many areas of the standard that act quite well for the purpose of data interchange but are less than optimal for the purpose of data collection, validation, manipulation, and reporting. Usage databases need to be set up to best meet the requirements of the users of the data. The interchange medium should only affect the design of the usage database minimally. Even table and field names in the working database do not need to match the standard. Care must only be taken to ensure that the usage database can be mapped to and from the transfer standard.

Examples

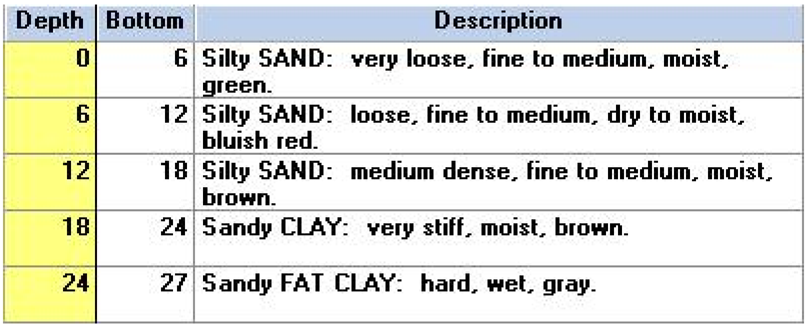
MonitoringPoint

This group of five objects and features is based on the AGS MONR/MONP groups. It is designed to handle any type of depth- and time-related data. This avoids creating special objects for each type of test. This makes the interchange standard robust and easily extensible without changing the structure. However, this structure is probably not one that either the data producer or consumer would wish to work with because there are fields to cover every type of monitoring device. Many, if not most, are irrelevant for any particular test type. Further, this structure mixes all the different monitoring tests together.

For a working structure, it is much easier to work with separate structures for each of the different monitoring tests. Therefore, there would be tables specific for piezometer information and their fields would only apply to piezometers. The same for slope inclinometer, settlement gauges, extensometers, strain gauges, and so on.

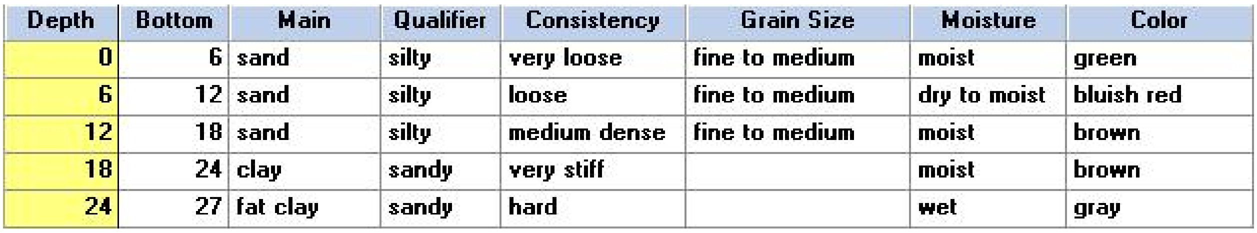
Soil and Rock Descriptions

If your requirements for descriptions entail just a single field, as below, then your database structure could match the DIGGS structure quite closely:



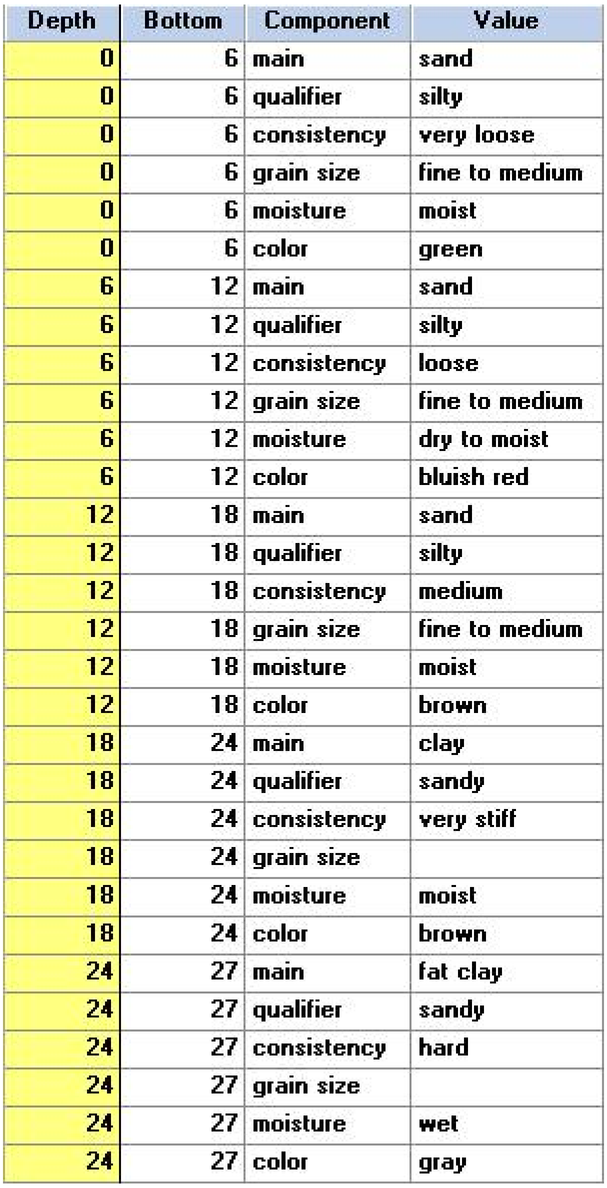
Depth	Bottom	Description
0	6	Silty SAND: very loose, fine to medium, moist, green.
6	12	Silty SAND: loose, fine to medium, dry to moist, bluish red.
12	18	Silty SAND: medium dense, fine to medium, moist, brown.
18	24	Sandy CLAY: very stiff, moist, brown.
24	27	Sandy FAT CLAY: hard, wet, gray.

If you have specific needs for component descriptions to better capture and query the data, then the interchange structure and the probable working structure are vastly different. Following is a sample of a simple working component description structure that is set up the way most people would like to work with these models:



Depth	Bottom	Main	Qualifier	Consistency	Grain Size	Moisture	Color
0	6	sand	silty	very loose	fine to medium	moist	green
6	12	sand	silty	loose	fine to medium	dry to moist	bluish red
12	18	sand	silty	medium dense	fine to medium	moist	brown
18	24	clay	sandy	very stiff		moist	brown
24	27	fat clay	sandy	hard		wet	gray

Following is how one might set up a working structure to mimic the DIGGS structure.



Depth	Bottom	Component	Value
0	6	main	sand
0	6	qualifier	silty
0	6	consistency	very loose
0	6	grain size	fine to medium
0	6	moisture	moist
0	6	color	green
6	12	main	sand
6	12	qualifier	silty
6	12	consistency	loose
6	12	grain size	fine to medium
6	12	moisture	dry to moist
6	12	color	bluish red
12	18	main	sand
12	18	qualifier	silty
12	18	consistency	medium
12	18	grain size	fine to medium
12	18	moisture	moist
12	18	color	brown
18	24	main	clay
18	24	qualifier	sandy
18	24	consistency	very stiff
18	24	grain size	
18	24	moisture	moist
18	24	color	brown
24	27	main	fat clay
24	27	qualifier	sandy
24	27	consistency	hard
24	27	grain size	
24	27	moisture	wet
24	27	color	gray

Like the DIGGS structure, the above table design is quite easy to add new components without having to alter the structure. All that is needed is to add more components to the lookup list. Since this structure mimics DIGGS the import and export process becomes easier as well. However, it is doubtful that many people (if any) would want to work with such a structure.

Laboratory and Field Test Results

The DIGGS standard does not contain raw data fields. Therefore, the data producer always needs many more fields and tables than the interchange standard provides to properly record the raw data for laboratory and field testing.

DIGGS has a multitude of objects and features devoted to laboratory and field test results. The data consumer may wish to have just one laboratory testing table that contains just the results of all laboratory tests and another table for field tests.

In both these scenarios the database needs to be set up differently than the interchange structure to better meet the needs of the data producer and consumer.

Conforming Database Structure to the Interchange Standard Requirements

The interchange standard cannot be ignored when designing a database. Certain fundamental structural elements must be in place. Without some basic structural alignment with the standard, it is possible that invalid data could be exported from the database and data could be lost on import into the database.

Table Key Fields

The key fields in a table uniquely define each data record. For example, in most tables in the U.S. storing depth-related data, the keys are generally the Hole ID and Depth, that is, within one hole there will always be a maximum of one record with any particular depth.

DIGGS generally defines more key fields for uniqueness in their tables than most organizations in are used to dealing with. For example, each record in the SAMPLE feature is defined by the Hole ID, Top Depth, Sample Type, and Sample Number. This allows for multiple records at the same depth in a hole, but the combination of all four key fields must be unique for each record.

Note that DIGGS does not explicitly define key fields for each table. Rather, the keys are implied via the nesting of the tables within the XML structure.

In exporting data to DIGGS, having fewer keys will result in valid records in DIGGS, as long as mandatory key fields are included. Importing data from DIGGS could result in loss of data if there are fewer keys in the database table than the feature in DIGGS. For example, let's say a DIGGS file has two different sample types at the same depth in the same hole. If the target database requires uniqueness of the Hole ID and Depth, only one of the records will be imported.

Table Relationships

It is common for the lab testing parent table to be a direct child of the HOLE table in North America. In DIGGS the corresponding feature (SPECIMEN) is a child of SAMPLE (which is a child of HOLE). With this non-conforming relationship, on export the mapping procedures must somehow associate each specimen with a sample. If this is not done the lab testing results will be "orphans" and this will invalidate the DIGGS file.

Data Types

It is common practice for database designers to specify the data type of fields that should be storing just numeric or date/time data as text fields. This allows input of comments when the data are not recorded. For example, a field that holds water depth could then have a note like "not encountered". Allowing such an entry is not good database design. With the DIGGS standard there is one more reason not to do this. The fields in DIGGS are properly typed, that is, a field holding water depth would be a numeric field and there would be an associate note field for comments. Exporting text data to a numeric field in DIGGS will invalidate the file.

Code Lists

A big part of the DIGGS standard (and many others) is associating lists of valid values (called "code lists" in DIGGS) with certain fields. These lists can be dictated by the standard, a national body, or by a client letting contracts. These lists need to be in line with those that are required, either by using the lists in the design of the database or by mapping your custom lists to the required lists in the import and/or export processes.

This is an important consideration in data usability. For example, if analyses are set up to correlate characteristics of soil based on certain types of drive sample results, and the data generator used different codes than those expected by the analyses, the analyses will not work or will be inaccurate.

Validation

DIGGS files are self-validating:

- the schema can be checked for accuracy
- data are checked against the specified data types
- values in fields associated with code list can be validated
- minimum and maximum values can be assigned to fields

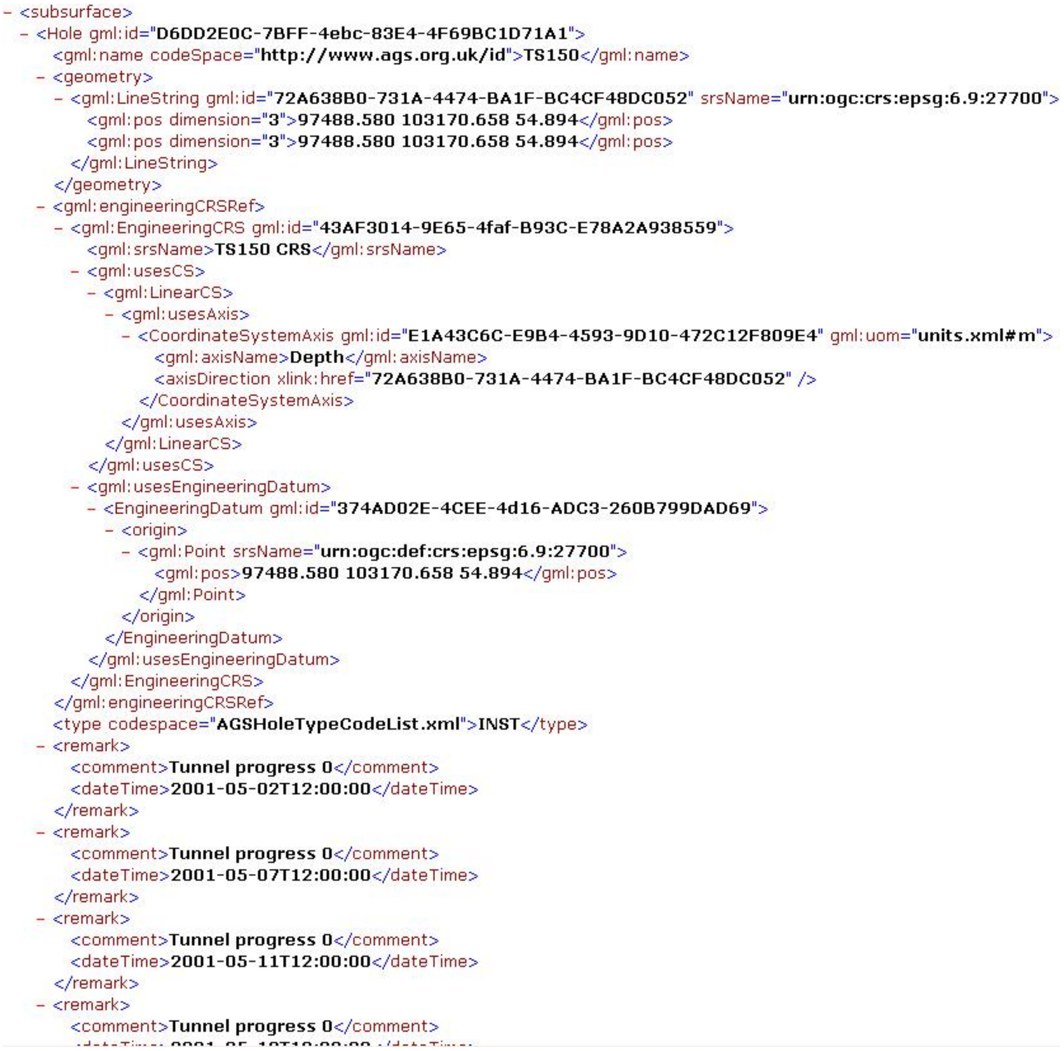
However, there are no automated methods inherent in the standard for performing dependent validations. For example:

- RQD must be less than or equal to Total Recovery.
- If an SPT penetration is 1.5 feet, there must be three blows.
- Specimen depths must be in the range of the corresponding parent sample depth range.
- Layers within the same description classification must not overlap.
- If the liquid limit is 35, a plastic limit of 52 is unreasonable.

With time, budget, and enough people, these dependent validation rules can be written. In the meantime, do not assume that because DIGGS is self-validating that the data are all reasonable.

XML is not Human

Some data interchange standards can be edited by real people using text editors or spreadsheets. DIGGS is not one of those standards. Following is a snippet of a DIGGS file:



```
- <subsurface>
  - <Hole gml:id="D6DD2E0C-7BFF-4ebc-83E4-4F69BC1D71A1">
      <gml:name codeSpace="http://www.ags.org.uk/id">TS150</gml:name>
    - <geometry>
      - <gml:LineString gml:id="72A638B0-731A-4474-BA1F-BC4CF48DC052" srsName="urn:ogc:crs:epsg:6.9:27700">
          <gml:pos dimension="3">97488.580 103170.658 54.894</gml:pos>
          <gml:pos dimension="3">97488.580 103170.658 54.894</gml:pos>
        </gml:LineString>
      </geometry>
    - <gml:engineeringCRSRef>
      - <gml:EngineeringCRS gml:id="43AF3014-9E65-4faf-B93C-E78A2A938559">
          <gml:srsName>TS150 CRS</gml:srsName>
        - <gml:usesCS>
          - <gml:LinearCS>
            - <gml:usesAxis>
              - <CoordinateSystemAxis gml:id="E1A43C6C-E9B4-4593-9D10-472C12F809E4" gml:uom="units.xml#m">
                  <gml:axisName>Depth</gml:axisName>
                  <axisDirection xlink:href="72A638B0-731A-4474-BA1F-BC4CF48DC052" />
                </CoordinateSystemAxis>
              </gml:usesAxis>
            </gml:LinearCS>
          </gml:usesCS>
        - <gml:usesEngineeringDatum>
          - <EngineeringDatum gml:id="374AD02E-4CEE-4d16-ADC3-260B799DAD69">
            - <origin>
              - <gml:Point srsName="urn:ogc:def:crs:epsg:6.9:27700">
                  <gml:pos>97488.580 103170.658 54.894</gml:pos>
                </gml:Point>
              </origin>
            </EngineeringDatum>
          </gml:usesEngineeringDatum>
        </gml:EngineeringCRS>
      </gml:engineeringCRSRef>
      <type codespace="AGSHoleTypeCodeList.xml">INST</type>
    - <remark>
        <comment>Tunnel progress 0</comment>
        <dateTime>2001-05-02T12:00:00</dateTime>
      </remark>
    - <remark>
        <comment>Tunnel progress 0</comment>
        <dateTime>2001-05-07T12:00:00</dateTime>
      </remark>
    - <remark>
        <comment>Tunnel progress 0</comment>
        <dateTime>2001-05-11T12:00:00</dateTime>
      </remark>
    - <remark>
        <comment>Tunnel progress 0</comment>
        <dateTime>2001-05-18T12:00:00</dateTime>
```

The XML/GML basis of DIGGS makes it understandable to some degree by many GIS software packages without special translation. Also, a host of tools are available for validation, coordinate transforms, and unit conversions. The downside is that no human can hope to properly create or edit a DIGGS file in any reasonable period of time.

Specialized software will be needed to read and write DIGGS files. The experience in the UK with the AGS is that the inability to manually manipulate the interchange files is probably a good thing. AGS files can be edited with text editors and spreadsheets and many people have done this and continue to do so. Unfortunately, the results of such human manipulation are generally not satisfactory.

Dialects of the Standard

It is inconceivable that any interchange standard will be implemented in exactly the same manner throughout the world. In fact, it is nearly certain that the standard will have different implementations within different niches of the industry, in different regions of a country, and for specific projects. This has already happened with the UK AGS data interchange standard where there are formal variations in Singapore and Hong Kong and soon to be released variations in Australia and New Zealand.

These variations do not necessarily break the "standardness" of the standard if the basic rules are followed. The DIGGS standard provides a structure which can be altered to accommodate specific requirements. Therefore, it will probably come to pass that there will be a DIGGS UK, DIGGS US, DIGGS AUSTRALIA, etc.

Code Lists

Many fields have finite numbers of possible entries. For example:

- Hole Types
- Sample Types
- Layer Legend Codes
- Triaxial Test Types
- Testing Specifications

For purposes of reporting and querying, lists of allowable values for these fields need to be created. For many of these code lists, internationally usable lists can be created that would be acceptable to the vast majority of users of the standard. However, many will need to be localized for specific parts of the world and different niches within the industry. Even those that can be universally accepted would need to be expanded in specific instances.

Description Classes

Descriptions of subsurface materials possibly represent the most variability within a geotechnical interchange standard. Most interchange specifications have just one field containing the full descriptions. In the past few years it has been recognized that this approach results in loss of significant information. Characteristics like strength, moisture condition, color, gradation, etc. are items that would be useful in analysis of the subsurface conditions. Homogenizing these attributes within one description field render them useless for querying and reporting purposes. In DIGGS, the way to regain that usefulness is to break out these attributes into description classes. This is a special class of a code list. This requires that agreed upon attribute names be assigned. Some internationally agreed upon names can be added to the base standard but custom attributes would probably be required for different parts of the world and different industry niches.

New Tables and Fields

No standard can hope to capture all types of data for all time. New tables and fields will be needed for different user groups.

Usage Guidelines

The richness of the DIGGS standard leads to the possibilities that the same data could be entered in different ways. Usage guidelines will be needed and they may vary between user groups.

Minimizing Variations in the Standard

For any standard to be accepted, the ability to alter it in the ways described above is crucial for acceptance and usability. However, for ease of transference of data between different user communities, variations such as these need to be minimized. This can be done by submittal of these variations with an international body that reviews them and incorporates variations that are appropriate for the international community into the base standard.

SUMMARY

The holy grail of easily interchangeable data is within our reach. However, this brave new world will require significant changes in the way we deal with data and software. New structures need to be put in place, and work must be put into the exchange process to ensure that it be routine and accurate. This will require new specialized software to be written and existing software to be modified significantly.

REFERENCES

Caronna, S (2006), “Practical Considerations in working with Data Interchange Standards”, Geohazards in Transportation in the Appalachian Region, Lexington, Kentucky, August 2006.

DIGGS Usage Guide and Data Dictionary. Draft of 1 July 2006 (unpublished).

Caronna, S and Wade, P (2005), "Problems with Using the AGS Format As a Working Database Structure", Geotechnical and Geoenvironmental Data in Electronic Format Production, Management, and Application, Birmingham, United Kingdom, 19 October 2005.

Caronna, S. (2005). "Data Granularity in the Storage and Reporting of Soil Exploration Information", The Second Annual Geotechnical, Geophysical, and Geoenvironmental Technology Transfer Conference and Expo, Charlotte, North Carolina, 14-15 April 2005.

Caronna, S. (2005). "Geotechnical Data Management Issues for Transportation Authorities", 6th Transportation Specialty Conference, Toronto, Ontario, 2-4 June 2005.

The Association of Geotechnical and Geoenvironmental Specialists (2005), "Electronic Transfer of Geotechnical and Geoenvironmental Data, (Edition 3.1) including addendum May 2005".