MANAGEMENT OF 3D GEOLOGICAL MODELS AT THE BRITISH GEOLOGICAL SURVEY

Wood, Benjamin[1], Nayembil, Martin [2]
[1] British Geological Survey, Keyworth, NG12 5GG, United Kingdom, bwood@bgs.ac.uk
[2] British Geological Survey, Keyworth, NG12 5GG, United Kingdom, mln@bgs.ac.uk

INTRODUCTION

The British Geological Survey (BGS) has been building digital 3D structural geological models for around 20 years. Today, we have many models, from local to national scale, which together comprise the National Geological Model[1]. The National Geological Model is constantly evolving and being extended and refined by a range of projects. Depending on the type of model (quaternary, bedrock), the geological complexity, the scale, and the nature and distribution of available input data (e.g. boreholes), these models are built using a range of methods. These include 1) the construction of interlocking networks of interpreted cross-sections and related subsurface coverage maps, 2) CAD-based geo-object modelling in a 3D scene, and 3) geo-statistical implict/numerical models. This heterogenous approach to model building allows the geologist to apply the best, most pragmatic method to the project at hand. However, this creates challenges for the systems developer who must seek to archive and manage the model data in a consistent and standardized form.

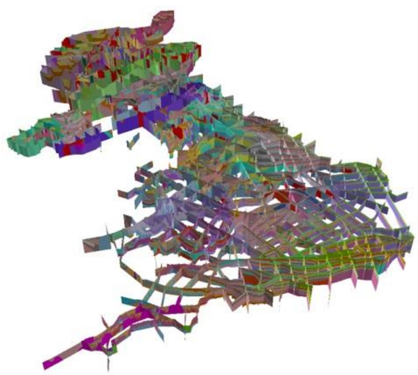

Figure 1 : UK3D National Bedrock Fence Diagram – an model built using interlocking cross-sections.

MODEL DATA TYPES

The various types of data structures resulting from the 3D geological modelling workflows within BGS have been analysed and presented below. Although some specialized edge-cases exist these are not considered here and the majority of data can be categorized into three broad categories;

1) Linework; This may come from the construction of interlocking digital maps and cross-sections, and from structure contours. Typically this interpreted data is used to constrain a surface fitting algorithm in order to achieve the full 3D result.

2) Meshes; These can result from workflows where algorithms (discrete/geometric or geo-statistical) are used to fit or interpolate continuous surfaces from point data. Linework can also be used to constrain such surfaces. They may take the form of surfaces or closed shells.

3) Grids; These can result from geo-statistical or stochastic algorithms and may be 2D (representing an elevation surface or property map) or 3D (representing a volume).

DATA MANAGEMENT

In the simplest case, completed 3D geological model projects can be stored as data files on a local network in native software formats without metadata. This is a practical, low-cost solution that requires no specialized tools or database work. Although this makes storage of models very straightforward, it makes finding them and re-using them more difficult in the long term. Also, file structures are potentially vulnerable to accidental modification, and it may not be clear which version of the data files relates to which version of the model. To tackle this, metadata in the form of documents and spreadsheets may be developed and stored alongside the model data. BGS maintains such a system for work-in-progress, and also as an archive of data files, and it has been found to be practical as a day-to-day solution for project teams, although it does incur manual overhead to maintain and organize.

In order to address the shortcomings of a file-based archiving system, and to make the model data more widely accessible, recently BGS has done a significant amount of work designing databases to hold key data types. The design is based on the concept of 'geological objects', with the logical design of the database mirroring the data model of the object types coming out of the various modelling workflows. Within BGS, the majority of modelling work is currently undertaken using the map and cross-section approach. This is because the methodology is accessible to the geologists and requires little specialized modelling software training, and so the barrier to entry is low. Such models are easy to iterate and refine simply by carrying out additional geological interpretation work, and therefore carry a high level of complexity in terms of versioning and metadata. As such, this is where we began the database implementation because we perceive the interpreted linework resulting from these workflows to be the most abundant stream of new model data, and also the most technically complex to manage.

GEOLOGICAL OBJECT STORE

BGS' corporate database platform is based on an Enterprise Oracle RDBMS with the Spatial extension, and this is the platform we have used to begin developing the 3D model storage database which we are calling the Geological Object Store (GOS). In order to handle the geometries emanating from the map and cross-section modelling projects we have first designed a linework component to the database which is spatially enabled. In-line with the 'geological object' concept, we have chosen not to dis-aggregate related sets of linework into their individual attributed vertices, but instead to keep them in related collections that represent classes of geological object – as sort of object-oriented approach, albeit within a relational database. For example, the database implements the concept of 'geological cross-section'; within this data structure, all linework which belongs to a particular cross-section is held in one data structure. Furthermore, all of the linework held in the database is held only in 2D. Cross-section correlation linework is held in the plane of the cross-section itself, which is essentially a 2D structure (a vertical map). Map linework, whether at surface, or in the sub-surface, is also only held in 2D; it is expected that the individual modelling packages will resolve the 3D element at calculation time, as necessary. Each line within the database has only minimal attribution – normally a coded rock layer value, which is based on the BGS lexicon of named rock units[2] and/or rock classification scheme[3] Each geological object, such as a cross-section, will have basic metadata including a name, and audit information including version numbers, timestamps and user ID's so that the iteration history and provenance of all geological objects can be tracked over time.

The database versioning system is based on that of software source-code versioning. The database maintains a constantly incrementing, global version number that can be stamped onto one or more objects being saved into the database by the geologist. New objects are automatically assigned a unique geological object ID number. Revised objects are stored as 'difference-only', rather than replicating parts of the geometries that have not been changed – this makes small changes to complex objects very efficient in the database, and also allows a detailed audit trail of an objects history to be extracted. The versioning system uses a check-out/check-in motif; geologists wishing to edit objects in the database perform a simple 'check-out' operation which locks the object to their user ID. Other workers can still download the locked objects, but they cannot save edits to those objects until the lock is relinquished. When the geologist is happy with their edit, they perform a simple 'check-in', and provide a comment. Alongside the versioning system we have also implemented an approval mechanism that allows individual objects (or groups of objects) to be moved through a multi-step approval chain. This is a critical part of the GOS database because it allows work-in-progress to sit alongside approved work – it is therefore both a working master version, and a published version. Project geologists will opt to 'check out' non-approved objects for their day-to-day work; product managers will opt to download read-only copies of fully approved objects, which can then be sent to clients directly, or built into 3D shapes for published model outputs. In time, we hope to automate product generation directly from the database.

MESHES AND GRIDS

Our next phase of work is to deal with meshes and grids which commonly result from the use of commercial 3D modelling packages. These objects will re-use the attribution, versioning and approval system already implemented in the GOS, but their geometries will be handled differently. Our analysis has shown that, for example, a triangulated mesh exported from a CAD-type 3D model is really a 'static' object; unlike a cross-section it will never be iterated directly because a revision of the model will always result in a logically 'new' surface object. Therefore we can deal with these objects as geological objects in their own right, and store them intact. Our plan is to store the geometry in both native and agnostic form as a binary object, use the relational aspect of the GOS for the versioning, and use the spatial aspect for object discovery. We will achieve the latter by deriving some form of coverage polygon or shapefile for the object on-the-fly, the result of which will be added to the spatial database.

REFERENCES

[1] BGS National Geological Model, http://www.bgs.ac.uk/research/ukgeology/nationalGeologicalModel/home.html
[2] BGS Lexicon of Named Rock Units, http://www.bgs.ac.uk/Lexicon/
[3] BGS Rock Classification Scheme, http://www.bgs.ac.uk/bgsrcs/

ACKNOWLEDGEMENT

The authors would like to thank all BGS staff who have contributed to the development of these systems over the years, in particular the geological modellers. This paper is published with the permission of the Executive Director of the British Geological Survey.