NCGMP09—a proposed standard format for digital publication of geologic maps

version 0.8.2, May, 2009

USGS National Geologic Map Database Project and Pacific Northwest Geologic Mapping Project
Contributors: Ralph Haugerud, Stephen Richard, David Soller, and Evan Thoms (in alphabetical order)

NOTE: For the most current version of this document, and for further information, see http://ngmdb.usgs.gov/Info/standards/NCGMP09/

Table of Contents

Introduction

This document proposes a standard format for geologic map publications funded by the U.S. Geological Survey's National Cooperative Geologic Mapping Program. It specifies a database schema to encode content analogous to that contained in a traditional geologic map published by the USGS and state geological surveys. It stipulates an ESRI database format in order to adhere to USGS policy* and because this is the GIS most commonly used in the USGS, in the state geological surveys, and in the larger community. Migration to a non-proprietary format, such as the GML-based GeoSciML, is a worthy goal, and the database schema described here was designed with this in mind.

Further, this design is intended to provide a stepping-stone toward development of multi-map databases, in particular the National Geologic Map Database (NGMDB). The NGMDB Project assists with coordination of database design work between the USGS and state geological surveys, and is mandated to build a national archive of standardized geologic map information. The database design proposed herein will significantly promote that goal.

Objective

Geologic mappers, geologic mapping agencies, and geologic map users would benefit from a standard database schema for digital representation of geologic maps. This document proposes such a schema for the representation of a single geologic map. The schema is focused on the transfer and archiving of map data, with less concern for the creation of map data, the visual representation of map data, or the compilation of data from many maps. With increased use of extended versions of this schema we anticipate reductions in the cost of map production and publication (data compilation and synthesis, review, editing, cartography, pre-press, training, and tool development).

We focus on the representation of a single map for two reasons: this is the issue the geologic community (and our workgroup) understands best, and this is the problem that we perceive is most in need of a solution. The construction and maintenance of multi-map databases brings several issues that we do not here address, including revision/versioning, multiple-scale representations, vocabulary management, maintenance of the stratigraphic lexicon, access control, and maintenance of metadata records.

For the purposes of this design, 'single geologic map' means a package of data (bearing in mind that much geologic 'data' is inherently an interpretation) that pertains to a single portrayal of the geology of some area (the map extent), directly analogous to the traditional paper geologic map. While this package may include different views of the data—e.g. the principal map view, one or more cross sections, perhaps one or more detail maps, each view is represented by a unique mapping between the data and symbols (graphical elements). As a publishable product similar to a conventional geologic map, the database package is attributed to a single author or authors who have either collected the original data and developed the data package and portrayal from that, or have compiled the data from existing sources and developed the portrayal.

This document is intended to bridge between geologic mapping and GIS communities at an operational level. We are codifying lessons from our experience and we expect that this document will be successful largely to the extent that it tells its readers what they already know.

Lessons learned in the last two decades

Geologic map data producers have been developing and using GIS representations of geologic maps for more than two decades. In the course of this effort we have learned some lessons.

The distinction between map data and its symbolization is important. Maps can be represented digitally by scanning them and storing the image file, but this is a very small step towards making the map more useful and its constituent data more easily used for various purposes. Similarly, maps should be more than vector graphic files (e.g. in Adobe Illustrator format). Map data are most usefully stored and analyzed in a geographic information system (GIS), with feature locations given in a real-world spatial reference framework (e.g. UTM10, NAD83) and feature attributes stored explicitly in database tables (e.g., line number 27 is an accurately located thrust fault, line 28 is an approximately located contact, line 29 is the shoreline of Lake Erie on Aug. 27, 1978). A map image, composed of lines, colored areas, patterns, and markers, is a symbolization of the data contained in the database, analogous to a tabular report based on financial data in an accounting database.

Maps need metadata for the overall dataset and for individual elements. Early GIS practices, largely stemming from limitations of storage space and database architecture, as well as paper-map precedent, led to the creation of a significant number of databases in which key fields were populated with symbols (e.g. map unit = Ks) and these symbols were not defined within the database. This is inadequate. Most geologic maps have mixed origins and data qualities; map users benefit from feature-level metadata that describes data source and quality. Map data should be closely linked to authorship because maps are interpretations made by individuals or workgroups, and linked to sponsoring entities because most maps could not be made without significant support from a governmental agency, academic institution, professional society, and (or) private industry.

Real-world database schemas reflect compromises between the intrinsic complexity of geologic map data, the needs of geologists and GISers who work with a schema, the capabilities of GIS and database software, and the limitations of the underlying computer operating system and hardware. Schemas that do not make such compromises are unlikely to be widely used. Even the names of data entities (e.g., of spatial feature sets, tables, fields) must be carefully crafted to be readily understood by users with different backgrounds, facilitate adaptation and re-use of software tools, and promote distribution, translation, and compilation of data.

It is difficult to obtain community acceptance for data architecture (tables and spatial feature sets), data attributes, attribute names, and attribute vocabularies that extend beyond the precedents set by our paper mapping tradition. This conservatism is a good thing because our paper map tradition embodies a great deal of hard-won wisdom. But it is also unfortunate because our tradition reflects compromises necessitated by the limitations of the paper map format.

There is also a widely-shared perception that paper geologic maps, with their subtleties of layout, sometimes carefully ambiguous descriptions, and textual and visual vocabularies that are often opaque to the uninitiated, are not readily used by the public that needs (and pays for) the information contained within these maps. We hope this proposed schema contributes to a better understanding and wider use of geologic map data.

* General policy stated in Section 6.1.3 (http://geology.usgs.gov/usgs/policy/policy6.shtml), supplemented May 24, 1999, by details shown at http://ngmdb.usgs.gov/Info/standards/dataexch/USGSpolicy.html.

Design considerations

We have attempted to meet the following considerations:

- Encode all the content of a traditional paper geologic map.
- Focus on the digital storage and transfer of a single geologic map. Facilitate interactive display and query. Provide a foundation for publication-quality visualization. Do not here try to solve the many-map database problem.
- Define the names and types of all constituent elements in order to meet user needs for consistency and to facilitate re-use of code and tools by map-producers. Use names that have obvious meaning to geologist and GISer alike.
- Address the persistent perception that traditional geologic maps do not meet the public's (and the scientist's) need for consistently named and defined earth materials data, by providing standard terms and definitions.
- Preserve, and facilitate the analysis of, map topology.
- Normalize map data for robustness and compactness of the database, but not to the extent that user comprehension is reduced.
- Allow queryable description of map features with as much (or as little) granularity as desired.
- For flexibility, interoperability, and data longevity, strive toward use of open file formats.

Content of a traditional geologic map

Traditional geologic maps have rich semantic content that should be preserved in any digital publication form. This content is outlined below. Yellow background denotes content for which we do not specify a digital form.

1. Map-graphic
    1. Base map
    2. Map-unit polygons (polygons that cover the mapped area with no voids & no overlaps. May include open water, permanent snowfields and glaciers, and unmapped areas)
    3. Contacts & faults that, with a few exceptions, bound and separate map-unit polygons
    4. As-needed elements
        1. Overlay polygons, e.g., alteration zones, perhaps extensive artificial fill, surface projection of mined-out areas. Note that while these polygons commonly represent features that are within, or beneath, the rocks and deposits represented by map-unit polygons, they are commonly represented on the map as a patterned overlay.
        2. Other lines, including traces of fold hinges, facies boundaries, isograds, cross-section lines, dikes and sills, marker beds, structure contours, etc. In general, overlay polygons and other lines do not conform to the strict topological rules that constrain map-unit polygons and contacts & faults (no polygon voids or overlaps, contacts lie on polygon boundaries, faults may dangle but contacts may not).
        3. Point data, which may include (but is not limited to) structural data (orientation measurements: axes and vectors), samples, geochronologic results, fossils,

chemical analyses, prospect locations, displacement (fault-slip) measurements, and points for map-unit polygons too small to show at scale.

2. Zero to many cross sections (each with elements analogous to map elements, except that the base map is replaced by a topographic profile)
3. Correlation of Map Units diagram that includes unit designators, brackets, dividing lines, and text.
4. Symbolization for above, including
    1. Map-unit area fills (color and optional pattern)
    2. Patterns for overlay polygons
    3. Line symbols and/or point markers for map-unit areas too small to show as polygons at map scale
    4. Text tags for some (but not necessarily all) polygons
    5. Leaders for text tags for some polygons
    6. Line symbols (with variable color, weight, dot-dash pattern, repeated marker ornament, etc) for some lines
    7. Text labels for some lines and groups of lines
    8. Point (marker and/or text) ornaments for some linear features
    9. Markers and/or text for some point data
    10. Leaders for markers and/or text for some point data
5. Description of Map Units, or List of Map Units with descriptions in an accompanying pamphlet. Traditionally does not describe water, permanent snow and glaciers, unmapped area, and some overlays and underlays. Includes headings and some units not shown on map (e.g., Group that is entirely mapped as constituent Formations). Is strongly hierarchical. Each unit shown on the map has area fill color and pattern, tag, unit name, age, description, position in hierarchy, and a paragraph style that (in part) denotes position in hierarchy. Headings and units not shown on the map lack area fill color and pattern and tag.
6. Explanation of line symbols
7. Explanation of point symbols
8. Miscellaneous collar material. Includes report title, author(s), date of publication, publisher, series and series number, mapped-by statement, edited-by statement, cartography-by statement, specification of spatial reference framework, and scale.
9. Zero to many figures
10. Zero to many tables
11. Zero to many additional maps (e.g., sources of map data; distribution of facies in the Cambrian)
12. Extended text, as needed
13. References Cited, as needed

Extensions to traditional geologic map content

We propose several extensions to traditional geologic map content. Two are required: feature-level metadata and the supplementing of map unit descriptions with standard lithology descriptions. Optional extensions are extended attributes and structured, more-detailed descriptions of the ages of geologic events.

Feature-level metadata

All elements of a geologic map database should be accompanied by an explicit record of the data source. Many elements should have explicit statements of scientific confidence: How confident is the author that a feature exists? That it is correctly identified? How confidently are feature attributes known? All spatial elements in the database should be accompanied by quantitative statements of how confidently their location is known. Specification of these confidence values may at first seem to constitute an undue burden on the geologic mapper. However, in some cases default values are appropriate; in others, tools to efficiently assign confidence values can be developed. There should be little extra work for the mapper, and users will benefit significantly from the additional information.

Data source (provenance). Typically, a single map database will have very few data source records, as many features will have identical sources. For a database composed entirely of new mapping, there could be a single data source: "this report". Some data elements have compound sources: geochemical analysis of a rock sample will typically have one source for the map location and stratigraphic provenance of the sample (the field geologist) and another source for the chemical analysis (the geochemist). In such cases, multiple source fields in the relevant data table are appropriate, e.g., LocationSource and AnalysisSource.

Location confidence (spatial accuracy). Reported locations of geologic features are commonly uncertain because of probable error in locating observation points (because, for example, of GPS error or an imprecise base map), because some geologic features are subtle and difficult to locate, or because the locations of features are known indirectly, by inference from the locations of other observations. Because most users locate geologic features in relation to an associated base map, and because most spatial analyses of geologic map data are in relation to the base map or to other data in the same database, we are most interested in the location confidence relative to features on the base map and to other data in the geodatabase. This location confidence is reported as the radius (in meters) of the circle of uncertainty about a point location, or the half-width (in meters) of the zone within which a line is asserted to be located as specified in the FGDC Digital Cartographic Standard for Geologic Map Symbolization (FGDC-STD-013-2006, http://ngmdb.usgs.gov/fgdc_gds/, section 4.2.2.1). As defined here the location confidence is a combination of the error in positioning of the observed feature relative to the base map (the positional accuracy of the FGDC Standard, section 4.2.2) and one aspect of the locatability of the FGDC Standard (how precisely, relative to where I am standing, can this contact be placed?).

Estimates by the map producer of location confidence are likely to be much better informed than estimates made by others, and will be of significant value to map users.

Location method. We propose a field to record the process of observation or inference used to locate each feature. Likely values for this field include "observed in field, location by map inspection", "observed in field, location by recreational GPS", "inferred beneath mapped covering unit", "color or texture change on aerial photograph". This field provides information useful for estimating location confidence and records another aspect of the locatability of the FGDC Standard (section 4.2.1), information that has been denoted on some maps by categorization of contacts and faults as *observed, inferred, or concealed*.

We note that the observed, inferred, or concealed categorization is problematic as a contact mapped as concealed is always inferred, but not necessarily vice versa, and most observation methods involve some degree of inference—e.g., that a color or vegetation change seen on an air photo or a break in slope on a hillside corresponds to a geologic contact.

Existence confidence, identity confidence, and scientific confidence. The FGDC Standard notes that scientific confidence may have multiple dimensions. For a map unit area, scientific confidence has one dimension: confidence that the map unit is correctly identified. In the case of faults, contacts, and other feature traces, the situation is more complex. There may be uncertainty as to whether a boundary between two units is a contact or fault. There may be uncertainty as to what kind of fault is mapped. In both cases, this uncertainty is specified by an identity confidence value. In some cases, the presence of a fault may be suspected but is not certain. Fold hinge surface traces, dikes, marker beds may also be mapped where their existence is suspected but not certain. This uncertainty is specified by existence confidence. Contacts are rarely mapped where their existence is uncertain; if different map units are identified, there must be a boundary of some sort between them, in which case the identity of that boundary may be questionable, but not its existence. Useful values for existence confidence and identity confidence are "standard" (or "std") and "low", or "certain" and "uncertain". This data model includes ExistenceConfidence and IdentityConfidence for line feature classes, and IdentityConfidence for polygon and point observation features. We discussed at length whether to combine these confidence concepts into a single ScientificConfidence field in the database, perhaps with 4 or 6 values to allow for various combinations of existence and identity confidence, but the group decided that for this initial version it makes more sense to leave both as separate fields, as specified in the FGDC Standard. Usage patterns and experience may dictate changes in subsequent versions. We expect that symbolization will in some cases be assigned on the basis of feature type and the appropriate confidence terms.

Orientation confidence. For measurements of rock structures (bedding, foliation, lineation, joints, etc.) it is useful to describe how accurately the orientation has been measured. This is specified as the circular error of a direction (for planar features, of the pole to the plane), which is most usefully expressed as an angular measure similar to the $alpha_{95}$ value often reported for paleomagnetic directions. The OrientationPoints feature class includes an OrientationConfidenceDegrees field to record this uncertainty.

Standard Lithology

A traditional "Description of Map Units" can provide a rich description of earth materials, but may use obscure nomenclature, may use it in a fashion that is inconsistent from map to map, and may fail to be quantitative. Jargon, lack of consistency, and lack of precision reflect the infinite variety of earth materials, imperfect geologic knowledge, and our evolving understanding of earth processes. Nonetheless, geologists and non-geologists alike have decried the lack of consistent, queryable, earth-materials descriptions in traditional geologic maps.

We supplement the richness and flexibility of traditional descriptions with strongly-structured descriptions using a small number of defined lithologic terms. These descriptions are encoded in the table StandardLithology, using term lists provided by the National Geologic Map Database Project (Appendix A). Description of a single map unit may span several rows in this table. This allows description of multi-part (spatially variable, interbedded, block-in-matrix) units, with quantitative or qualitative description of the relative abundance of each component. See the specification of StandardLithology (below) for details.

Geologic Events (ages)

Standard ages in traditional map-unit descriptions—the terms in parentheses following the map-unit name in Description of Map Units—offer limited resolution and do not easily represent multiple ages (e.g., ages for deposition and metamorphism). Traditional paper geologic maps provide no mechanism, short of mention in extended text, for assigning ages to faults and other features that are not map units. An optional GeologicEvent table, in conjunction with ExtendedAttribute associations, allows representation of complex history with multiple ages of some units, more age resolution, and association of ages and history with faults or other structures. The GeologicEvent table includes attributes for assigning upper and lower bounding ages, either using stratigraphic era names or numeric age values. It allows ages to be associated with geologic events such as deposition, crystallization, eruption, cooling, peak metamorphism, and fault displacement.

Extended attributes

An optional ExtendedAttributes table allows structured, queryable description of any element in the database. This table correlates an owning item (e.g. a DescriptionOfMapUnits record) with a property and property value pair (e.g. 'bulk density g/cc', 2.7). Information associated with any table in the database can be added by defining a new property in the Glossary, and assigning values using the ExtendedAttributes table.

Particular strengths of this data structure are its extensibility and efficiency. Extensible in that it doesn't require that designers of the database foresee all attributes that might be stored in the database. Efficient in that memory is not allocated to store values for attributes of features for which they are not defined.

Naming database elements

Fixed, easy-to-comprehend names for all elements are key to a functional geodatabase schema. Names have been chosen accord to the following criteria:

Names convey content to the geoscientist, to the GISer, and to the naive public
Names use uniform concatenation protocol (CamelCase)
Names do not exploit case sensitivity. Note that case should be conserved, as some languages and operating systems distinguish between *ThisName* and *thisName*
Names do not contain spaces or special characters
Long names are OK. A lookup table is provided to convert long names to ESRI shapefile-compatible short names
Names are easy to code and calculate
Names reflect data type
Names point to related tables
Field names which contain "_ID" are reserved for primary keys. These are of the form *TableName_ID* or *FeatureClassName_ID*

We have chosen not to encode the publication identity (map name or map series number) in the names of feature datasets and feature classes. Feature dataset and feature class names that include a map identifier (name or series number) simplify the joint display of multiple publications in an ArcMap project because each layer name will automatically include the map identifier for the layer. Our choice to use a single name for feature datasets and feature classes in all delivery databases keeps the naming scheme simple, and facilitates the coding and sharing of tools to manipulate geodatabases. Utilizing this approach, users must manually update the names of layers in ArcMap projects to reflect the map source for the data in a layer.

Transparent identifiers

We recommend that identifiers in the database for map units, line types, and point feature types have fairly obvious plain English meaning. The map unit identifier is used as a foreign key from the DMU table to various other tables, and our recommendation is that this simply be the unique label used to identify that unit in map displays. Entries in the DMU that are not symbolized on the map may have null map unit identifier values. The type identifiers for lines and points are references to terms in the glossary, and we recommend that these simply be the geologic term for the line or point type represented. This is in contrast to common database design that dictates that identifiers used as foreign keys in database are best implemented as numbers or text string that have no inherent meaning to human users, referred to as opaque identifiers. Though opaque identifiers may be more robust, we think this advantage is outweighed for a data delivery database by the greater intelligibility for people gained using human interpretable identifiers. Note that this specification does not prohibit the use of opaque identifiers, particularly for primary key (table_ID) values.

Open file formats

We encourage the use of open file formats whenever feasible. (1) Open formats facilitate writing and redistribution of 3rd-party code. (2) Open formats reduce the risk of locking data up in formats that become obsolete and unreadable. When open formats are superseded, documentation for them is likely to remain available. (3) Open formats avoid privileging any particular vendor. (4) Open formats are likely to change in a more measured fashion than proprietary formats. Many in the geologic mapping community are still coping with the costs of the relatively rapid transitions from coverages to shapefiles and from shapefiles to geodatabases.

Text should be stored as .txt (ASCII text, ANSI X3.4-1986), .html (HTML, ISO/IEC 15445:2000, W3C HTML 4.01), .odt (Open Document Format, ISO/IEC 26300:2006 or its successor), or .pdf (Page Description Format, ISO 32000-1:2008) files. For images, .png files (ISO/IEC 15948:2004) are preferable to .gif or .tif files. Tables may stored as .dbf files, for which there appears to be no published standard but for which documentation is readily available (e.g., http://www.clicketyclick.dk/databases/xbase/format/), or as xml files that most modern database software can import.

Our desire to endorse open file formats is superseded by our need to prescribe a database file format that preserves topology, allows long attribute names, and works well within ArcGIS, thus we specify the use of ESRI's personal geodatabase (.mdb) or file geodatabase (.gdb) file formats for spatial data. To make geologic map data more widely available, we strongly suggest that data also be released in shapefile format (see below). We look forward to wider implementation and use of text-based, application-independent delivery formats such as GeoSciML.

Review, comment, and revision

This document introduces the proposed NCGMP standard database schema. We do not consider it final. We ask that you review the document and the schema and provide comment so that we can improve the database design, the documentation that explains it, and the tools and templates that facilitate its use.

This design is an outcome of more than a decade's work among numerous individuals and projects, in NCGMP and in other programs and in many agencies. In particular, during this period the National Geologic Map Database project has been charged under the Geologic Mapping Act to support the development of geologic map standards and guidelines. Throughout this time, it has been clear that there is no single "right" approach, that database needs vary from project to project, and these needs may contrast markedly with the long-term requirements for the national database. This design, as noted, is intended to address the needs of geologic mapping projects particularly for data delivery, and so we seek consensus on a design that meets those needs.

We welcome your comments and suggestions for improvements to this design, to the documentation, and to the tools that facilitate its use. Please contact us via email, at ncgmp09@flagmail.wr.usgs.gov.

Regarding availability and maintenance of this database design, under the authority of the Geologic Mapping Act of 1992 (and subsequent reauthorizations), the National Geologic Map Database project will function on behalf of the NCGMP as coordinator of database design changes and maintenance. This activity will be conducted in cooperation with NCGMP projects and other identified stakeholders (e.g., the Association of American State Geologists).

Required and suggested contents of a digital geologic map publication

For a map publication named mapXYZ, the publication package includes the following files:

mapXYZ.pdf *Reference map visualization. Publication quality*

mapXYZ-browse.png *Browse graphic. A **small** file*

mapXYZ-pamphlet.pdf *Map pamphlet, as needed*

mapXYZ-metadata.txt (or .html) *FGDC metadata*

mapXYZ-gdb.zip *When unzipped, this file contains:*

- mapXYZ.gdb (file geodatabase folder) or mapXYZ.mdb (personal geodatabase file)
- resources (folder)
  - figures (.png, .pdf, .tif) *As needed*
  - tables (.dbf, .ods, .xls) *As needed*
  - CMU (.pdf, .png, …) *Optional. Graphic representation of correlation of map units*
  - DMU (.pdf) *Optional. Additional document for description of map units*
  - AreaSymbol.style *ArcGIS style file for area fills used in preferred symbolization of map. Suggested; could be replaced with layer file*
  - FgdcSymbol.style *ArcGIS style file for line and marker symbols used in preferred symbolization of map. Suggested, could be replaced with layer file. May be a subset of the complete FGDC geology symbol set, but must include all symbols specified elsewhere in database*
- mapXYZ-pamphlet.pdf
- base.gdb or base.mdb (folder or file) *Optional, needed if base-map geospatial data are not published elsewhere*
- mapXYZ-metadata (.txt., .html, …)

mapXYZ.mxd	*ArcMap document stored with relative paths and including relevant VBA macros*
mapXYZ-simple.zip	*Optional simple version of database. See below for contents*
mapXYZ-shp.zip	*Optional open version of database. See below for contents*

The geodatabase schema

There are required, as-needed, and optional elements in a single-map geologic map geodatabase (Figure 1). These elements are specified below. For each element (feature dataset, feature class, non-spatial table) we provide a name, identify the element type, state whether it is required, to be used as needed, or optional, and enumerate the fields (attributes) in the relevant table. Unless otherwise noted, all fields are of data type text (=string). Any length is appropriate, so long as it is sufficient to store the associated values; we recommend 255 characters for most fields. For each field we briefly discuss content and domains where appropriate. For some elements this is followed by a short example table and further discussion.

All values in some fields must be defined in Glossary or a referenced external data dictionary. Such fields are shown with cyan backgrounds below.

Every feature class and table has a primary key field with a name of the form <TableName_ID>. Where values of this primary key populate a field in another feature class or table, that field has a different name. For example, values of DataSources_ID populate fields named DefinitionSourceID (DescriptionOfMapUnits) and LocationSourceID (point data tables) and DataSourceID (many tables).

Example primary keys here are generated with a three letter prefix based on the name of the containing table and an integer numeric suffix that could be the string representation of the ObjectID included in all geodatabase-registered tables. As long as all table prefixes are unique in the dB, this will provide unique identification across the dataset, as well as some human intelligibility of the keys in foreign key relationships. If data loaded into this delivery database do not already have a user-managed primary key, this paradigm is suggested for generating primary keys.

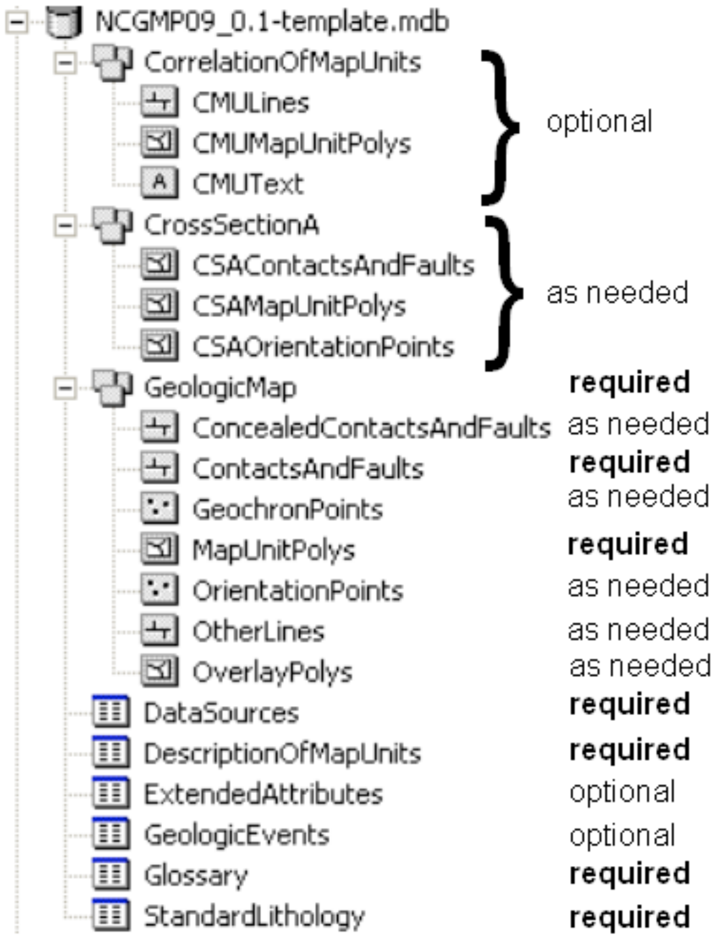

Figure 1. ArcCatalog view of NCGMP09-style geodatabase, showing required, as-needed, and optional database components. There may be more than one cross-section feature dataset, named CrossSectionA, CrossSectionB, …

GeologicMap (feature dataset, required)

Equivalent to the map graphic: contains all the geologic content (not the base map) within the neatline. All elements share a single spatial reference framework.

MapUnitPolys (polygon feature class, required)

Fields:

MapUnitPolys_ID *Primary key. Example Values = MUP1, MUP2, MUP3, etc. Values must be unique in database as a whole*

MapUnit *Short plain-text key (identifier) for the map unit. USGS tradition is no more than 4 characters. Example values: Qal, Tg, Kit, Trc3, etc. Foreign key to DescriptionOfMapUnits table. Null values not permitted—a mapped polygon must have an assigned map unit.*

IdentityConfidence *How confidently is this polygon identified as MapUnit? Value is usually 'std' (=standard) or 'low'. Null values not permitted. Suggest setting default value to 'std'*

Label *Calculated from MapUnit//Label and IdentityConfidence: if IdentityConfidence = low, append "?" to MapUnit//Label. Allows for subscripts and special characters.*

Null values OK

AreaSymbol *References an area fill symbol (background color + optional pattern). Area fill symbols, as style file or layer file, should be included in database package. Null values permitted*

Notes *Null values OK. Free text for additional information specific to this polygon*

DataSourceID *Foreign key to DataSources table, to track provenance of each data element. Null values not permitted*

Topology rules: polygons must not overlap. Polygons must not have gaps. Boundaries must be overlain by lines in ContactsAndFaults.

Note that open water (lakes, double-line rivers), glaciers, and unmapped areas are polygons, have non-null MapUnit values (perhaps wtr, glacier, unmapped). Water and glacier areas commonly are not labeled (Label=null).

ContactsAndFaults (line feature class, required)

Fields:

ContactsAndFaults_ID *Primary key for database record. Example values = COF1, COF2, … Values must be unique in database as a whole*

Type *Specifies the kind of feature represented by the line. Values = 'contact', 'fault', 'scratch boundary', 'waterline', 'glacier boundary', 'map boundary', etc… Values must be defined in Glossary or by reference to external glossary. Null values not permitted*

LocationConfidenceMeters *Half-width in meters of positional uncertainty envelope; position is relative to other features in database. Data type=float. Null values not permitted*

ExistenceConfidence *Values = 'std', 'low'. Null values not permitted. Suggest setting default value = 'std'*

IdentityConfidence *Values: 'std' , 'low' Null values not permitted. Suggest setting default value = 'std'*

LocationMethod *Short text string that specifies how a line feature was located. Domain is NGMDB LocationMethodTerms list. Value is 'not specified' if no other information is available.*

FgdcSymbol *References an FGDC geology symbol. Calculated from Type, LocationConfidenceMeters, ExistenceConfidence, and IdentityConfidence. Null values OK.*

Label *Typically blank, can be used to store fault name, or human-readable name for each line feature. To group line segments into a specific structure trace, e.g."San Andreas Fault", use Extended Attributes. Null values OK*

Notes *Null values OK. Free text for additional information specific to this feature*

DataSourceID *Foreign key to DataSources table, to track provenance of each data element. Null values not permitted*

Topology rules: Must not overlap. Must not self-overlap. Must not self-intersect. Must not have dangles, unless marked as exceptions. All dangling-line exceptions should be Type=fault (or one of its subtypes, e.g. thrust fault, or low-angle normal fault, or …).

Lines shown as "contact", "inferred contact" and "approximately located contact" are Type = 'contact', but have differing LocationConfidenceMeters, ExistenceConfidence, and (or) IdentityConfidence.

Calculation of symbol from Type, LocationConfidenceMeters, ExistenceConfidence, and IdentityConfidence is of the form

> **if** Type = ‘contact’ **and** LocationConfidenceMeters < ConfidenceZone(mapscale) **and** ExistenceConfidence = ‘std’ **and** IdentityConfidence = 'std'
>
> **then** Symbol = ‘FDGC-geology-XXXX__’

ConfidenceZone(mapscale) is the permissible uncertainty for an accurately-located line at a given scale. It might be calculated as

> ConfidenceZone(mapscale) = 0.001 meters * ScaleDenominator

E.g., for 1:24,000 scale, 24 meters; for 1:100,000 scale, 100 meters. Note that this calculation is specific to the scale of the visualization. If visualization scale changes the calculation must be repeated. The multiplier (0.001 meters, above) may vary from map to map and should be specified in the metadata for the dataset as a whole

OverlayPolys (polygon feature class, as needed)

Fields:

OverlayPolys_ID *Primary key. Values = OVP1, OVP2, OVP3, … Values must be unique in database as a whole*
MapUnit *Short plain-text key (identifier) to the overlay map unit. USGS tradition is no more than 4 characters. Qal, Tg, Kit, Trc3, etc. Foreign key to DescriptionOfMapUnits table. Null values not permitted*
IdentityConfidence *How confidently is this polygon identified as MapUnit? Value is usually 'std' or 'low'. Null values not permitted.*
Label *Calculated from MapUnit//Label and IdentityConfidence: if IdentityConfidence = Low, append “?” to MapUnit//Label. Allows for subscripts and special characters. Null values OK*
AreaSymbol *References an area fill symbol (background color + optional pattern). Area fill symbols, as style file or layer file, should be included in database package. Calculated from MapUnit. Null values OK*
Notes *Null values OK. Free text for additional information specific to this feature*
DataSourceID *Foreign key to DataSources table, to track provenance of each data element. Null values not permitted*

Topology rules: None.

Overlay polygon boundaries will typically have complex relationship with lines in ContactsAndFaults and ConcealedContactsAndFaults: in part coincident, in part not coincident. In general, overlay polygon boundaries will not be stroked.

ConcealedContactsAndFaults (line feature class, as needed)

Fields:

ConcealedContactsAndFaults_ID *Primary key. Values = CCF1, CCF2, CCF3, … Must be unique in database as a whole*
Type *Values = ‘concealed contact’, ‘concealed fault’, ‘'concealed thrust fault' … Values must be defined in Glossary or by reference to external glossary. Null values not*

permitted

LocationConfidenceMeters *Data type = float. Half width in meters of positional uncertainty envelope. Null values not permitted*

ExistenceConfidence *Values = 'std', 'low'. Null values not permitted. Suggest setting default value = 'std'*

IdentityConfidence *Values: 'std', 'low'. Null values not permitted. Suggest setting default value = 'std'*

LocationMethod *Short text string that specifies how a line feature was located. Domain is NGMDB LocationMethodTerms list. Default value is 'concealed beneath mapped cover'.*

FgdcSymbol *References an FGDC geology symbol. Calculated from Type, LocationConfidenceMeters, ExistenceConfidence, and IdentityConfidence. Null values OK*

Label *Typically blank, can be used to store fault name, or human-readable name for each line feature. To group line segments into a specific structure trace, e.g."San Andreas Fault", use Extended Attributes. Null values OK*

Notes *Null values OK. Free text for additional information specific to this feature.*

DataSourceID *Foreign key to DataSources table, to track provenance of each data element. Null values not permitted*

Topology rules: Must not overlap. Must not self-overlap. Must not self-intersect. All dangling nodes, unless marked as exceptions, should be coincident with nodes in ContactsAndFaults.

OtherLines (line feature class, as needed)

Fields:

OtherLines_ID *Primary key. Values = OTL1, OTL2, OTL3, … Values must be unique in database as a whole*

Type *Values='cross-section line', 'syncline hinge surface trace', 'biotite isograd', … Values must be defined in glossary or by reference to external glossary. Null values not permitted*

LocationConfidenceMeters *Data type = float. Half width in meters of positional uncertainty envelope. Null values not permitted.*

ExistenceConfidence *Values = 'std', 'low'. Null values not permitted. Suggest setting default value = 'std'*

IdentityConfidence *Values: 'std', 'low'. Null values not permitted. Suggest setting default value = 'std'*

LocationMethod *Short text string that specifies how a line feature was located. Domain is NGMDB LocationMethodTerms list. Value is 'not specified' if no other information is available. Concealed features in this feature class must specify 'inferred beneath covering mapped unit' for use in symbolization rules.*

FgdcSymbol *References an FGDC geology symbol. Calculated from Type, LocationConfidenceMeters, ExistenceConfidence, IdentityConfidence, and LocationMethod.*

Label *Typically blank, can be used to store fault name, or human-readable name for each line feature. To group line segments into a specific structure trace, e.g."San Andreas Fault", use Extended Attributes. Null values OK*

Notes *Null values OK. Free text for additional information specific to this feature*

DataSourceID *Foreign key to DataSources table, to track provenance of each data element. Null values not permitted*

Topology rules: Must not self-overlap. Must not self-intersect.

'Hinge surface trace', 'approximately located hinge surface trace', and 'inferred hinge surface trace' are all Type = 'hinge surface trace' but have differing LocationConfidenceMeters, ExistenceConfidence, and (or) IdentityConfidence. Note that concealed dikes, marker beds, veins, hinge surface traces, etc., are included in this feature class. Assignment of a dotted line symbol in these cases should be based on the value of LocationMethod.

Dangles, overlays, concealed contacts, and topology: what goes where?

By convention, a geologic map depicts the distribution of earth materials on a particular map horizon. At every location there is one map unit and there is only one map unit at any location. Another way to state this is that map units are opaque: one cannot see through them. Map unit polygons are bounded by contacts, faults, shorelines, snowfield boundaries, scratch boundaries, or the map boundary. With some exceptions (which are unusual enough to require mention), map-unit polygons are not interrupted by contacts—equivalently, contacts do not separate polygons of the same map-unit. Map-unit polygons may be interrupted by faults.

Often we show underlying material, or overlying material, or some additional aspect of earth materials (dike swarm, alteration zone, etc.) with an overlay. On a map graphic, such an overlay is commonly shown by a pattern—diagonal lines, scattered red dots, or other—overprinted on the map-unit color and (optional) map-unit pattern. The topological relations of these overlays are likely to be complicated (e.g., alteration area boundary does not coincide with bedrock map-unit boundaries, but does coincide with unconsolidated-deposit boundaries) and not easily prescribed by a simple set of rules. The edges of overlay polygons are often not shown with a line (not stroked).

Dikes, veins, cross-section lines, hinge-surface traces, isograds, ice-sheet margins, structure contours, and so on typically will be stored in OtherLines. All of these lines share the property that they do not participate in map-unit topology.

We understand that some producers of geodatabases will choose to create polygons and edit linework in the absence of a topology relationship class, as they exist within the ESRI geodatabase framework. For instance, rather than using topology editing tools to synchronously edit shared boundaries between lines and polygons, many users prefer to edit through a procedure involving lines, polygon attribute label points, and the creation of polygons only when the linework is finished, without the use of geodatabase topology rules. For the purposes of this schema, i.e. data delivery, the method used to produce the feature classes does not matter, only that the feature classes in the database follow the topology rules outlined above.

About point data

Observations of structure orientations, mineral occurrences, or geochemical, paleontologic, geochronologic and other kinds of sample analyses, etc. are made at field stations or on samples collected at field stations. There are two modes for representing such observations and their accompanying locations: (1) a normalized mode, in which a Stations feature class stores location information, a Sample table stores information on samples related to stations, and other non-spatial tables store observations and analyses, one for each observation or analysis type, related to either a sample or station; and (2) a denormalized mode, in which there is a separate feature class for each type of observation or analysis that also—and in some cases repeatedly—stores station location and sample information. Each mode has advantages. The first allows error-resistant editing of location and sample information and is best suited for a data management and archiving system. The second allows slightly easier symbolization and makes it easier to separate analytical information from the geodatabase by simply copying the relevant feature class.

For this data delivery database schema we endorse the second mode. We note that to create a compliant database it is likely to be useful to start in the first mode, creating a Stations point feature class with related sample and non-spatial data tables, and from these create the appropriate data-type-specific point feature classes that will be included in the delivery database.

Below, we describe two possible point feature classes, one for measurements made directly at a station, and one for measurements related to a sample collected at a station. Neither is required, though both are likely to be needed for many maps. Other point feature classes (e.g., GeochemPoints, PhotoPoints, FieldNotePoints, SamplePoints, and FossilPoints) should be created as needed, following the patterns outlined here. Each point feature class shall contain the following fields:

TableName_ID *Primary Key. Substitute actual table name for 'TableName'. Null value not permitted.*
Type *Values must be defined in Glossary or by reference to external glossary. Null values not permitted*
StationID *Foreign key to Stations point feature class. Null values OK*
MapUnit *One commonly would like to know what map unit an analysis or observation pertains to. Foreign key to DescriptionOfMapUnits. Null values not permitted*
FgdcSymbol *References an FGDC geology symbol. Null values OK*
Label *What text should accompany the symbolization? Null values OK*
LocationConfidenceMeters *Radius in meters of positional uncertainty envelope. How well located is the observation or sample locale? Null values not permitted*
PlotAtScale *Data type = float. At what scale (or larger) should this observation or analysis be plotted? At smaller scales, it should not be plotted. Useful to prevent crowding of display at small scales and to display progressively more data at larger and larger scales. Value is scale denominator. Null values not permitted.*
Notes *Null values OK. Free text for additional information specific to this feature*
LocationSourceID *Foreign key to DataSources. Identifies source of point location. Null values not permitted*
DataSourceID *Foreign key to DataSources. Identifies source of data at this point. Null values not permitted*

All sample-oriented point feature classes shall also have fields:

FieldSampleID *Sample ID given at time of collection. Null values OK*
AlternateSampleID *Museum #, lab #, etc. Null values OK*
MaterialAnalyzed *Null values OK*

OrientationPoints (point feature class, as needed)

Point structure data (bedding attitudes, foliation attitudes, slip vectors measured at a point, etc.) shall be stored in OrientationPoints, one point per measurement. Data creators must ensure that multiple measurements at a single station (e.g., bedding and cleavage) have coincident locations. ExtendedAttributes relationships may be necessary to represent relationships between measurements (e.g. lineation in foliation, intersection lineation to intersecting foliations). OrientationPoints should have the generic point-data fields listed above, as well as

OrientationPoints_ID *Primary Key. Example values = ORP1, ORP2, ORP3, … Null values not permitted*
Azimuth *Data type=float. Values limited to range 0..360. Strike or trend, measured in degrees clockwise from geographic North. Use right-hand rule (dip is to right of azimuth direction). Horizontal planar features may have any azimuth. Null values not permitted.*
Inclination *Data type=float. Values limited to range -90..90. Dip or plunge, measured in degrees down from horizontal. Negative values allowed when specifying vectors (not axes) that point above the horizon, e.g., paleocurrents. Types defined as horizontal (e.g., horizontal bedding) shall have Inclination=0. Null values not allowed.*
IdentityConfidence *Values = 'std', 'low'. Specifies confidence that observed structure is of the type specified. Null values not permitted*
OrientationConfidenceDegrees *Data type=float. Circular error, at 95% confidence level, in degrees. For planar features, error in orientation of pole to plane. Null values not permitted*

The Type field identifies the orientation type (bedding, overturned bedding, stretching lineation, open joint, etc.). Type definitions shall record the orientation-measurement convention for the Type (strike & dip or trend & plunge).

GeochronPoints (point feature class, as needed)

Point-data fields as defined above, plus:

GeochronPoints_ID *Primary Key. Values = GCR1, GCR2, GCR3, … Null values not permitted*
Age *Data type=float. Appropriate value is the interpreted (preferred) age calculated from geochronological analysis, not necessarily the date calculated from a single set of measurements. Null values not permitted*
AgePlusError *Data type=float. Record type of error (RMSE, 1 sigma, 2 sigma, 95% confidence limit) in Notes field. Null values OK*
AgeMinusError *Data type=float. Record type of error (RMSE, 1 sigma, 2 sigma, 95% confidence limit) in Notes field. Null values OK*
AgeUnits *Values = years, Ma, ka, radiocarbon ka, calibrated ka, … Units defined in glossary or by reference to published vocabulary. Null values not permitted*
FieldSampleID *Null values OK*
AlternateSampleID *Null values OK*
MaterialAnalyzed *Null values OK*

Use the Type field to identify the geochronological method (K-Ar, radiocarbon, whole-rock RB-Sr isochron, etc.). Analytical data can be represented using ExtendedAttributes, or in an analysis-specific table such as KArPoints if there is much data of a single analysis type.

Cross Sections (feature datasets, as needed)

Cross sections should be identified as cross-section A, cross-section B, cross-section C, and so on, abbreviated as CSA, CSB, CSC in the dataset and feature class names. Each cross-section exists in a separate map-space, and thus we dictate a separate feature dataset for each cross-section. For each cross section there are, at a minimum, two feature classes:

CSAContactsAndFaults *(primary key is CSAContactsAndFaults_ID, values = CSACOF1, CSACOF2, …)*
CSAMapUnitPolys *(primary key is CSAMapUnitPolys_ID, values = CSAMUP1, CSAMUP2, …)*

Field names, data types, usage, and topology rules for these feature classes are identical with those for ContactsAndFaults and MapUnitPolys. If lines that don't participate in MapUnit topology or point-based data are depicted on the cross-section, the appropriate feature classes (e.g., CSAOtherLines, CSAOrientationPoints) should be created.

Correlation of Map Units (feature dataset, optional)

The Correlation of Map Units (CMU) diagram found on many geologic maps can be encoded as a feature dataset in a geodatabase. Doing so makes it easier to have symbolization of the CMU match symbolization of the map and stores the information in the CMU in a fashion that is (slightly) more queryable than storing the CMU as a dumb graphic. Two feature classes are necessary and a third (CMUText) will almost always be needed. If map units are depicted as point features an additional feature class is needed.

CMUMapUnitPolys (polygon feature class)

Fields:

CMUMUP_ID	*Primary key. Example values - CMUMUP1, CMUMUP2, CMUMUP3, … Null values not permitted*
MapUnit	*Foreign key to DescriptionOfMapUnits. Null values not permitted*
Label	*Value = MapUnit//Label. Null values OK*
AreaSymbol	*Value = MapUnit//Symbol. Null values OK*

Values for Symbol and Label are calculated with reference to DescriptionOfMapUnits. Ghost boxes (e.g., protolith of a metamorphic unit) may be shown as MapUnitPolys with Symbol = null. Or the box outline alone can be stored in CMULines.

CMULines (line feature class)

Fields:

CMULIN_ID	*Values are CMULIN1, CMULIN2, CMULIN3, … Null values not permitted*
Type	*Term to classify meaning of lines. Values include Contact, GhostContact, CmuLeader, CmuRule, CmuBracket, or <MapUnit>_line. Values must be defined in Glossary. Null values not permitted*
FgdcSymbol	*Identifier for symbol to use for line portrayal. Null values OK*

CMUText (annotation feature class, as needed)

Fields:

CMUTEX_ID	*Primary key. Example values - CMUTEX1, CMUTEX2, CMUTEX3, … Null values not permitted*

ParagraphStyle *Null values not permitted*

Annotation text and annotation attributes, including font, font size, font effects, and text angle, are stored in default fields of the annotation feature class. Values for font, font size, and font effects can be calculated from ParagraphStyle.

CMUPoints (point feature class, as needed)

Fields:

CMUPNT_ID	*Primary key, example values - CMUPNT1, CMUPNT2, CMUPNT3, … Null values not permitted*
Type	*Values are <MapUnit>_point. Values must be defined in Glossary. Null values not permitted*
Symbol	*Null values OK*

Non-spatial tables

DescriptionOfMapUnits (non-spatial table, required)

This table captures the content of the Description of Map Units (or equivalent List of Map Units and associated pamphlet text) included in a geologic map published by the USGS and state geological surveys.

Fields:

DescriptionOfMapUnits_ID	*Primary key: DMU1, DMU2, DMU3; ExtendedAttributes table OwnerID is a foreign key using this value. Null values not permitted*
MapUnit	*Short ASCII string that identifies map unit: Qal, Tec, Qvt; Triassic Newark Formation can be Trn or TRn, your choice! Unit abbreviations must be unique in the database. Null values OK, and are commonly associated with headings or headnotes. Use of special characters is not recommended in this field.*
Label	*Text string used to place label in map display; includes graphic elements such as special fonts and formatting for subscripts. For example, Triassic Newark Formation might be "<font=SpecialAgeFont>#n". Null values OK for units that do not appear on map or are not labeled, e.g. headings, headnotes, water, glacier, some overlay units.*
Name	*Boldface name in traditional DMU, identifies the unit within its hierarchical context. Examples: 'Escabrosa Formation', 'Beckers Butte Member'. Null values OK*
FullName	*Full name of unit, may include identification of containing higher rank units, e.g. 'Beckers Butte Member of Martin Formation,' 'Prichard Formation of Ravalli Group of Belt Supergroup'. Text you would like to see as fly-out when cursor lingers over polygon in an electronic map display. Null values OK (headings, headnotes, geologic units not shown on map)*
Age	*As shown in bold within parentheses in traditional DMU. Null values may be used for map units that inherit Age from a parent unit, or for headings, headnotes, or overlay units. To designate age with more resolution than permitted by DMU standards, or to record multiple ages (e.g., deposition and metamorphism) for a unit, create entries in ExtendedAttributes and GeologicEvent tables*

Description *Free-format text description of map unit. Commonly ordered (lithology, thickness, color, weathering and outcrop characteristics, distinguishing features, genesis, age constraints) and terse. Allows markup (e.g. HTML) specification of new paragraphs, super- and subscripts, and geologic-age font (sans-serif and with special characters). Null values OK*

HierarchyKey *Has form nn-nn-nn, nnn-nnn, or similar. Numeric, left-padded with zeros, dash-delimited. Each HierarchyKey fragment of each row MUST be the same length to allow text-based sorting of the DMU entries. These strings are useful for resolving queries involving hierarchical relationships, e.g. 'find all members of formation x', 'what is the parent unit of map unit y'. Null values not permitted*

ParagraphStyle *Domain is Heading1st, Heading2nd, Heading3rd, …, Headnote, DMU1, DMU2, DMU3, … Formatting associated with a style should be explained with a definition of the style in the glossary. Null values not permitted*

AreaFillRGB {*Red, Green, Blue} tuples that specify the color (e.g., '255,255,255' for white) of area fill for symbolizing the unit. Null values OK (headings, headnotes)*

AreaFillPatternDescription *Text description (e.g.,' random small red dashes') provided as a convenience for users who must recreate symbolization. Null values OK (headings, headnotes, unpatterned map units)*

DefinitionSourceID *Foreign key to DataSources. Identifies source of DescriptionOfMapUnits entry. Null values not permitted*

The traditional Description of Map Units (DMU), or equivalent List of Map Units with descriptions in an accompanying pamphlet, is strongly formatted and typically hierarchical. The hierarchy can carry a significant amount of information. This table encodes the traditional DMU as specified in Suggestions to Authors (Bishop et al., 1978, p. 137-140; Hansen, 1991, p. 49-52) without loss of information and without imposing additional structure or content. It is not designed to produce uniform, easily-queried earth-materials descriptions; for this, see StandardLithology below.

There are entries in this table for all map units and overlay units assigned to polygons on the map (or in any of the cross sections), and for all headings and headnotes beneath "DESCRIPTION OF MAP UNITS" (or under LIST OF MAP UNITS). The entries should include map units that are traditionally not listed in the DMU/LMU such as 'water', 'glacier', and 'unmapped area', and all geologic units that are listed in the DMU/LMU as parent units but are not represented as polygons on the map.

The text of headings and headnotes should be stored in the Description field. Heading and headnote text should have initial capitalization only and no font specifications—these are given by ParagraphStyle.

The ParagraphStyle field eases automatic construction of a traditional text DMU or LMU from DescriptionOfMapUnits. ParagraphStyle values can, with difficulty, be calculated from HierarchyKey, Description text, and MapUnitPolys. The partial redundancy between HierarchyKey and ParagraphStyle allows some automated checking of DescriptionOfMapUnits for logical consistency.

DescriptionSourceID commonly points to Source = 'This report' or Source = 'Modified from <earlier report>'.

StandardLithology (non-spatial table, required)

This table represents the lithologic composition of map units by associating with the unit one or more lithology categories from a NGMDB controlled vocabulary. Each associated lithology category has a part type that indicates how the rock type occurs within the unit (veins, layers, stratigraphic part, interbedded, inclusions, blocks…) and a proportion (either a qualitative term or numeric value).

Fields:

StandardLithology_ID *Primary key. Example values = STL1, STL2, STL3, … Null values not permitted*
MapUnit *Unit abbreviation, foreign key to DescriptionOfMapUnits. Null values not permitted*
PartType *Domain is NGMDB StandardLithologyPartType list. Null values not permitted*
Lithology *Domain is NGMDB StandardLithology list. Null values not permitted*
ProportionTerm *Domain is NGMDB StandardLithologyProportion list*
ProportionValue *Data type = float. Range 0..1.0. Must not sum to more than 1.0 for a given MapUnit*
ScientificConfidence *Values = 'std', 'low'. Default value = 'std'. Value of 'low' indicates either that the assignment of the constituent to a lithology category from the controlled vocabulary is problematic, or that the proportion is poorly constrained. Null values not permitted*
DataSourceID *Foreign key to DataSources. Identifies source of StandardLithology description. Null values not permitted*

Standard-Lithology_ID	MapUnit	PartType	Lithology	ProportionTerm	ProportionValue
STL26	Tx	Interbedded	Sandstone	Dominant	
STL327	Tx	Stratigraphic part	Siltstone	Minor	
STL579	Tx	Stratigraphic part	Tuff	Minor	
STL264	Txt	Interbedded	Tuff	Dominant	
STL265	Kit	Only part	Tonalite	Dominant	
STL266	KJz	Interbedded	Limestone		.55
STL770	KJz	Interbedded	Mudstone		.45

Table 1. Partial examples of StandardLithology data. Field names are at the top of each column, and each row represents a separate data instance. Numeric proportions are fractional values between 0 and 1.

StandardLithology provides a simple structure for describing all map units in terms of a limited number of lithology categories. Use it in parallel with DescriptionOfMapUnits, which allows for unstructured free text descriptions, and ExtendedAttributes which permits open-ended structured descriptions.

Use ProportionTerm or ProportionValue as appropriate. Both may not be null in a single record.

If you generate StandardLithology records by interpreting map unit descriptions in an existing map or database, set DataSources_ID to point to Source = 'Smith and others, USGS Map I-37, interpreted by <your-name>' or similar.

DataSources (non-spatial table, required)

Fields:

DataSources_ID *Primary key. Example values = DAS1, DAS2, DAS3, … Null values not permitted*
Source *Plain-text description of data source. By convention, for DataSources_ID = DAS1,*

Source = 'This report'. Null values not permitted

Notes *Can include full citation and (or) URL. Null values OK*

DataSources_ID	Source	Notes
DAS1	This report	Field compilation automated by A. Digitdroid, using georeferenced scan of green-line mylar, ESRI ArcScan tools, and manual editing
DAS2	This report: Ralph Haugerud field data, 2003	
DAS3	USGS Open-file Report 2004-197	
DAS4	C. A. Hopson, University of California-Santa Barbara, written communication 17 July 2005, scale 1:24,000	Sketch map of lower Chelan creek, used for tonalite phase - gabbro phase contact.
DAS5	Beta Laboratories, Report 1999-451.	
DAS6	Jackson, J.A., 1997, Glossary of Geology: Alexandria, VA, American Geologic Institute, 657 p.	Cited in Glossary table for sources of term definitions
DAS7	Modified from DAS3	Digitized 3 new large landslides.

All features and table entries need to be associated with a data source. For maps that contain all new information and use a single vocabulary source, this table will be very short. For compilations, with data from many sources which have been edited and (or) reinterpreted so that the data source has effectively been changed, this table becomes longer and more useful. See ChangeLog (below) for advice on maintaining accurate DataSources_ID values.

Glossary (non-spatial table, required)

Fields:

Glossary_ID *Primary Key. Example values = GLO1, GLO2, GLO3, … Null values not permitted*

Term *Natural language word for a concept. Values must be unique within database as a whole. Example values: granite, foliation, syncline axis, contact, thrust fault, std, low, fission track, K-Ar. Null values not permitted*

Definition *Plain-English definition of Term. Null values not permitted*

DefinitionSourceID *Foreign key to DataSources. Identifies source of Definition. Null values not permitted*

Glossary_ID	Term	Definition	DefinitionSourceID
GL001	contact	Line denoting genetic boundary (depositional, intrusive, metamorphic, ...) between two geologic map units	DAS6
GL002	Biotite isograd	Line marking 1st appearance, going up-grade, of newly-formed biotite in metamorphosed siltstones and shales	RichardDGM18

Terminology used in the database must be defined in this Glossary or in a referenced external glossary. Terms that require definition include all values of Type; all non-numeric ScientificConfidence values; Property names, non-numeric PropertyValue terms, and Qualifiers for ExtendedAttributes; Lithology and ProportionTerm in StandardLithology, and some other terms. Lithology terms used in StandardLithology *must not be redefined* from the NCGMP standard. If there are no intellectual property restrictions, it is permissible to replicate all or part of an external glossary here. Be sure to provide appropriate credit via the DefinitionSourceID. Values of Term must be unique within the database because they are used in fields in other tables where they function as foreign keys to the Glossary table.

There shall be a clear statement in report-level metadata that all terms not defined in Glossary are defined in external glossaries e.g. the AGI Glossary of Geology (Neuendorf et al., 2005), or Webster's Dictionary. This typically will be accompanied (preceded) by statements like "Igneous rock nomenclature follows Streckeisen (1976)" or "Numerical ages of geologic time periods after Ogg et al., (2008)."

We expect that building Glossary tables for the first few reports produced by a workgroup will be a significant effort. Subsequent Glossaries should be much easier, as a prior Glossary can be recycled (with updated DefinitionSourceIDs) with minor amendments.

ExtendedAttributes (non-spatial table, optional)

Fields:

ExtendedAttributes_ID	*Primary key. Example values = EXA1, EXA2, EXA3, … Null values not permitted*
OwnerTable	*Full name of table that contains owning element, e.g., DescriptionOfMapUnits, or OverlayPolys. May be any table in the database. Null values not permitted*
OwnerID	*Foreign key to table specified by the OwnerTable value. If Owner_ID record is deleted, associated extended attribute should be deleted (cascade delete). Null values not permitted. Convention is that this Foreign key will link to the TableName_ID field in the OwnerTable.*
Property	*Name of property specified by this attribute or relationship between Owner and ValueLink items. Values defined in Glossary or external glossary; we strongly recommend Glossary definitions of all properties used in the ExtendedAttributes table. Definition of property should include explanation of formatting and units used to specify property values. Null values not permitted*
PropertyValue	*string, could be number (+ measurement unit) or defined term. Not closed. Data-entry tool might enforce consistency between PropertyValue and Property (such that Property=thickness does not have PropertyValue=fine-grained). NGMDB or individual projects might choose to supply Property | PropertyValues lists. Numeric values (for instance, 500 meters) are not defined in Glossary*
ValueLink	*Foreign key to data instance that specifies property value. E.g., GrainSizeAnalyses_3. Or a link to another ExtendedAttributes record (e.g., this thing overlies / succeeds / is-a-part-of another thing). Null values OK. If null, PropertyValue must be non-null, and vice-versa. Definition of Property must specify table to which the ValueLink is a foreign key*
Qualifier	*Expresses variability or extent of PropertyValue. Must be defined in Glossary or an external glossary. Null values OK*
Notes	*Null values OK*
DataSourceID	*Foreign key to DataSources table, to track provenance of each data element. Null values not permitted*

Extended-Attribute_ID	Owner-Table	Owner ID	Property	Property-Value	Value-Link	Qualifier	Notes	DataSourceID
	DescriptionOfMapUnits	DMU3	Permeability	Low		Typical	Rock is full of alteration clays	DS2140
	DescriptionOfMapUnits	DMU3	Permeability	High		Rare		DS0001
	DescriptionOfMapUnits	DMU27	MetamorphicGrade	Low		Uncommon		DS0364
	DescriptionOfMapUnits	DMU27	MetamorphicGrade	Medium		Typical		DS2069
	DescriptionOfMapUnits	DMU27	MetamorphicAge	Early Proterozoic		Probable		DS2106
	DescriptionOfMapUnits	DMU27	MetamorphicAge	Middle Cretaceous		Possible		DS045
	Geologic-Events	Slip-Event1	Displacement	4 km				DS1045
	Geologic-Events	Slip-Event1	DisplacementType	Right-lateral strike slip				DS1130
	Geologic-Events	Slip-Event1	Successor		GE2466			DS1205
	Geologic-Events	GE2466	Displacement	200 km				DS1135
	Geologic-Events	GE2466	DisplacementType	Right-lateral strike slip				DS0980
	DescriptionOfMapUnits	Txt	Permeability	Low			Rock is full of alteration clays	DS8625
	MapUnit-Polys	Txt37a	Note				Big outcrop, good place for a quarry	DS2586
	Contacts-AndFaults	COF22	HasPhotograph	Photo2008-11-12b				DS2640
	Contacts-AndFaults	COF22	ContactCharacter	Gradational				DS3656

Note that all OwnerIDs are table_IDs; map units are referenced by DMU_ID, not MapUnit. This contrasts with use of MapUnit as foreign key to the DescriptionOfMapUnits table in other places; the alternate convention is adopted here for consistence with references from ExtendedAttributes to other database tables.

This table provides a general structure for linking attributes of any sort with any feature in the database. Analytical data can be stored here, or stored in an existing feature class (in some cases), a special feature class, or a non-spatial table (in some cases). We anticipate that best practice recommendations will emerge for particular kinds of data.

Use of ExtendedAttribute table needs some careful consideration. If normalized data are to be recoverable from this data structure, then each of these extended attributes must be a single fact.

For example to represent a slip displacement event in a sequence of displacements on a complex fault or fault segment: “SlipEvent1Displacement; 4 km right lateral strike slip” is several facts: 1. SanAndreasFault has GeologicEvent xxxx with EventTerm=’fault slip’. 2. GeologicEvent xxxx hasSuccessor = GeologicEvent yyyy (if there is a slipEvent2). 3. GeologicEvent xxxx has ExtendedAttribute displacementMagnitude_m = 4000. 4. GeologicEvent xxxx has ExtendedAttribute displacementType = ‘Right Lateral Strike Slip’. The GeologicEvent xxxx age value is the time bracket for the slip event.

The OwnerID in ExtendedAttributes is a foreign key that may link to a data instance in any table, e.g. DescriptionOfMapUnits, Glossary (for named faults that appear as ‘superset’ in ContactsAndFaults feature class), some mapped feature for description of individual poly, or GeologicEvent to describe a displacement event (if logic above is followed) or add additional process and environments associated with an event. The OwnerTable attribute is the name of the table that OwnerID references. The same issue applies to use of the ValueLink property, but in this case the definition of the ExtendedAttribute property must specify the table that contains the linked values.

ValueLink allows links to data elements in other tables as values for attributes. Having a pointer value to specify a property opens the door for use of the extended attributes table to represent any kind of semantic relationships between data in the database. Such relationships could include for example the association of a lineation and foliation in a compound fabric, or multiple bedding measurements associated with a derived fold hinge orientation. The ExtendedAttribute Property in this case specifies a relationship type.

Inclusion of explicit StandardLithology and GeologicEvent tables represents a design trade-off between the flexibility of the ExtendedAttribute pattern for attribute assignment, and the clarity of including information in separate, explicitly defined tables. We judged that the lithologic composition and geologic history are especially significant and common to most maps, and thus merit explicit table entities (StandardLithology, GeologicEvent).

GeologicEvents (non-spatial table, optional)

Geologic ages are assigned by association with an event that is recorded in the rock record. Each event has an assigned age, specified either numerically or using a named era from a stratigraphic time scale.

Fields:

GeologicEvents_ID *Primary key for event in this database. Example values = GEE1, GEE2, GEE3, ... Required*

Event *This is the geologic process responsible for the observed, dateable feature in the rock record that is the basis for the age assignment. Example values: deposition, metamorphism, slipEvent1, etc. Required. Foreign key to Glossary or vocabulary authority cited in dataset metadata.*

AgeDisplay *Formatted text that conveys the age assignment to a human reader, analogous to the Age attribute in the DMU table. Required*

AgeYoungerTerm *Younger bound of interval for age of geologic event. Specified by a named time ordinal era from a stratigraphic time scale that is specified in the dataset metadata. Required if no numeric age provided*

AgeOlderTerm *Older bound of interval for age of geologic event. Specified by a named time ordinal era from a stratigraphic time scale that is specified in the dataset metadata. Required if no numeric age provided*

TimeScale *Name of a geologic time scale in which the age terms are defined. Various time scales may be used in a single data set, e.g. ICS 2008, North American Land Mammal Stages 2005. Required if age terms are used*

AgeYoungerValue *Data type = float. Number that specifies the younger bound of the interval for the age assignment. Use of numeric age range boundaries makes for simpler geologic age query resolution. Units used for numeric age assignment should be consistent within the database and the units should be specified in the Notes field.*

Required if no age term provided

AgeOlderValue *Data type = float. Number that specifies the older bound of the interval for the age assignment. Use of numeric age range boundaries makes for simpler geologic age query resolution. Units used for numeric age assignment should be consistent within the database and the units should be specified in the Notes field. Required if no age term provided*

Notes *Free text, any additional information on this event or age assignment. Null values OK*

DataSourceID *Foreign key to DataSources table, to track provenance of each data element. Null values not permitted*

Geologic-Events_-ID	Event	Age-Display	Age-Younger-Term	Age-Older-Term	Age-Younger-Value	Age-Older-Value	Notes	Data-SourcesID
GE00001	FaultSlip	Early Miocene	Early Miocene	Early Miocene	20	22		DS26904
GE00022	FaultSlip	Pliocene to Quaternary	Quaternary	Pliocene	0	4		DS62016
GE2465	Deposition of Tvt	Miocene Deposition	Miocene	Miocene	8	22		DS105
GE23609	Laramide orogeny	Laramide age	Early Eocene	Cenoma nian	40	80		DS20656

The GeologicEvents table allows explicit representation of complex histories and non-simple ages. Geologic events may be associated with multiple processes and environments (e.g. depositional environments) through extended attributes. This content is required for compatibility with GeoSciML. AgeYoungerValue and AgeOlderValue are numeric and represent ranges or bounds on the 2-sigma uncertainty envelop on a measured numeric age, unless otherwise specified in the Notes field for the age.

There are four ways to represent an event in the history of a map unit: (1) the age field of DMU table, by convention this field has limited age resolution and can only represent the dominant event in the history of the unit; (2) in the Description field of table DescriptionOfMapUnits; (3) in the table ExtendedAttributes (property=MinimumAge, propertyValue=Maastrichtian); (4) this GeologicEvents table, with link via ExtendedAttributes table (property = preferredAge, ValueLink = GEE13). For ages of other features (e.g., faults, single map-unit polygons) methods 3 and 4 are applicable, as is recording the age in the Notes field of the appropriate record(s) of the relevant spatial feature class.

We provide multiple options to record geologic ages because (a) we're not sure which option is best (and hope that in a short time best practice recommendations will emerge), and (b) we think it is likely that the best option depends on the quality and quantity of age information to be recorded.

Shapefile versions of the database

We suggest that two optional shapefile versions of the database be provided: a simple version, designed to permit ready symbolization and query without need to establish relates or joins to non-spatial tables, and without all the content of the full database; and an open version that supplies as much of the database content as possible in an open file format.

Simple version

At a minimum, the simple shapefile version of the database should include MapUnitPolys and ContactsAndFaults shapefiles. OverlayPolys, ConcealedContactsAndFaults, OtherLines, and various point-feature shapefiles are possible additions. Attribute data is included with every shape record, thus there are no related tables or joins required to browse the data.

To create the MapUnitPolys shapefile, join DescriptionOfMapUnits and DataSources tables to the MapUnitPolys feature class and export to a polygon shapefile. Map long field names from the geodatabase to short, DBF-compatible names as prescribed in Appendix B. Certain fields (e.g., Text field in DescriptionOfMapUnits) are likely to be truncated to fit the 255-character limit for DBF fields; this is acceptable. Create a new field, StdLith, and populate it with values created by concatenating the appropriate StandardLithology records using the protocol $PartType_1$:$Lithology_1$:$Proportion_1$; $PartType_2$$Lithology_2$:$Proportion_2$; … where subscripts refer to successive StandardLithology records for a map unit and records are ordered from largest to smallest proportion, following the proportion rankings given in Appendix _. Drop the DataSources_ID field. If the geodatabase contains an OverlayPolys feature class, translate it to an OverlayPoly shapefile following the same procedure.

To create the ContactsAndFaults shapefile, join DataSources table to the ContactsAndFaults feature class and export to a line shapefile. Map long field names from the geodatabase to short, DBF-compatible names as prescribed in Appendix B. Drop the DataSources_ID field. If the geodatabase contains ConcealedContactsAndFaults or OtherLines feature classes, translate them to shapefiles following the same procedure.

Generate point shapefiles for each point feature class included in the geodatabase. Again, map long field names from the geodatabase to short, DBF-compatible field names as prescribed in Appendix B. Join the DataSources table to the Glossary table and export the extended Glossary as a tab-delimited text file. The DataSources_ID field may be dropped..

Open version

The open shapefile version of the geodatabase consists of shapefile and dbf equivalents of all feature classes and non-spatial tables. Export each to a shapefile or dbf table as appropriate, mapping long field names to short field names as specified in Appendix B. Values of some fields (particularly in the Text field of DescriptionOfMapUnits, the Notes field of DataSources, and Definition field of Glossary) are likely to be truncated. For this reason, we recommend also exporting these tables as comma-delimited text.

Export of geodatabase content as a collection of shapefiles and dBase (or tab-delimited text) files is relatively straightforward. Long field names should be mapped to the 10 character names as prescribed in Appendix B. This can be done by loading the tables into an ArcMap project and assigning the dBase compatible names as aliases before exporting from ArcMap. Alternatively, the export tool in ArcCatalog (right click on the table in the table of contents and select ‘Export’) allows field names to be modified in the tool dialog. To capture long text fields, the data will need to be exported to tab-delimited text files. This must be done in ArcMap, by opening the table, click the ‘Options’ button, select ‘Export’, click the ‘Browse’ button for the location to save the file, and in the ‘File Save’ dialog, click the ‘Save as’ type dropdown and select text as the export format.

In the long run, we recommend that an application-independent, open interchange file format be adopted as an alternate data delivery mechanism. The IUGS Commission for Management and Application of Geoscience Information (CGI) has been supporting development of an xml-based markup for geoscience information interchange (GeoSciML, http://www.geosciml.org/), which has the potential to be this format. GeoSciML is now in version 2, and has been tested in 3 international test bed exercises. Open Geospatial Constorium (OGC) Web Feature Service (WFS) data delivery using GeoSciML is part of the OneGeology level 2 service profile (http://onegeology.org/technical_progress/technical.html#specs_and_requirements). As part of the Geoscience Information Network project, the Arizona Geological Survey is developing a mapping to and from GeoSciML of the geodatabase schema defined in this report.

Building a compliant database

Note to readers: The following section is, of necessity, incomplete pending finalization of the database schema. When the schema is finalized, we expect to flesh this section out with further advice on how to construct compliant databases.

Empty compliant databases into which data can be imported or created can be built from scratch using the specification in this document, by running script ncgmp09_create.py, by copying an empty geodatabase template, or by importing an ESRI XML-workspace file. From within ArcMap, the Extract Data Wizard on the Distributed Geodatabase toolbar can be used to create an empty geodatabase with the data model schema but different spatial references. XML schema files can be imported into empty geodatabases in ArcCatalog, but the process does not allow for the spatial references of any objects to be changed. That must be done by hand before the import and is tedious and syntactically difficult. XML schema files will, perhaps, be most useful for schema validation. The National Park Service has generated some tools to facilitate this process, which may be adoptable for our purposes.

We imagine that the most common objects (tables or feature classes) included in addition to those specifically described in this document will be point feature class tables. See 'About point data' above for the required fields. The design of all other fields is at the discretion of the producer.

The production of a compliant database should be assisted by a number of custom tools and scripts. For example, we imagine tools to automate the population of the ChangeLog table, calculate symbol field values (line symbols, for instance, reflect values in the Type, LocationConfidenceMeters, and ExistenceConfidence, and IdentityConfidence fields as well as the output map scale), and export geodatabase objects to a shapefile version. Script ncgmp09_validate.py checks the names of feature data sets, feature classes, tables, and fields, checks data types, and finds illegal null values, missing Glossary entries, undefined map units, etc.

Additional database elements

Construction of compliant databases will be facilitated by the creation of two additional point feature classes and a non-spatial table.

Stations (point feature class, optional)

Useful mostly for creating a compliant geodatabase. Following fields are suggestions

Stations_ID *Primary Key. Example values = STA1, STA2, STA3, ...Unique in database.*
FieldID *Primary key within field stations list/table. E.g., RH09-234. Identifier for station assigned by original station locator.*
LocationConfidenceMeters *Radius in meters of positional uncertainty envelope. How well located*

is the observation or sample locale? Null values not permitted*

MapUnit *Foreign key to DescriptionOfMapUnits. The map unit identified as outcropping at the station*

Notes *FreeText; any observation narrative associated with station*

FgdcSymbol *Identifier for symbol to use in map portrayals of station location. Null values indicates station should not be shown in map display*

Label *Text string to display on map portrayal next to station symbol*

PlotAtScale *Data type = float. At what scale (or larger) should this observation or analysis be plotted? At smaller scales, it should not be plotted. Useful to prevent crowding of display at small scales and to display progressively more data at larger and larger scales. Value is scale denominator. Null values not permitted*

DataSourceID *Foreign key to DataSources table, to track provenance of each data element. Null values not permitted*

A stations point feature class might also include these fields:

TimeDate Time and date of observation at station

Observer *Name of the person who located station*

OutcropDiameterMeters *Significant diameter of outcrop, in meters. Null values OK. Value of 0 indicates no outcrop at station location*LocationMethod *Values = RecreationalGPS, SurveyGradeGPS, ByInspection, ByOffset, …*

GPSCoordinates *May differ from map coordinates because of GPS error or (more likely) base map error*

PDOP *Data type=float. Predicted Dilution Of Precision; an estimator of GPS accuracy*

MapCoordinates *Station coordinates as compiled on the base map; base map should be identified for the station in the DataSources recor.*

MapUnitPoints (point feature class, optional)

Some map producers generate the MapUnitPolys feature classs from the ContactsAndFaults feature class and a feature class of 'label' points that holds the attributes associated with the polygons. This workflow utilizes the Feature to Polygon tool in the Data Management toolbox. If this workflow is adopted, we recommend that a user-defined and maintained field be used as the primary key (e.g. MapUnitPoints_ID) linking the label points to the MapUnitPolys, not the ESRI-maintained ObjectID.

Most map producers will find it easier to attach correct symbol values to polygon features if they first add field AreaSymbol to the DescriptionOfMapUnits table and populate this field with signifiers for the chosen area fills.

ChangeLog (non-spatial table, optional)

This table maintains information about updates to information contained in the database and is essential for documentation of the provenance of data from another source that are modified in the course of creating a new geologic map database. Each record records changes to a single database row, with old value, new value, and (if desired) the reason for a change in a NOTES field. One ChangeLog entry can record simultaneous changes to values in several fields of a single record. All fields except Notes could be populated automatically upon editing of a data record, and, for the sake of completeness, we highly recommend this. Changes to feature geometry (e.g., moving a vertex) are recorded by indicating that the changed field is 'shape'. To simplify the logging process, record only that the geometry was changed, not the explicit geometric changes. Creation of a new record need not generate a ChangeLog entry; the creation event is recorded in the DataSources record initially associated with the data item.

ChangeLog_ID *Primary key. Example values = CHL1, CHL2, CHL3, … Null values not permitted*
OwnerTable *Full name of table that contains owning element, e.g., DescriptionOfMapUnits, or OverlayPolys. Null values not permitted*
OwnerID *Foreign key to any table in the database. Null values not permitted.*
ChangedWhen *System clock date/time. Null values not permitted. Date and time of most recent update to the indicated records.*
ChangedBy *System userID. Null values not permitted. Generally obtained by operating system request. Login name for account under which the application is running.*
OldValue *String tuple of former values of all attributes changed, placeholders for unchanged attributes, with a flag for shape. Null values OK if entry documents a new feature record*
NewValue *String tuple of new values of all attributes changed, with placeholders for unchanged attributes, flag for shape. "Deleted" is special value. Null values not permitted*
Notes *Place to (optionally) record why an attribute or shape has been changed. Null values OK*

Frequently asked questions

What about annotation?

There are multiple ways to create and store annotation. We are not sure what data structure will best facilitate publication-quality cartography and allow economical creation and editing of annotation, so we have not prescribed a protocol for annotation. Users may wish to include one or more ESRI annotation feature classes along with instruction on how to use them.

My map is a grid. How does it fit into this schema?

Grid-based datasets are outside the scope of this schema. Suggestions for good raster-based database design are encouraged.

How should I encode structure contours?

You have at least two choices. Structure contours may be encoded in the OtherLines feature class, with Type=TopFormationX (or whatever is contoured), where the type has a corresponding Glossary entry to clearly define the contoured surface. The elevation values could be put in the label field for the line, or associated through ExtendedAttributes. Alternately, create a new, appropriately-named line feature class with an elevation attribute.

Contours are difficult to analyze automatically. The information contained in structure contours might be better stored as a raster (ESRI grid) or triangulated irregular network (TIN).

Does this standard apply to a visualization of already-published data?

No. However, it does apply to a digital transcription (automation) of a geologic map that has only been published in analog (paper or PDF) form.

What about my fault map? (It doesn't show geologic units)

A fault map is not a geologic map, so this standard does not apply. However, most fault maps are analogous to parts of geologic maps and this standard may provide useful guidance. Faults are lines that could be encoded in ContactsAndFaults and associated tables. There could be at least one polygon, outlining the mapped area, and its map-unit might be 'area covered by this map'.

May I give my clients databases in another format?

Certainly. But make this format available also.

My map has auxiliary maps showing data sources and the distribution of sedimentary facies in the Miocene. Where do these maps fit in this schema?

The answer varies. Not all information depicted via an analog auxiliary map needs a separate digital map (feature class). Data sources are best handled as attributes of various map features and recorded via DataSourcesID and the DataSources table. Distribution of Miocene sedimentary facies could be handled via ExtendedAttributes for polygons of Miocene sedimentary rocks, via overlay polygons, or via a new polygon feature class. Use your judgment.

I have an existing database with thickness, minimum age, maximum age, and lithology fields for each map unit. Where does this information go if I translate the database to this schema?

There are several options. In general, such attributes can be (1) mapped into the appropriate existing fields in the geodatabase schema, (2) stored in ExtendedAttributes, (3) stored in new fields added to existing non-spatial tables, or (4) stored in new non-spatial tables. **Where option 1 is available, use it!** Minimum and maximum ages of map units should be stored in the GeologicEvents table and associated with DescriptionOfMapUnits using ExtendedAttributes. If the information in your existing lithology field can without loss be transformed into StandardLithology, it need not be stored separately.

If you are sure that your information has no designated place in this schema (e.g., unit thickness, tabular lithologic descriptions more nuanced than can be supported by StandardLithology), consider options 2, 3, and 4. Your choice between them should be driven by (a) how many data are there (if attributes are only known for a few features, ExtendedAttributes is a more likely choice), (b) where are database users most likely to find and understand the data, and (c) what option is the least work and the least likely to generate transcription errors?

How can I tell if a database is compliant?

Try testing the database with script ncgmp09_validate.py. Note that passing the tests in this script does not ensure compliance; however, if a database fails these tests it is not compliant.

How do I use one of these databases to make a publication-quality map image?

This is a non-trivial problem. By standardizing a database schema we hope to see the emergence of community tools to solve it. Here are some hints: (A) Proper symbolization of faults with line ornaments (thrust triangles, extensional fault ticks) that are segmented by abutting contacts and (or) are locally concealed requires that you create a continuous fault trace analogous to 'routes' in workstation Arc-Info. Draw individual fault arcs as thick lines, thick dashed lines, and thick dots. Smooth (generalize, spline) the meta-faults and draw them with thrust triangles or extensional ticks as appropriate, but no line stroke. (B) Create good annotation (see FAQ on annotation above). We are not aware of tools that successfully automate this task. Dip and plunge values for measured orientations, text associated with other point data, map unit labels, and place names all may need to be positioned, eliminated, duplicated, or moved and have leaders added (unit labels). (C) Do as much of the preparation of the map image in ArcMap as possible. If necessary, the map image(s) can be exported to Adobe Illustrator, translating fonts as needed, for detailed graphic fine tuning. Insofar as possible, avoid cartographic work in a Illustrator or similar software as this often leads to synchronization problems, with the geology portrayred on the map image different from that recorded in the database. (D) Lay out the map sheet with page-layout software (e.g. Adobe InDesign), not Illustrator, as text formatting and figure placement are much easier.

I still don't know what metadata for a geologic map should look like. What do I do?

See http://geology.usgs.gov/tools/metadata/

Who is going to enforce this?

If adopted by the National Cooperative Geologic Mapping program, conformance to some degree may be required on delivery of products to the Program. If adopted by the USGS as a whole, Enterprise Publication Network may check for conformance as part of the publication process. If the schema is widely adopted, users will demand conformance so that tools developed to manipulate these databases work.

I've got a better design for a standard geologic-map database. How do I go about getting this proposal changed?

See Review, comment, and revision section, above.

Appendix A. Pre-defined vocabularies

Vocabularies for use with the delivery database are included in a collection of Excel spreadsheets, MS Word documents and Adobe Acrobat files included in a Vocabularies directory with the database distribution package. Some included vocabularies are those applicable to the delivery database schema that have been adopted for use by the GeoSciML Testbed 3 working group. Other vocabularies have been developed by the NGMDB project in conjunction with a variety of activities over the last 10 years. As part of the review process for this schema we invite comments on these vocabularies. We expect to see some of these vocabularies (particulary those associated with StandardLithology) finalized as this schema is adopted, as there are substantial benefits to having these vocabularies be stable.

Scientific confidence terms

std	The attribute assignment is considered reliable with a standard level of confidence
low	The associated attribute assignment is uncertain,
unk	Unknown reliability, generally for use with legacy data.

Property Value qualifier vocabulary

Some example values that might be used to qualify property values in ExtendedAttributes.

Always	Denotes that property value or relationship applies at all observed locations, and is expected to apply everywhere.
Common	Denotes that property value or relationship applies at most observed locations, and is expected to apply at most locations.

Sometimes	Denotes that property value or relationship is observed at less than 25 percent of locations, and is expected to apply in to less than a quarter of locations.
Rare	Denotes that property value or relationship is observed at less than 1 percent of locations, and is expected to apply only rarely.
Never	Denotes that property value or relationship has not been observed, and is not expected to apply at any location or under any condition.

Property vocabulary

The following table lists a variety of other properties that might be associated with a map unit through the ExtendedAttributes table. These have been extracted from the GeoSciML version 2 model, and from NGMDB vocabulary compilations. Vocabularies for populating these properties have been compiled but are not included with this package. The NGMDB vocabularies are available at http://ngmdb.usgs.gov/Info/standards/NGMDBvocabs/; please note – these are draft unpublished documents, offered to the community in order to provide terminology lists and definitions that may be found useful by projects and agencies, and to improve the vocabulary content. Please send comments to Dave Soller and Steve Richard (drsoller@usgs.gov, steve.richard@azgs.az.gov).

Property	**Scope notes**
Bedding Pattern	Term(s) specifying patterns of bedding thickness or relationships between bedding packages, Examples: thinning upward, thickening upward
Bedding Style	Term(s) specifying the style of bedding in a stratified geologic unit, e.g. lenticular, irregular, planar, vague, massive
Bedding Thickness	Term(s) or numeric values characterizing the thickness of bedding in the unit.
Body Morphology	The geometry or form of a Geologic Unit. Examples include: dike (dyke), cone, fan, sheet, etc. Morphology is independent of the substance (Earth Material) that composes the Geologic Unit.
Clast weathering degree	The degree of weathering intensity of clasts in sedimentary surficial deposits. Classification is based on degree of weathering of clasts that were originally indurated material.
Clast weathering style	The weathering style of clasts on a surface. Examples: pitted, etched, weathering rinds.
Composition Category	Term to specify the gross chemical character of geologic unit. Examples: silicate, carbonate, ferromagnesian, oxide. Chemical classification terms for igneous rocks also go here. Examples: alkalic, subaluminous, peraluminous, mafic, felsic, intermediate.
Contained Structure	Geologic structures that are present in a geologic unit.
Exposure Color	Typical color at the outcrop of a geologic unit.
Genetic Category	A term that represents a summary geologic history of a geologic unit. (ie, a genetic process classifier term) Examples include igneous, sedimentary, metamorphic, shock metamorphic, volcanic, pyroclastic.
Magnetic Susceptibility	Material magnetic susceptibility, customarily measured in SI units. The ratio of induced magnetization to the strength of the magnetic field causing the magnetization. Note that volume magnetic susceptibility is dimensionless, being magnetization (magnetic dipole moment) in amperes per meter (SI) divided by the

	applied field, also in amperes per meter. However, many tables of magnetic susceptibility and some instruments give cgs values that rely on different definitions of the permeability of free space than SI values. The cgs value of susceptibility is multiplied by 4pi to give the SI susceptibility value. For example, the cgs volume magnetic susceptibility of water at 20°C is -7.19x10-7 which is -9.04x10-6 in SI. The xml encoding should specify whether the uom is SI or cgs, and if in cgs provide a
Metamorphic Facies	A description of characteristic mineral assemblages indicative of certain metamorphic P-T conditions. Examples include Barrovian metasedimentary zones (e.g.: biotite facies, kyanite facies) or assemblages developed in rocks of more mafic composition (e.g.: greenschist facies, amphibolite facies).
Metamorphic Grade	A term to indicate the intensity or rank of metamorphism applied to an EarthMaterial (eg: high metamorphic grade, low metamorphic grade). Indicates in a general way the P-T environment in which the metamorphism took place. Determination of metamorphic grade is based on mineral assemblages (ie, facies) present in a rock that are interpreted
Outcrop Character	Describes the nature of outcrops formed by a geologic unit. Examples: bouldery, cliff-forming, ledge-forming, slope-forming, poorly exposed
Permeability	The measure of the capacity of a porous material to transmit a fluid under unequal pressure. Customary unit of measure: millidarcy
Porosity	The percentage of the bulk volume of a material that is occupied by interstices, whether isolated or connected.
Protolith	An interpretation of the EarthMaterial that constituted the pre-metamorphic lithology for this metamorphosed CompoundMaterial.
Stratigraphic Rank	Term that classifies the geologic unit in a generalization hierarchy from most local/smallest volume to most regional. Scoped name because classification is asserted, not based on observational data. Examples: group, subgroup, formation, member, bed, intrusion, complex, batholith
Soil Development	Characterization of soil in a surficial deposit.
Surface morphology	Characterization of the form of the surface developed on a unit.
Surface dissection	The degree to which the upper surface of unconsolidated sedimentary material has been degraded and incised by erosion after the unit has been abandoned by the geologic processes that formed it.
Surface armoring	Characterization of the development of pavement or other surface armor on a surficial deposit.
Surface varnish	The degree of development of rock varnish on clasts on an outcrop surface.
Weathering Degree	term to specify degree of modification from original material, e.g. slightly weathered, strongly weathered, weathered rock grade III
Unit Thickness	Typical thickness of the geologic unit.
Weathering product	material result of weathering processes, e.g. saprolite, ferricrete, clay, calcrete. Materials observed in a soil profile could be identified using this property, but EarthMaterial content model does not allow representation of relationships between materials in a soil profile. A full soil profile description would have to use GeologicUnitParts and Composition part?
Weathering Environment	Terms to specify the environmental context of the weathering description. Typically would be specified by terms for climate (tropical, arid, termperate, humid, polar..)
Peak metamorphic	A numerical value to indicate the estimated temperature at peak metamorphic

temperature	conditions.
Peak metamorphic pressure	A numerical value to indicate the estimated pressure at peak metamorphic conditions.
Density	Material mass per unit volume
Weathering Process	Characteristic weathering process, e.g. leaching, accumulation

Appendix B. Translation of geodatabase field names to shapefile field names

See associated spread sheet FieldNamesLookupFor_dbf.xls.

Appendix C. FGDC geology symbolization

To be determined. Might include symbolization (style file, layer file, …).