DRAFT -- To be published in DMT'10 Proceedings
(see http://ngmdb.usgs.gov/Info/dmt/)

The New Mexico Bureau of Geology & Mineral Resources geologic data model, a comparison with other existing models

By Adam S. Read, Geoff Rawling, Daniel J. Koning, Sean D. Connell, J. Michael Timmons, David McCraw, Glen Jones, and Shannon Williams

New Mexico Bureau of Geology & Mineral Resources
801 Leroy Place
Socorro, NM 87801
Telephone: (505) 366-2533
email: adamread@gis.nmt.edu
see: http://geoinfo.nmt.edu/statemap/datamodel

WHAT IS A GEOLOGY DATA MODEL AND WHY WOULD I WANT TO USE ONE?

Like most mapping agencies, the New Mexico Bureau of Geology and Mineral Resources (NMBGMR) has produced geologic maps for many years using a Geographic Information System (GIS). A GIS is essentially a geospatial database that stores information about the shape and position of mapped features as well as associated data. In order for GIS-based maps to be interoperable with other maps, their geo-spatial databases must be organized with a consistent structure. A data model is a standardized database structure (also called a database schema) that defines what features (or entities) are recorded, what their attributes are (often with a pre-defined set of possible values), and how they relate to one another.

HASN'T A GOOD GEOLOGY DATA MODEL ALREADY BEEN CREATED?

Yes and no—several comprehensive data models have been proposed, but none are in common use throughout the country or the world. Geologic maps are extremely complicated documents that attempt to record both geological observations and interpretations in four dimensions—through space and time. There are many reasonable approaches to encoding geological data and a lot of institutional inertia to keep using what has been working, albeit in some cases imperfectly, because it is painful to migrate existing data to a new schema. Adoption of new ways of doing things only occurs when old methods are either too difficult to continue using, and/or newer methods have obvious benefits.

When we decided we needed a better data model, we looked at existing geologic data models at the time and found that they were either too complex to be practical, or otherwise didn't fit our needs. Consequently, we chose to create our own model from scratch, borrowing useful ideas from other models. Since both field mapping and digitization of maps are already fairly labor-intensive, we didn't want to add needless complexity to the process of producing maps. However, we did want the ability to create a fully attributed geologic map in a GIS. Other groups came to the same conclusion and independently produced their own geologic data models.

MODEL COMPARISON

Our model (figure 1) was developed in tandem with the NCGMP09 (http://ngmdb.usgs.gov/Info/standards/NCGMP09/) model and ESRI's Geologic Mapping Template (http://arcscripts.esri.com/details.asp?dbid=16317), and shares several design features with both – but also has some important differences:

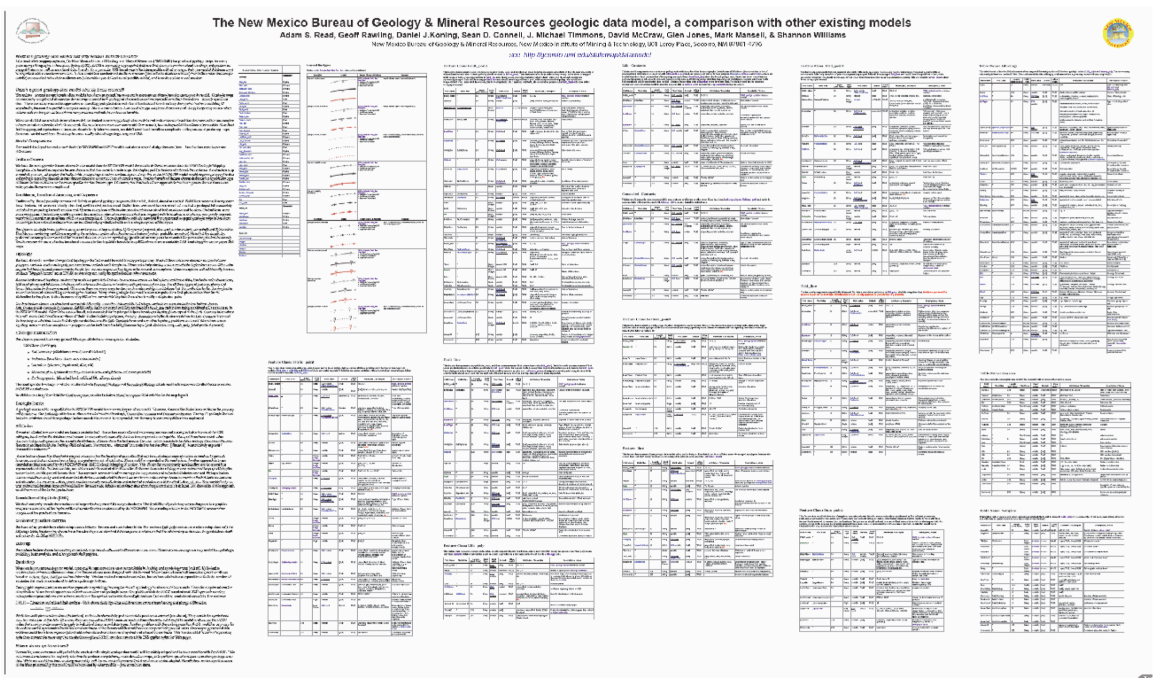

Figure 1. The New Mexico Bureau of Geology & Mineral Resources geologic data model (presented as a poster at the DMT meeting; see full-resolution image at http://ngmdb.usgs.gov/Info/dmt/docs/DMT10_Read.pdf).

Feature Classes

Our model has more feature classes than the NCGMP09 model does and a different structure than the ESRI Geologic Mapping Template. The benefit of many separate feature classes is that it is easier to create maps that display just the features of interest. For instance, if a tectonic map is needed, you can easily just display the faults, folds, and perhaps structure contour layers. To do this in the NCGMP09 model would require querying the data and perhaps exporting features to new feature classes to construct these derivative maps. Another benefit of feature classes dedicated to a particular type of feature is that attributes can be more specific for that feature type. This is particularly apparent with contacts and faults. Our model separates these into separate feature classes with attributes specific to each whereas the NCGMP09 does not. Of course, the drawback of our approach is that having more feature classes makes our geodatabases somewhat larger files.

Confidence, Locational Accuracy, and Exposure

Traditionally, lines (generally contacts and faults) on printed geologic maps are either solid, dashed, dotted, or queried. Solid lines were used to represent linear features that were confidently identified, well located, and exposed. Dotted lines were used for concealed features that a geologist felt reasonably confident in projecting beneath another unit. Queries along lines reflected decreased confidence about both existence and location. Dashed lines were more mysterious. Dashed lines could represent decreased confidence because a contact was mapped with binoculars or air photos, was poorly exposed, wasn't well located in areas of low relief, or was interpolated. The main problem with the standard line types traditionally used on paper geologic maps is that there are multiple inter-related attributes for linear geologic features that cannot effectively be symbolized with such a simple system.

We chose to base our symbolization of linear features on a combination of two required attributes, (1) **Exposure** (exposed, obscured or intermittent, concealed), and (2) **Scientific Confidence** combining confidence regarding the existence and/or identification of a feature (certain, probable, uncertain). Note that for simplicity, positional accuracy is not recorded for lines and does not affect our symbology (positional accuracy can be recorded for points along the line however). Another reason for not attributing locational accuracy is that it quickly becomes very difficult to create a workable field symbology for use on paper field maps. We also allow for the attribution of the **Identification Method** for lines which provides an indication of the locational accuracy expected, but this attribute is generally not symbolized. To see how the combination of **Exposure** and **Confidence** might look on a geologic map, see: http://geoinfo.nmt.edu/statemap/datamodel/symbology/lines

Topology

We have defined a number of important topology rules that should be valid for any geologic map. Most of these rules are obvious: no gaps between polygons, contacts must overlie polygon boundaries, contacts can't dangle, etc. These rules help identify and fix inevitable digitization errors. Other rules require that fold and fault measurements should lie on their respective line types or be marked as exceptions. These exceptions will additionally have an attribute "MappedFeature" set to FALSE so that they can easily be symbolized as minor structures.

A more fundamental topologic relationship exists for point data that can have measurements for both planar and linear data, like faults with slickenlines, fold axial planes and fold axes, foliations with extension lineations, or bedding with paleocurrent vectors. For all these types of features, planar and linear data resides in the same record. Of course, there are many ways to store such a relationship in a database, but this method is by far the simplest to see and understand when editing or viewing the database. Many other geologic data models store one point for a fault plane and another for the slickenline in that plane. It then becomes very difficult to extract this key data from what is fundamentally a single data point.

Our line feature classes are structured somewhat differently than other data models. Lithologic contacts are separated into two feature classes: *Lith_Contacts* and *Concealed_Contacts*. Additionally, faults are stored and fully attributed in *Fault_line* rather than being combined with contacts as in the NCGMP09 model. After faults are attributed, non-concealed faults (that participate in polygon topology) are copied to the *Lith_Contacts* layer where they will retain their LineClass attribute of 'fault'. Before building polygons, the *Map_Boundary* polyline is also copied to the *Lith_Contacts* layer and the topology is validated. Faults that dangle are deleted from the *Lith_Contacts* layer and any other topology problems are fixed. When there are no longer any topology errors -- or exceptions -- polygons can be built from the *Lith_Contacts* layer (and attributed using Lith_poly_label points if present).

Lithologic Classification

We chose to proceed from very general lithologic attributes to more specific attributes:

- **LithClass**: (LithType)
    - **Sedimentary**: (siliciclastic, mixed, nonsiliciclastic)
    - **Volcanic**: (lava flow, dome, ash, volcaniclastic)
    - **Intrusive**: (plutonic, hypabyssal, dike, sill)
    - **Metamorphic**: (metasedimentary, metavolcanic, metaplutonic, unknown protolith)
    - **Anthropogenic**: (disturbed land, artificial fill, tailings, dump).

The most specific lithologic attributes are divided into *PrimaryLithology* and *SecondaryLithology* which could either use uncontrolled terms or use the National Geologic Map Database (NGMDB) vocabularies (note: these are no longer available online, but have been superseded by NCGMP09 vocabularies).

In addition to a long Text field for *UnitDescription*, we also include a *ShortDescription* field suitable for the map legend.

Geologic Events

A geologic events table as specified in the NCGMP09 model is not currently part of our model. However, features like faults have attributes for *Ancestry* and *LastActive*. Our *Lithology* table has attributes for min/max/preferred age, *GeneticEnvironment* and *DepositionalSystem*. Having all geologic features linked to attributes about their geologic history sounds like it could be very useful, but that may be extremely difficult to implement – with any schema.

Extended Attributes

Some feature classes like *DataPoint* are just containers for the location of point data that can be attributed in separate tables as needed. In general, however, most feature classes have a fairly comprehensive set of attributes. These could be expanded as the need arises. Another approach is to use extended attributes as used by the NCGMP09 and ESRI Geologic Mapping Template. This allows for uncommonly used attributes to be stored in a separate related table. In these models, one table is used for extended attributes for all feature classes by relying on user-maintained keys specifying the parent feature and the parent feature class. This approach seems difficult to manage if a large number of extended attributes are used. Perhaps feature classes that have rarely used attributes could have extended attributes in a dedicated table and use one-to-one relationship classes to maintain the link between features and attributes. For instance, Fabric_point(s) could store rarely used attributes and extended attributes in a table called Fabric_pt_attr. This wouldn't rely on user-maintained database keys and would allow for automatic deletion of attribute data when the parent feature is deleted. The downside of this approach would be a proliferation of tables in the geodatabase.

Relationship Feature Classes

We have set up geodatabase relationship classes between features and stand-alone tables. For instance *Lith_poly* units are in a relationship class with the *Lithology* table. Relationship classes have the advantage over standard database joins or relates in that the relationship is stored in the geodatabase itself and not in the ArcMap MXD file.

Subcrop

We include feature classes for creating a bedrock map beneath alluvium/colluvium or other cover. These derivative maps are very useful for hydrologic modeling, basin analysis, and other geotechnical projects.

Symbology

When we began constructing our model, Cartographic Representations were not available in ArcMap, and symbology was (and still is) limited to combinations of three attributes at a time. Our feature classes were designed with this in mind. Many feature classes had somewhat generic attributes based on a *Class*, *Type*, *SubType* attribute hierarchy. This has evolved somewhat over time, but we have tried wherever possible to limit to three the number of attributes that must be considered to define symbology.

Cartographic Representations are another approach to symbology but require that all symbology be abstracted from a code. They also require orientation of symbols to be attributed opposite to azimuth conventions on geologic maps. One problem with the FDGC standard and ESRI approach to using cartographic representations of it is that a number of the symbol codes refer to multiple features that should be symbolized separately (see, for example, figure 2).

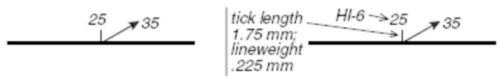

Figure 2. FGDC Geologic map symbol 2.11.13 -- Lineation on inclined fault surface—Tick shows fault dip value and direction; arrow shows bearing and plunge of lineation.

While the fault plane and lineation (slickenline) are both fundamentally part of one data point measurement (see figure 2), they need to be symbolized with two instances of the data. Of course, there are individual FDGC codes for each of these elements, but it might be useful to eliminate the FGDC codes that aren't granular enough to apply to individual features and data types. Another problem with the code approach is that it would be very easy for the code to not be synchronized with the actual attributes of the feature, which would become very confusing to end-users. One way to get around this problem would be to have separate joined tables that allow determination of symbol codes based on attributes. This has the added benefit of separating style from content the same way that standards-compliant HTML encodes semantic content while CSS applies styles for display of Web pages.

WHERE DO WE GO FROM HERE?

Eventually, some consensus will probably be reached and a single geology data model will be widely adopted and be interoperable with GeoSciML (http://www.geosciml.org/). This will make it much easier for anyone who tries to produce compilations, create derivative maps, or to perform spatial analyses of existing geologic map data. While our model has been working reasonably well for us, we don't presume that it will be the model adopted. Nonetheless, we do hope that some of the ideas presented by this model will be borrowed by other models – just as we have done.