Representing Geo-Engineering Data from Instruments and Transducers

David G. Toll

School of Engineering, Durham University, Durham DH1 3LE, United Kingdom and Dept. of Civil Engineering, National University of Singapore; d.g.toll@durham.ac.uk

ABSTRACT: This paper examines ways to store and exchange data from geotechnical instruments and tranducers, as used in the field and the laboratory. The discussion is intended to form part of the on-going debate about standard ways to represent geo-engineering data in electronic form. Data representation standards such as GML (Geography Markup Language), SensorML and TransducerML are considered for their applicability to geo-engineering and for compatibility with work underway in the geo-engineering community (GEF, AGS, NEES and DIGGS). It is concluded that SensorML can be applied in geotechnical engineering even though it has greater complexity than is needed for many commercial geo-engineering applications. An example is given of its use for cone penetration test (CPT) data.

INTRODUCTION

The paper addresses current issues of representation of data from geotechnical instruments and transducers using XML (eXtensible Markup Language). This is part of a larger initiative to develop standard representation schemes for geo-engineering data. The three international geo-engineering societies (International Society for Soil Mechanics and Geotechnical Engineering (ISSMGE), International Society for Rock Mechanics (ISRM) and International Association for Engineering Geology and the Environment (IAEG)) have formed a Joint Technical Committee, JTC2 (http://www.dur.ac.uk/geo-engineering/jtc2); JTC2 will oversee the development of internationally agreed forms of representation of geo-engineering data that can be used to store such data on the World Wide Web and transfer data between computer systems. It is essential that common data standards are agreed internationally, since files created using XML will be globally available on the World Wide Web. The emergence of different formats in different continents would greatly hinder the interchangeability of geo-engineering information.

Weaver et al (2008) identify the advantages of a standard data exchange format as being able to:

- exchange data between databases within an agency or organization
- receive data from consultants in a standard format
- exchange information with other agencies

- perform data quality checks
- exchange data between software packages and providers
- produce software products that are more standardized and more compatible with other products
- utilize and analyze data from various sources in an integrated geoenvironmental/ geotechnical data management system.

Toll (2007) discusses some of the initiatives underway to develop data standards for geo-engineering. Schemes that are particularly relevant here are GEF (Geotechnical Exchange Format: http://www.geffiles.org/), AGS (Association of Geotechnical and Geoenvironmental Specialists: http://www.ags.org.uk/agsml/), NEES (Network for Earthquake Engineering Simulation: http://it.nees.org/) and DIGGS (Data Interchange for Geotechnical and Geoenvironmental Specialists: http://www.diggsml.org/). These are compared with generic representation schemes for observations and sensor data like Geography Markup Language (GML), SensorML and TransducerML.

GEF is a data exchange format developed in the Netherlands (CUR, 2000) mainly for exchanging data from cone penetration tests. It uses a simple file format to represent columns of data. It has recently been applied to centrifuge data (http://www.geffiles.org/).

The AGS format is a data exchange standard for site investigation data (which includes standard field and laboratory tests). The AGS-M format was developed specifically for monitoring data (AGS, 2002). The latest version of the AGS format (AGS, 2004) includes the AGS-M proposal. Chandler et al (2006) describe how the AGS standard can be implemented in XML (AGS, 2005).

NEES (Kutter et al, 2002) has looked at representing research data from physical modelling (particularly centrifuge) and includes ways of storing some of the metadata associated with the raw data itself.

DIGGS (Bray et al, 2008; Weaver et al, 2008) has been established to bring together the work of the AGS with that from University of Florida and COSMOS. DIGGS is building on these existing standards and is now extending into new areas of representation (e.g. geo-environmental aspects) (Weaver et al, 2008).

This paper reviews these proposals and looks for common elements and useful features from each. It does not intend to propose a definitive form of representation for data from geotechnical instruments. The intention is to provide some preliminary proposals to stimulate discussion and to feed into the debate about emerging data standards in geo-engineering.

GEOTECHNICAL DATA STRUCTURES

GEF (CUR, 2000) is a flat-file format and it is difficult to interpret a structure since some commands make specific references to column numbers and scan numbers. The main advantage of GEF is that data is stored as a single data block, which is more efficient in terms of storage space. A simple example of part of a GEF file for Cone Penetration data (CPT) is given in Fig. 1. The command structure of GEF does not fit readily into an object oriented representation; while GEF could be represented in XML, there are other representation techniques that could be adopted more easily.

```
#GEFID = 1,0,0
#PROCEDURECODE = GEF-CPT-Report, 1,1,0, -
#COMPANYID = Durham University
#PROJECTID = GC2008
#FILEDATE = 2007,02,18
#TESTID = GEO-55
#FILEOWNER = D.G. Toll
#COLUMN = 4
#LASTSCAN = 5
#COLUMNINFO = 1, m, penetration length, 1
#COLUMNINFO = 2, MPa, Cone, 2
#COLUMNINFO = 3, MPa, Friction, 3
#COLUMNINFO = 4, kPa, Pore pressure u1, 5
#COLUMNSEPARATOR = ,
#MEASUREMENTTEXT = 9, ground level, horizontal reference level
#ZID = 31000, -2.41
#EOH =
1.52,0.382,0.0127,-2.1
1.54,0.382,0.0137,4.5
1.56,0.375,0.0151,12.6
1.58,0.375,0.0164,19.1
1.60,0.414,0.0169,5.1
```

Fig. 1. An example of GEF for CPT data (after CUR, 2004)

The AGS data exchange format for representing monitoring data (AGS, 2002) was intended to be a generic structure for monitoring instruments. It replaced what had been individual tables for different instruments used in earlier AGS format definitions.

The AGSML structure (AGS, 2005) used the objects <MonP> to represent a monitoring point, <MonR> to represent monitoring readings and <Icct> for insitu chemical and contaminant tests (Fig. 2). In Fig. 2 the <monitoring> object is attached to <Hole>. This can be used to represent a borehole or probe hole, or can be a "location equivalent" i.e. a monitoring point defined by location (even above ground), rather than a physical hole.

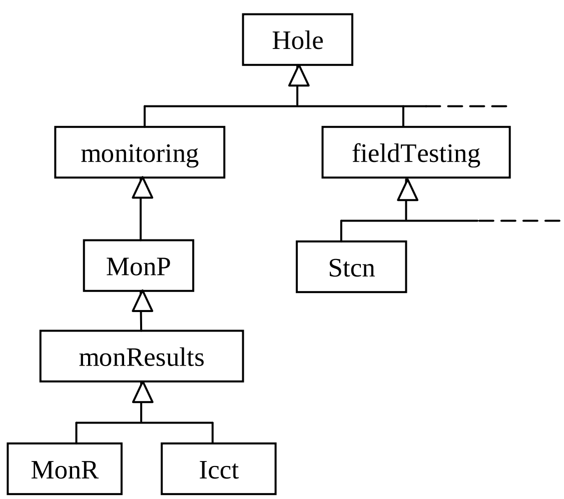

Fig. 2. AGSML representation of monitoring data (after AGS, 2005)

The object <MonR> has the properties <MonR_Date> and <MonR_Time> that would be common to all monitored data. The remaining properties can be used to represent distance, displacement, pressure, rotation/tilt, strain, force, temperature, depth to water, position, head of water or flow, depending on the type of instrument.

The object <Stcn>, used to store cone penetration data (CPT), is also shown in Fig. 2 as an example of a field test data group. The <Stcn> data group is the subject of on-going discussion and will be updated in the next edition of the AGS data format (AGS, 2004). An example of AGSML for CPT data is shown in Fig. 3.

```
<Proj>
    <projId>GC2008</projId>
    <projEng>Durham University</projEng>
    <projDate>2007-02-18</projDate>
    <projAGS>3.1</projAGS>
    <holes>
        <Hole>
            <holeId>GEO-55</holeId>
            <fieldTesting>
                <Stcn>
                    <Stcn_Dpth uom="#m">1.52</Stcn_Dpth>
                    <Stcn_Res uom="#MPa">0.382</Stcn_Res>
                    <Stcn_Fres uom="#MPa">0.0127</Stcn_Fres>
                    <Stcn_Pwp1 uom="#kPa">-2.1</Stcn_Pwp1>
                </Stcn>
                <Stcn>
                    <Stcn_Dpth>1.54</Stcn_Dpth>
                    <Stcn_Res>0.382</Stcn_Res>
                    <Stcn_Fres>0.0137</Stcn_Fres>
                    <Stcn_Pwp1>4.5</Stcn_Pwp1>
                </Stcn>
                <Stcn>
                    <Stcn_Dpth>1.56</Stcn_Dpth>
                    <Stcn_Res>0.375</Stcn_Res>
                    <Stcn_Fres>0.0151</Stcn_Fres>
                    <Stcn_Pwp1>12.6</Stcn_Pwp1>
                </Stcn>
                <Stcn>
                    <Stcn_Dpth>1.58</Stcn_Dpth>
                    <Stcn_Res>0.375</Stcn_Res>
                    <Stcn_Fres>0.0164</Stcn_Fres>
                    <Stcn_Pwp1>19.1</Stcn_Pwp1>
                </Stcn>
                <Stcn>
                    <Stcn_Dpth>1.60</Stcn_Dpth>
                    <Stcn_Res>0.414</Stcn_Res>
                    <Stcn_Fres>0.0169</Stcn_Fres>
                    <Stcn_Pwp1>5.1</Stcn_Pwp1>
                </Stcn>
            </fieldTesting>
        </Hole>
    </holes>
</Proj>
```

Fig. 3. An example of AGSML for CPT data

The NEES data model (Kutter et al, 2002) is shown in Fig.4. This has objects that can represent an analog sensor, a sensor manager (which uses sensor characteristics to convert sensor outputs into engineering values), data formatting (to specify the data type, row, column formatting) and a data concentrator (to convert data streams from individual sensors into a concentrated data stream). This is combined with experiment characteristics (sensor location and orientation).

Bray (2007) has proposed a structure for representing monitoring instruments within DIGGS. This is shown in Fig. 5. Each instrument is assigned a <name> and <readingGroups> for storing the individual readings as <dateTime>, reading pairs. Each reading is hard-typed (i.e. tags with a name specific to that type of reading such as <position> or <planeAngle> or <velocity> are used to define the type of reading).

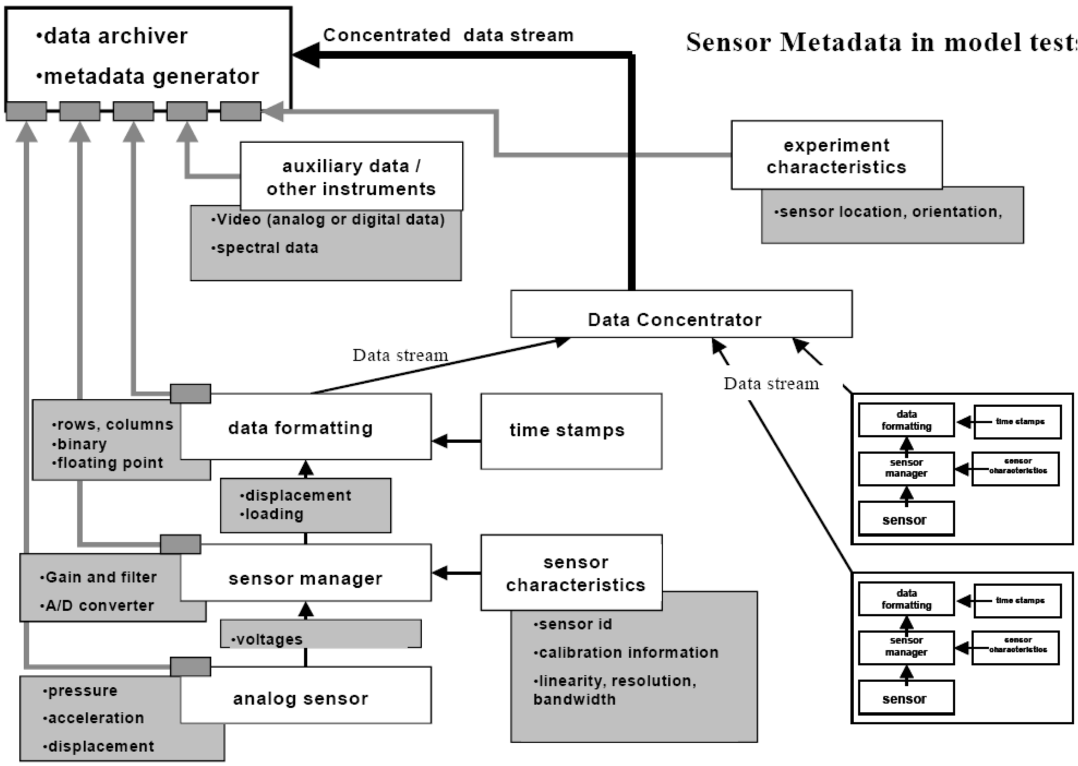

Fig. 4. NEES data model (Kutter et al, 2002)

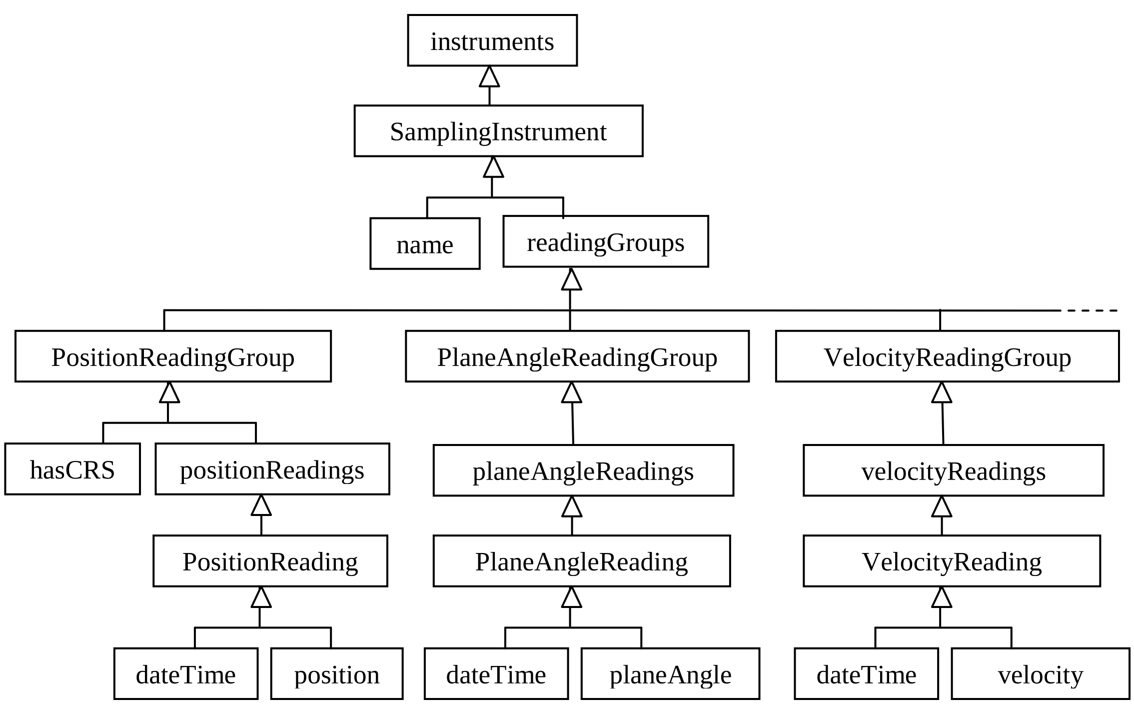

Fig. 5. Proposal for representing instruments within DIGGS (after Bray, 2007)

GENERIC DATA REPRESENTATION SCHEMES

GML (Geography Markup Language; (http://www.opengis.net/gml/) has a data model for representing observations. This uses <location>, <using> and <target> to define the observation device and position. It uses <validTime>,<resultOf> for storing the time and measurement result for a single reading.

The limitations of this simple model in GML have led to the development of SensorML (http://vast.uah.edu/SensorML/). This is largely focused on sensors for earth observation by remote sensing. Nevertheless, much of the work is generic enough to be applicable to other sensors. There is parallel work to develop TransducerML (http://www.transducerml.org/) that can represent greater levels of detail for transducers (sensors and transmitters).

A major feature of SensorML is defining positional data. Two objects, <LocationData> and <OrientationData> are used to define position and orientation of the "Local Frame" with respect to the "Reference Frame". Data can be aggregated together in data groups and data blocks can be used to store data more efficiently than using individual time, measurement pairs.

SensorML (2005) allows either soft or hard "typing". If "soft typed" a generic <component> object is used that is given a "name" attribute that defines the type of measurement (e.g. <component name="coneResistance">). If "hard-typed" a specific tag is created for an individual measurement type (e.g. <coneResistance>). Fig.6 shows an example of how a data group could be defined for a CPT test (using soft typing). This shows the same data set used in Figures 1 and 3.

```
<swe:Data>
    <swe:definition>
        <swe:DataDefinition>
            <swe:dataComponents>
                <swe:DataGroup>
                    <swe:component name="conePenetration">
                        <swe:Quantity definition="Penetration Length" uom="#m"/>
                    </swe:component>
                    <swe:component name="coneResistance">
                        <swe:Quantity definition="Cone Tip Resistance" uom="#MPa"/>
                    </swe:component>
                    <swe:component name="coneFriction">
                        <swe:Quantity definition="Friction Sleeve Resistance" uom="#MPa"/>
                    </swe:component>
                    <swe:component name="conePwp1">
                        <swe:Quantity definition="Pore Water Pressure 1" uom="#kPa"/>
                    </swe:component>
                </swe:DataGroup>
            </swe:dataComponents>
            <swe:encoding>
                <swe:AsciiBlock decimalSeparator="." tokenSeparator="," tupleSeparator=" "/>
            </swe:encoding>
        </swe:DataDefinition>
    </swe:definition>
    <swe:value>
            1.52,0.382,0.0127,-2.1
            1.54,0.382,0.0137,4.5
            1.56,0.375,0.0151,12.6
            1.58,0.375,0.0164,19.1
            1.60,0.414,0.0169,5.1
    <swe:value>
</swe:Data>
```

Fig. 6. An example of SensorML for CPT data

SensorML also provides the ability to provide metadata about the instruments, the person having responsibility for the data etc. It also has quite complex ways to define Processes (how outputs are derived from sensor inputs) and Systems (collections of sensors and processes). However, linear (or even non-linear) calibration curves (which would be the main use in geo-engineering) can be represented more simply.

TransducerML (2006) seems over complex for the relatively routine use of transducers in most commercial geo-engineering applications. However, with continued efforts to combine (or blend) SensorML and TransducerML it might make some functions available that could potentially be useful.

DISCUSSION

SensorML has all the capabilities that would be needed for geo-engineering applications. While Bray's (2007) proposal for representing monitoring instruments for geotechnical applications is a sensible approach, it would be better to adopt the more generic SensorML. Bray's proposal has many similarities with SensorML and could adopt the same approach and terminology without losing any functionality. SensorML also provides similar features to GEF (such as allowing a storage-efficient data block), but has much greater functionality than GEF. AGSML is not storage-efficient in its current form (as each data value has its own pair of tags) and SensorML provides a better solution than this.

The use of a local frame in SensorML could also be used to define a datum relative to a borehole datum (for instance if positions are measured relative to casing level rather than ground level).

SensorML would also be suitable for use in laboratory testing and physical modelling applications. The NEES approach (Kutter et al, 2002) draws on a similar approach to SensorML. SensorML has facilities for storing the metadata and sensor information, and the simple datablock facility illustrated in Fig. 6 would be sufficient for most conventional experiments. It is likely that TransducerML would be needed to provide more complex data streaming facilities (using different time bases for different measurements) that could be necessary for larger-scale physical modelling such as centrifuge testing.

CONCLUSIONS

It is concluded that SensorML has all the capabilities that would be needed for geo-engineering applications. The SensorML approach could be adopted without any loss of functionality compared to current proposals. The data representation approach that has been proposed for incorporating geotechnical monitoring data into DIGGS has many similarities to SensorML, so it would be better to adopt the more generic scheme. This would allow the possibility of greater interoperability between geo-engineering applications and other applications. The SensorML approach can also be extended to data from field tests, and an example is provided for cone penetration test (CPT) data.

Data representation schemes for laboratory test results and physical modelling could also make use of SensorML. It may be that TransducerML could have applications for larger-scale physical modelling where data is being streamed on different time bases.

REFERENCES

AGS (2002) "The AGS-M Format for the Electronic Transfer of Monitoring Data", Construction Industry Research and Information Association Project Report 82, Association of Geotechnical and Geoenvironmental Specialists, Beckenham, Kent.

AGS (2004) "Electronic Transfer of Geotechnical and Geoenvironmental Data", Edition 3.1, Association of Geotechnical and Geoenvironmental Specialists, Beckenham, Kent (with addendum May 2005) (available from http://www.ags.org.uk).

AGS (2005) "Electronic Transfer of Geotechnical and Geoenvironmental Data using XML data format", Association of Geotechnical and Geoenvironmental Specialists, Beckenham, Kent (available from http://www.ags.org.uk/agsml/).

Bray, C.J. (2007) "Monitoring and the SamplingPoint object - Part 2", DIGGS Blog (http://www.diggsml.com/blog).

Bray, C.J., Chandler, R.J., Wathall, S., Hoit, M. and Lefchik, T. (2008) "Extending the Geotechnical Dictionary: Best Practice for Customizing the International Framework for Geotechnical Data", *GeoCongress 2008* (Ibid).

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", *GeoCongress 2006: Geotechnical Engineering in the Information Technology Age*, Atlanta, USA, ASCE, Reston, pp. 112-117.

CUR (2000) "Description of the GEF Language", Civieltechnisch Centrum Uitvoering Research en Regelgeving, Gouda (available from http://www.geonet.nl/software/gef).

CUR (2004) "Geotechnical Exchange Format for CPT-Data", Civieltechnisch Centrum Uitvoering Research en Regelgeving, Gouda (available from http://www.geonet.nl/GEF/GEF-CPT-Report.pdf).

GML (2004) "Geographic information – Geography Markup Language (GML) Version 3.1.1", Report ISO/TC-211/WG-4/PT-19136, International Organisation for Standardisation (http://www.iso.org/).

Kutter, B.L., Wilson, D.W. and Bardet, J.P. (2002) "Metadata Structure for Geotechnical Physical Models (and Simulations?)", http://it.nees.org/documentation/pdf/mtml.pdf .

SensorML (2005) "Sensor Model Language (SensorML) Implementation Specification", Report OGC 05-086, Open Geospatial Consortium Inc. (http://www.opengeospatial.org/).

Toll, D.G. (2007) "Geo-Engineering Data: Representation and Standardisation", *Electronic Journal of Geotechnical Engineering*, http://www.ejge.com/2007/Ppr0699/Ppr0699.htm .

TransducerML (2006) Transducer Markup Language Implementation Specification", Report 06-010r2, Open Geospatial Consortium Inc. (http://www.opengeospatial.org/).

Weaver, S.D., Lefchik, T., Hoit, M. and Beach, K. (2008) "Geoenvironmental and Geotechnical Data Exchange: Setting the Standard", *GeoCongress 2008* (Ibid).