DIGITAL MAPPING TECHNIQUES 2020

The following was presented at DMT’20
(June 8 - 10, 2020 - A Virtual Event)

The contents of this document are provisional

See Presentations and Proceedings from the DMT Meetings (1997-2020)

http://ngmdb.usgs.gov/info/dmt/
Attribute Rules and Dictionary Symbology in ArcGIS Pro Help Streamline Geologic Map Compilation in GeMS

Tracey J. Felger
USGS, Geology, Minerals, Energy, and Geophysics Science Center
Flagstaff, AZ
(tfelger@usgs.gov)
Talk Overview

• Background
• What are Attribute Rules?
  • Test Case – apply to ‘Traditional’ workflow
• What is Dictionary Symbology?
  • Test Case – apply to ‘Granular’ data-driven workflow
• Combining Attribute Rules and Dictionary Symbology for an Optimized Workflow
• Summary
Background

- **Current projects** - all funded by the National Cooperative Geologic Mapping Program (NCGMP)
  - Mojave Desert/Eastern California Shear Zone (MOJO)
  - Lower Colorado River (LOCOS)
  - Enterprise GIS for the NCGMP Community

- **Goals for Fiscal Year 2020**
  - Embrace the Geologic Map Schema (GeMS), which is mandated by NCGMP, instead of just coping with it!
  - Transition compilation workflow from ArcMap to ArcPro in order to leverage new functionality that would improve compilation efficiency and data quality
Traditional vs Granular Approach

- **Traditional (Cartography-driven)** – the symbol and the data are intertwined, and can be easily represented in one field (e.g. ALACARTE)
  - Main purpose is to make a cartographic product

- **Granular (Data-driven)** – information is disaggregated into the smallest pieces possible, and is stored in many fields (e.g. GeMS)
  - Machine readable (good for analysis)
  - Can be aggregated and disaggregated in different ways as needed
  - Easily merged with other data and effectively integrated and managed
  - Better for making a multi-map compilation!

<table>
<thead>
<tr>
<th>Feature Layer</th>
<th>ALACARTE Fields</th>
<th>GeMS Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contacts and Faults</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>Map Units</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>Structural Measurements</td>
<td>3</td>
<td>12</td>
</tr>
</tbody>
</table>
Coping Strategy…

- Populate 1 or 2 fields while editing and then use scripts in the GeMS toolbox, such as Attribute By Key Values and Reset ID Values (shown in snapshot to right) to populate other fields, either when map is finished, or periodically during compilation

- GeMS toolbox (https://github.com/usgs/gems-tools-pro)

- Ryan Crow’s Map Extractor and Schema Converter tools includes other useful scripts not available in the GeMS toolbox (https://github.com/rcrow/MapExtractor_SchemaConverter)

*This works, but for me it isn’t very satisfying, and is an obstacle to embracing GeMS!
December, 2019 - Started compiling new MOJO mapping

- Downloaded the GeMS toolbox for Pro (https://github.com/usgs/gems-tools-pro)

- Ran the Create New Database script (and it worked – yay!)
  - It took 12 minutes to create an empty gdb with all the bells and whistles – the ArcMap version took < 2 minutes!

- Symbolized ContactsAndFaults twice, once on Symbol, and a second time on LocationConfidenceMeters (LCM)
  - This is a neat trick that Ralph demonstrated at the 2019 DMT that makes it very easy to visualize LCM spatially

- Started compiling using a ‘traditional’ workflow
  - Digitize ContactsAndFaults - Populate Symbol, LCM, and DataSourceID fields
  - Digitize MapUnitPoints - Populate MapUnit and DataSourceID fields
  - Build MapUnitPolys from ContactsAndFaults and MapUnitPoints

- Ran the Validate Database script (and it worked – yay!)
  - But I had a lot of errors because of unpopulated fields (boo-hiss!)

- Run Reset ID values, Attribute by Key Values; rerun Validate Database (sigh…)
Tracey

There’s gotta be a better way!!
Can Attribute Rules Lighten the GeMS Workload?

Attribute rules:

• user-defined Arcade scripts embedded within a geodatabase

  *Arcade is a lightweight scripting language specific to the ESRI platform; similar to JavaScript*

• triggered automatically by editing operations or can be run on-demand

  *Similar to formulas and macros in Microsoft Excel, and triggers and stored procedures in SQL*

• 3 types of rules:
  • calculation rules - automatically populate attributes
  • constraint rules - restrict invalid edits during edit operations
  • validation rules - perform quality assurance checks on existing features
Test Case – Apply Attribute Rules to a ‘Traditional’ Workflow

• Goals:
  • Don’t let me create or modify features unless certain fields are populated
  • Automatically populate feature_ID field
    • so I don’t need to run the Reset ID values script
  • Automatically populate dependent fields based on the value in Symbol
    • so I don’t need to run the Attribute By Key Values script

• I guess I need to learn Arcade!

*Arcade is a lightweight scripting language specific to the ESRI platform; similar to JavaScript
If LCM and DSID are not populated and you try to create a feature (or modify an existing feature), you will get a message like this one, and you won’t be able to create the feature or save changes.
MapUnit Points (Traditional) – Constraint Rules

Note: The same rules can be applied to MapUnitPolys if you create polys interactively, instead of from lines and points.
# ContactsAndFaults (Traditional) – Calculation Rules

**Rule #1 = Reset ID values script**

**Rule #2-5 = Attribute By Key Values**

**Example:**

If Symbol = 01.01.01 (Contact—Identity and existence certain, location accurate)

Type = Contact

IsConcealed = N

ExistenceConfidence = certain

---

<table>
<thead>
<tr>
<th>#</th>
<th>Rule Name</th>
<th>Description</th>
<th>Subtype Field</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Calc ID</td>
<td>calc ID from DMD</td>
<td>ContactsAndFaults</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Calc Type</td>
<td>calc_type from Symbol</td>
<td>ContactsAndFaults</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Calc IsConcealed</td>
<td>calc_is_concealed from Symbol</td>
<td>ContactsAndFaults</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Calc ExistenceConfidence</td>
<td>calc_existence_confidence from Symbol</td>
<td>ContactsAndFaults</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Calc IdentityConfidence</td>
<td>calc_identity_confidence from Contact</td>
<td>ContactsAndFaults</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>Calc GeMS Ltype</td>
<td>calc_geMS_ltype from ContactsAndFaults</td>
<td>ContactsAndFaults</td>
<td>6</td>
</tr>
</tbody>
</table>

User-populated fields = Symbol, LCM, and DSID

Must add GlobalID field to use attribute rules (ESRI-defined, auto-populated)

Logic-based concatenation of Type, IsCon, ExConf, and LCM

215 else-if statements! Easily created from FGDC_Lookup.xls
MapUnitPoints (Traditional) – Calculation Rules

User-populated fields = MapUnit, IDConf, and DSID

Could be changed to populate with the FGDC symbol number

Could be changed to something more complicated, such as incorporating Geofont symbols (e.g. ^ for Triassic), or compound surficial map units
Yay! I don’t have to run Reset ID Values and Attribute By Key Values anymore!

Goals:

✓ Don’t let me create or modify features unless certain fields are populated
✓ Automatically populate _ID
✓ Automatically populate dependent fields based on the value in Symbol

But I’m still thinking in a cartographic, not ‘granular’, data-driven way, partly because of the way we symbolize our data…
ContactsAndFaults (Traditional) Symbology

• Symbolize on unique values in a field
  • E.g. Match to FGDC_GSC.style on Symbol
  • All symbol permutations must exist
  • Limited to unique values in no more than 3 fields

But the intent of GeMS is that the symbol would be calculated on the basis of map scale and the attributes Type, IsConcealed, LocationConfidenceMeters, ExistenceConfidence, and IdentityConfidence

• The Set Symbol Values script will do this, but I want to symbolize on all of those attributes while I’m editing!
Can Dictionary Symbology Help?

• Dictionary Symbology:
  • An ‘intelligent’ style file (mobile .stylx) stored in an SQLite database that contains:
    • Symbol ‘building blocks’
    • Embedded configuration parameters (JSON)
      • Specifies which fields are used for symbology, which fields are used for text, and properties that can be user-controlled
    • Embedded dictionary of rules
      • user-defined Arcade scripts that control how symbol components are aggregated to visually represent the data, depending on the feature attributes
  • Builds symbols on-the-fly from components, based on one or more attributes of a feature, and logic (optional)

Very powerful and complicated - hard to explain in < 20 minutes!
1. Get the value in Type (e.g. Contact)
2. Get the value in IsConcealed (IsCon)
   - If IsCon = Y, get a dotted contact line
   - If IsCon = N, go to Step #3
3. Get the value in LocationConfidenceMeters (LCM)
   - If LCM <= 20, get a solid contact line
   - If LCM >20 and <50, get a short-dash contact line
   - If LCM >=50, get a long-dash contact line
4. Get the value in ExistenceConfidence (ExConf)
   - If ExConf = questionable, get a contact-weight questionmark
5. Symbolize the line with the symbol parts ‘collected’ in steps 1-4
Yay! Now I can interact with my data on a granular level!

Tracey
Attribute Rules + Dictionary Symbology = Optimized GeMS Workflow
ContactsAndFaults – Optimized Workflow

#1. Populate Type, IsCon, LCM, ExConf, and DSID, and use Constraint rules to make sure LCM and DSID are populated

<table>
<thead>
<tr>
<th></th>
<th>IsConcealed</th>
<th>LCM</th>
<th>ExConfidence</th>
<th>DataSourceID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>No</td>
<td>10</td>
<td>certain</td>
<td>TFELGER</td>
</tr>
<tr>
<td>Contact</td>
<td>Yes</td>
<td>10</td>
<td>questionable</td>
<td>TFELGER</td>
</tr>
<tr>
<td>Fault</td>
<td>No</td>
<td>5</td>
<td>certain</td>
<td>TFELGER</td>
</tr>
<tr>
<td>Normal fault</td>
<td>No</td>
<td>50</td>
<td>questionable</td>
<td>TFELGER</td>
</tr>
</tbody>
</table>

#2. Use Calculation rules to populate _ID, IdentityConfidence, Symbol, and GeMS_Ltype

![Calculation rules screenshot]

#3. Symbolize twice – once using Dictionary Symbology, again on LocationConfidenceMeters

![Symbolization screenshot]
MapUnitPolys – Optimized Workflow

#1. Create polys interactively, and populate MapUnit, IdConf, and DSID, and use Constraint rules to make sure MapUnit and DSID are populated

#2. Use Calculation rules to populate _ID, Label, and Symbol

#3. Create polys interactively from ContactsAndFaults, but maintain a MapUnitPoints layer in case I don’t stick with that! Use Calculation rules to get all attributes from ‘parent’ poly
Summary  Game changer!

Benefits:

1. **Improved efficiency** – attributes automatically populated on the fly, no need to periodically run several scripts

2. **Improved data quality** –
   a) features can’t be created/modified unless certain fields are populated
   b) all fields are always fully populated, so Validate database script produces fewer errors

3. **Helping me embrace GeMS instead of just coping with it!**

Challenges and Limitations:

1. Can only use in Pro - not backward compatible with ArcMap *(there are workarounds)*

2. Have to learn Arcade

3. Dictionary symbology is very powerful, but very complicated and not well documented!!

Testing so far:

- Spatial data only (haven’t tested on non-spatial tables)
- ContactsAndFaults and MapUnitPts/Polys only
- Calculation and constraint rules only (haven’t tried validation rules) *(I’m happy to share!)*
- Limited to 215 FGDC contacts and fault types so far
- Latitude 7212 ruggedized tablet; Windows 10, Pro 2.5 *(I started in 2.4; after upgrading to 2.5 the GeMS Validate Database script stopped working!)*

Next Steps:

- Expand and refine as I need to for my mapping
- Try validation rules
- Dictionary symbology - implement user-control options, and symbol override capabilities