DIGITAL MAPPING TECHNIQUES 2021

The following was presented at DMT’21
(June 7 - 10, 2021 - A Virtual Event)

The contents of this document are provisional

See Presentations and Proceedings from the DMT Meetings (1997-2021)

http://ngmdb.usgs.gov/info/dmt/
Building an Automated GeMS-style Submission from an Enterprise Geodatabase

Christian Halsted
Director, Earth Resources Information
Maine Geological Survey
Christian.H.Halsted@Maine.gov
The Maine Geological Survey has a seamless, statewide, multi-quadrangle and multi-scale enterprise geodatabase that is used to provide data management for all bedrock and surficial mapping products. This past year MGS developed the automation tools to convert and export our data into the GeMS schema. This has allowed us to continue mapping with our database schema and tools while also being able to meet Level 3 GeMS compliance requirements. It is also leading to quicker adoption of GeMS by the mapping geologists because the impact on their workflow has been minimized. This presentation will review the process that went into completing the project as well as open questions we still have to resolve.
How Did We Get Here?

1996: Quadrangle-based mapping with ArcInfo Coverages
2014: SQL Server Express Enterprise Geodatabase multi-quadrangle, multi-scale, multi-map type
2017: NCGMP09
2018: MGS - GeMS gap analysis
2021: First GeMS submission
Guiding Principles

• Minimal disruption to current map production
• Minimal level of additional work by the mapping geologists and information management staff
• Treat GeMS as a data standard, not map standard
• Enable MGS to fully comply with GeMS Level 3 submission requirements
• Create a defined, scripted, documented workflow for bedrock and surficial maps at 24K and 100K scales
• Improve MGS workflows around GeoLex, feature-level metadata, map text
What Is Missing?

Inventoried MGS feature classes to discover required GeMS attributes that need to be captured.

<table>
<thead>
<tr>
<th>GeMS Field</th>
<th>MGS Feature Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDENTITY_CONFIDENCE</td>
<td>• Bedrock_Contacts&lt;br&gt;• Bedrock_Line_Features&lt;br&gt;• Bedrock_Overlay_Polygons&lt;br&gt;• Bedrock_Units&lt;br&gt;• Bedrock_XSection_Lines&lt;br&gt;• Bedrock_XSection_Overlay_Polygons&lt;br&gt;• Bedrock_XSection_Units&lt;br&gt;• tbl_Bedrock_Feature</td>
</tr>
<tr>
<td>EXISTENCE_CONFIDENCE</td>
<td>• Bedrock_Contacts&lt;br&gt;• Bedrock_Line_Features&lt;br&gt;• Bedrock_XSection_Lines</td>
</tr>
<tr>
<td>LOCATION_CONFIDENCE</td>
<td>• Bedrock_Contacts&lt;br&gt;• Bedrock_Line_Features&lt;br&gt;• Bedrock_XSection_Lines&lt;br&gt;• tbl_Bedrock_Points</td>
</tr>
<tr>
<td>ORIENTATION_CONFIDENCE</td>
<td>• tbl_Bedrock_Feature</td>
</tr>
<tr>
<td>LOCATION_METHOD</td>
<td>• tbl_Bedrock_Points</td>
</tr>
</tbody>
</table>

List of additional fields needed for GeMS compliance in Bedrock 24K mapping feature classes and tables.
MGS map explanation table needed a number of new fields to control GeMS destination feature classes and description of map units.
Reference Tables

Reference tables for geomaterials, paragraph styles, and FGDC colors, symbols, and patterns were created in the MGS database to control data entry. In the case of colors and patterns, these tables also provide a crosswalk between the values traditionally used by MGS and the FGDC values.
The GeMS Create New Database tool was used to create a file geodatabase with all the default tables and feature classes. These objects were imported into the MGS geodatabase to serve as staging tables/feature classes for GeMS data.
Data Mapping

The existing MGS feature classes and tables, or subsets of features, were mapped to the required GeMS feature classes and tables.

<table>
<thead>
<tr>
<th>MGS Feature Class (criteria)</th>
<th>GeMS Feature Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Units</td>
<td>MapUnitPolys</td>
</tr>
<tr>
<td>Contacts (not overlapped by Bedrock_Lines)</td>
<td>ContactsAndFaults</td>
</tr>
<tr>
<td>Lines (with GEMS_FeatureClass = 'ContactsAndFaults')</td>
<td></td>
</tr>
<tr>
<td>Units (dissolved units to make map boundary)</td>
<td></td>
</tr>
<tr>
<td>Points</td>
<td>Stations</td>
</tr>
<tr>
<td>Points_Labels_Symbol (with GEMS_FeatureClass = 'OrientationPoints')</td>
<td>OrientationPoints</td>
</tr>
<tr>
<td>Points_Labels_Symbol (with GEMS_FeatureClass = 'GeologicPoints')</td>
<td>GeologicPoints</td>
</tr>
<tr>
<td>Units_Labels</td>
<td>GenericPoints</td>
</tr>
<tr>
<td>Line_Features_Labels</td>
<td>GeologicLines</td>
</tr>
<tr>
<td>Miscellaneous_Points (SYMOLOGY &lt;&gt; 'Geochronology point')</td>
<td>CartographicLines</td>
</tr>
<tr>
<td>Line_Features (GEMS_FeatureClass = 'GeologicLines')</td>
<td></td>
</tr>
<tr>
<td>Leaders</td>
<td></td>
</tr>
<tr>
<td>Units (SYMOLOGY_OVERLAY IS NOT NULL)</td>
<td>MapUnitOverlayPolys</td>
</tr>
<tr>
<td>Overlay_Polygons</td>
<td>OverlayPolys</td>
</tr>
<tr>
<td>Quadrangle_Split_Authors</td>
<td>DataSourcePolys</td>
</tr>
<tr>
<td>tbl_Publications</td>
<td>MiscellaneousMapInformation</td>
</tr>
<tr>
<td>XSection_Units</td>
<td>CSXMapUnitPolys</td>
</tr>
<tr>
<td>XSection_Lines (with GEMS_FeatureClass = 'CSXContactsAndFaults')</td>
<td>CSXContactsAndFaults</td>
</tr>
<tr>
<td>XSection_Overlay_Polygons</td>
<td>CSXOverlayPolys</td>
</tr>
<tr>
<td>XSection_Lines (with GEMS_FeatureClass = 'CSXCartographicLines')</td>
<td>CSXCartographicLines</td>
</tr>
<tr>
<td>XSection_Leaders</td>
<td></td>
</tr>
<tr>
<td>XSection_Lines (with GEMS_FeatureClass = 'CSXGeologicLines')</td>
<td>CSXGeologicLines</td>
</tr>
<tr>
<td>XSection_Labels</td>
<td>CSXGenericPoints</td>
</tr>
</tbody>
</table>
Individual feature types in existing MGS feature classes can be mapped to their GeMS destination feature class using the GeMS:Feature Class field in the MGS Map Explanation table.

Example of different feature types in a single MGS feature class being mapped to two different GeMS feature classes.
Automated Data Migration

A custom Python tool in an MGS ArcPro toolbox:

• calls a SQL Server database stored procedure that
  • truncates (deletes) any existing data in each of the staging tables/feature classes
  • loads data to each of the staging tables and feature classes for the selected quadrangle, scale (24K or 100K), and map type (bedrock or surficial)
  • creates the GeMS submission fgdb in a staging folder
  • copies the staged GeMS data to the fgdb

This provides the flexibility to create GeMS submissions for any map published by MGS.
Automated Data Migration

SQL Server database stored procedure: uspGEMS_Stage_Data

355 SELECT statements from MGS data to create 22 GeMS feature classes/tables
~3200 lines of code

YIKES!!!
Manual Steps

Run GeMS Tools manually:
- Load default file geodatabase metadata and edit
- Run Validate Database tool and deal with any errors
- Run Geologic Names Check tool
- Run Metadata 1, 2, and 3 tools
- Run Translate to Shapefiles tool

Build GeMS submission map document which is not the same as the original map document
GeMS Submission Package

Manually compile all the components of the GeMS Package

- Original map PDF
- ArcMap document (not original – GeMS data only)
- Export map photos, text, figures, tables
- Collect style and font files
- Transmittal letter
- Map browse graphic

Create zip file

Post and notify NGMDB
Next Steps/Improvements

Metadata tools needed for ArcPro
Incorporate Validate Database into SQL Server db
Script gathering of resource (map photos, figures, tables, and text), map document, and graphic files
Script running of GeMS tools in a batch or in database
Script cleanup of intermediate validation and metadata files
Script building GeMS map document
Accomplishments

- Costs quantified
  - 8-12 hours for author geologist to review and complete GeMS data entry on a published quadrangle (longer if not the author)
  - 4-8 hours for data manager to compile GeMS Submission package
- Staff trained
- GeMS submission process fully documented
- MGS to GeMS translation logic centralized in MGS database
- Three Level 3 and one Level 1 GeMS packages successfully submitted in May 2021.

Questions?

Thanks!