

Automated Production of a Large Complex Mapset Using ArcGIS

by

**Eric Powell and Chris Shanks
Environmental and Geographic Information Systems
Bechtel Savannah River Incorporated
Aiken, South Carolina 29808**

October 21, 2002

Automated Production of a Large Complex Mapset Using ArcGIS

by

Eric B. Powell and Chris Shanks

Bechtel Savannah River, Inc.

Savannah River Site, Aiken, SC

Abstract

Database cartography and automated map production can result in the ability to produce large complex mapsets. This results in significant time and cost savings when utilizing various GIS database, and automation tools in ArcGIS, MS Access, and Visual Basic. These tools enabled the Environmental & Geographic Information Systems (E&GIS) group to create a set of 150 maps based on a complex set of criteria from a relational database. E&GIS developed software application generated queries within the JET database engine and then inserted into map documents as event themes. The maps were then used to help with the reporting of a large complex geo-chemical dataset for environmental characterization.

Problem

The Environmental and Geographic Information Systems (EGIS) group was asked to generate nature and extent maps for a large geochemical dataset containing 200,000 records of analytical data, 45 constituents, and 4 media types (e.g. soil, surface water, sediment, and groundwater). The maps were to be used by environmental scientists for waste site characterization and ecological risk assessment. The information was presented in a Remedial Investigation/Baseline Risk Assessment (RI/BRA) document submitted to USEPA and the South Carolina Department of Health and Environmental Control.

The required mapset was to contain approximately 150 maps, one map per constituent of concern (COC) per extent, with three event themes per map: Detected Values and locations, Not-Detected locations, and Not-Analyzed/Sampled locations. The Detected-Value themes were to be color-coded by range, and all three event themes were to have pre-defined symbols.

The map extents were to be based on media, geographical extent, and soil depth. When the specified subunit contained more than one sample interval, an additional step was to be required to show the maximum detected value per location. Each of the extents was to be associated with one or more sub-units. Sub-units were defined as groupings of sample locations base on horizontal and vertical extents. Extents refer to the fixed area shown by a map template. The map templates contained base GIS data information such as roads, elevation contours, rivers, buildings, etc. The map set to be created had to contain the same base information with the same exact symbology and extent. Once the base information was created for each map, the constituent of concern data (such as lead) was added to the map showing the extent and depth of contamination.

The data for the maps were stored in an existing MS Access database containing geochemical data as well as sample location

information. The initial database design did not include tables to store all of the information needed to complete this project.

The questions for each type of event-theme were determined to be:

Detected-Value:

Generate a list of locations, coordinates and values where the analytical result is greater than the detection limit for the specified constituent in the specified subunit for the specified media.

-If the subunit contains more than one depth interval, show the value for the interval with the highest value.

Not-Detected:

Generate a list of locations and coordinates where analytical results for the specified constituent and the specified subunit are less than the detection limit.

-If the subunit contains more than one depth interval, only show those locations where all depth intervals are non-detects.

Not-Analyzed

/Sampled:

Generate a list of locations and coordinates where there were no geochemical analysis result for the specified subunit for the specified media for the specified constituent.

-If the subunit contains more than one depth interval, show only the locations where there were no geochemical analysis for the specified analyte in any of the depth intervals.

-“Not-Analyzed/Sampled” was an important feature of each map because it told the

characterization team and the regulatory agencies where data was not available or could not be

collected because of weather conditions, lab conditions, or programmatic conditions (some constituents weren't analyzed because it wasn't part of the contract).

Solution

The solution

involved developing an application that uses several toolsets to accomplish the bulk of the work for the mapset. These toolsets included: MS DAO 3.x, MS ADO 2.7, ArcObjects 8.x and MS Visual Basic 6.0. The solution was composed of three elements: 1) An MS Access Database that stored the analytical and cartographic data, 2) A Visual Basic Application that generated queries and added them to the base maps, and 3) The map templates to be used as base maps by the application. This architecture is diagrammed in Figure 1.

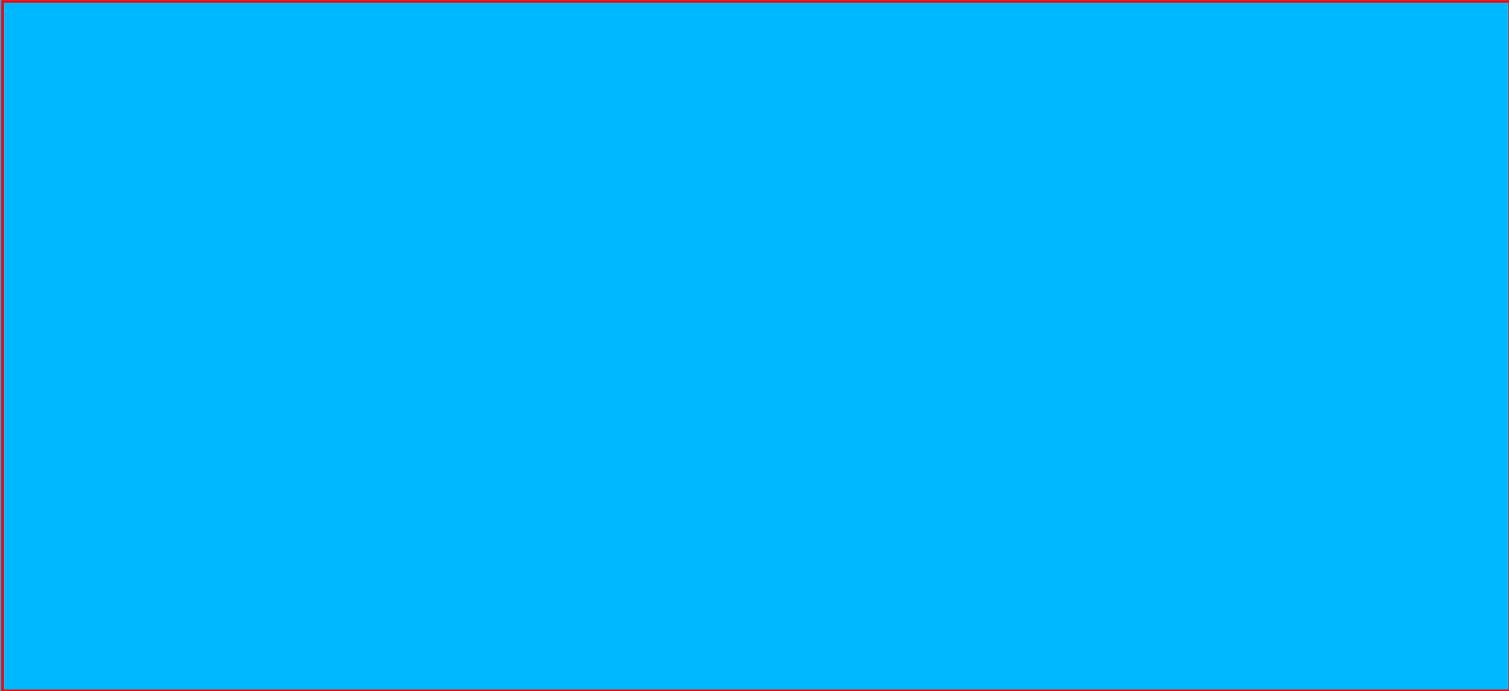


Figure 1: Solution Architecture

The application read all input from the project database tables and coupled it with built-in logic to generate the queries, map name, and the name of the template to use for each map. Table 1 lists the tables in the MS Access database portion of the solution as well as summarizing their functions.

Table Name	Function
Sample Locations	Contains Sample Location Name, XY Coordinates
Sample Depths	Links to Sample Locations, Contains depth intervals, Sub-unit assignments
Map Names	Contains the final map name associated with legend, Links to Map_Key
Map Key	Contains Detected_Value theme codes, legend text, constituent name, units

Subunit_ID	Lookup table for Subunit Names, and associated matrix
Analyte_Data	Contains the geochemical analysis results
Analyte Table	Lookup table for constituent names

Table 1: Database Cartography Tables

The solution required that map templates be developed and approved by the customer for each geographical extent. These templates were named using a project-specific naming scheme and stored in a templates directory.

The application was designed to generate the appropriate queries in the database for each COC, copy the appropriate map template to a working directory, and then add and symbolize the event themes to it for each scenario. The final map was saved under a distinct final name determined at project initiation, and was stored in the MS Access database.

Steps in Solution Development

The initial steps in implementing the solution to the problem required streamlining and modifying the existing MS Access project database. Tables were added to the database to contain cartography information such as map names, template names, and the themes to be added to each map. The relational integrity of the spatial and non-spatial data was improved such that queries would be more efficient. Some data integrity checks were run at this stage of the process for QA purposes.

The next step included the development of logic to control automatic assignment of the proper themes to the proper basemap and then saving the resultant document under a pre-defined name read from the map_names table in the MS Access database. A class module was developed to automate the generation of the queries that were used as a basis of the event themes for the final maps. The MS DAO 3.x object library was used to add the queries to the MS Access database, MS ADO 2.7 was used to perform the value look-ups, and ArcObjects was used to manipulate the map templates and to add and symbolize the event themes.

The query logic to answer the event theme questions evolved throughout the project with continuing revisions from the project team. Because a Class Module generated the queries, changes to the code were isolated from the functionality of the rest of the Visual Basic application. Thus frequent changes in the SQL for the database queries did not slow E&GIS down during any part of the process. The queries were broken

down into three types: Detected Value, Non-Detect, and Not-Analyzed. At least one query from each group was included on every map.

The Detected-Value themes were defined as all locations within a subunit for a given media that had an analytical result greater than the detection limit. Some maps had to contain a special case of this query that showed the maximum detected value for subunits that contained more than one depth interval.

The Not-Analyzed theme was defined as being all locations in a given subunit (grouping of locations by spatial and vertical extent and media) for a given media that had no data in the analyte_data table for the specified constituent in any interval in the subunit. To determine which locations in the sample_locations table should be plotted with this theme, four steps were required:

- 1)Generate a list of unique Sample Locations and their coordinates within the given subunit.
- 2)Generate a unique list of all Sample Locations and results within the subunit with valid analytical data for the specified constituent.
- 3)Left join the first list to the second list on the sample location field.
- 4)All locations with a Null value in the right sample location field are Not-Analyzed locations.

Figure 2 associates the steps in the process with the resultant query names.

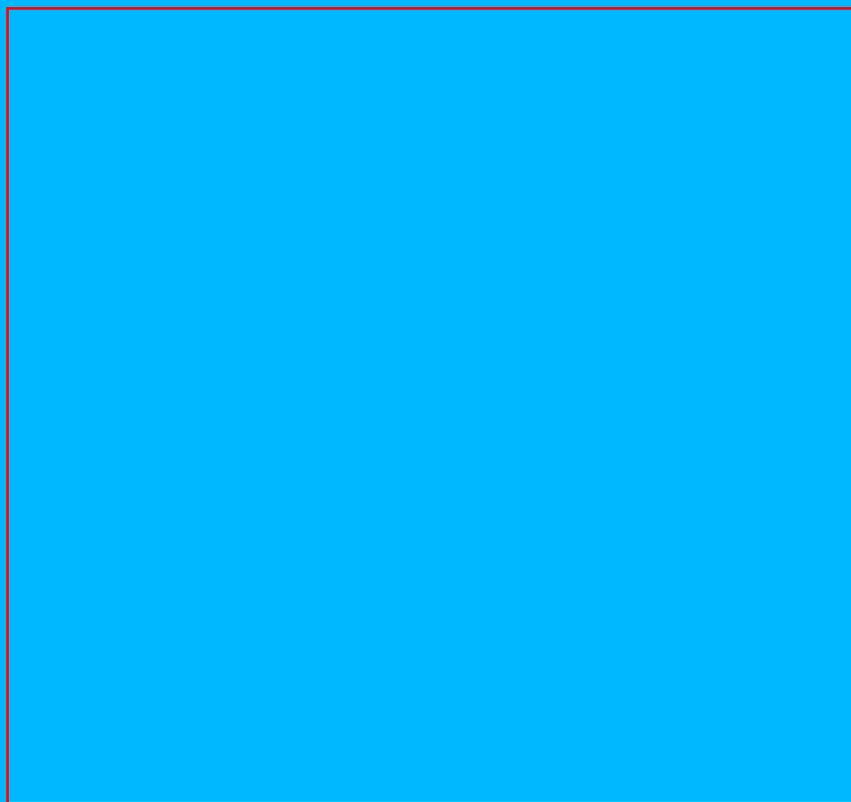


Figure 2: Not Analyzed Query Logic

The Not-Detected themes were defined as all locations within a subunit for a given media and given analyte where

the analytical result was less than the detection limit for all intervals within the subunit. The methodology for producing this theme is slightly more complex than the methodology for determining the Not-Analyzed theme and takes additional steps:

- 1)Generate a list of all locations and intervals located within a subunit
- 2)Count the number of intervals for each location in the subunit
- 3)Generate a list of locations and intervals where the analytical result is less than the detection limit
- 4)Count the number of intervals from Step 3.
- 5)Left join the counted list from Step 2 to the counted list in step 4 on the sample location field.

- 6)Sample locations where the count in the list from Step 2 equals the count in the list from Step 4 are the locations that should be plotted in the Non-Detect theme

Figure 3 relates the steps in the querying process to the query name generated by the application.



Figure 3: Non-Detect Query Logic

In all three-event themes, the final dynaset to be plotted as an event theme was stored in the database as a “View” and all sub-queries were stored under the nomenclature of “Query”.

Concurrently with the database and programming work, the map templates were developed and reviewed by the project team. Once the templates were finalized and approved, they were used to produce a set of draft maps using the solution.

The final step was the development of a class module within the application that used the ArcObjects programming tools to add the final views to the maps as XY themes, and then symbolize them appropriately. The Detected-Value (and Max –Value) themes were plotted as solid filled circles that were color-coded based on ranges. The color-coding was stored in the map_key table in the database. The Not-Analyzed values were plotted as open squares and the Not-Detected themes were plotted as open circles. These last two symbols were assigned by hard coding in the application because they did not need to be dynamic.

Results

At project initiation the estimated time to generate one map was 4.5 hours per map. Upon project completion, the actual time to generate one map was calculated to be 1 hour. The final time includes 30 minutes per map for labeling, legends and titles. The remaining 30 minutes were consumed with development time for the database, templates and the VB application. The final hand edits (labeling, legends and titles, and QA checks) were completed within one week of the application being completed. This meant that we created 150 maps ready for our customer in one week.

The solution produced a variety of increased efficiencies. First, members of the E&GIS organization were able to work in parallel on the various aspects of the solution. This advantage was leveraged by the relative ease of manipulating the application code to alter the SQL statements in the database without requiring changes to the nearly completed map documents. Second, the multi-faceted architecture allowed for several test runs of the toolset in order for the team to use the 'Rapid Prototype' development model. Third, the increased flexibility of the multi-faceted architecture was noted when the customer requested changes to the templates late in the project. The effort involved in re-creating the mapset after making these changes consisted of about two hours processing time. Finally, the solution led to increased quality of the output mapsets by making errors in the database more readily apparent to the customer. Because of the multi-faceted architecture, changes and corrections to the database caused the maps to update dynamically.

Conclusion

The multi-faceted architecture utilized on this project produced significant increases in cost effectiveness, flexibility, and allowed for the timely completion of an otherwise difficult task. The combination of ArcObjects and MS Access provide a very robust toolset. The "Rapid Prototype" development model fit this project well because of the

ambiguity of the requirements at project inception. Using Object Oriented Programming techniques in the coding allowed for increased flexibility allowing the team to work in parallel to produce a timely and cost-effective product for the customer.