Geospatial Multistate Archive and Preservation Partnership (GeoMAPP)

Archival Challenges Associated with the Esri Personal Geodatabase and File Geodatabase Formats

December 6, 2011

Introduction

Spatial databases play a prominent role in geospatial data production and management, storing a range of data types including geographic features, attribute information, satellite and aerial imagery, surface modeling data, and survey measurements. In addition to storing data, some types of spatial databases can also model the relationships between data, handle data validation, and support complex data models, versioning, and multi-user editing, all of which greatly improve data integrity and analysis capabilities. The most robust spatial databases are stored within a server-oriented relational database system rather than as discrete relational database files. **Esri's ArcSDE Geodatabase** is a widely implemented server-oriented spatial database system, while a variety of commercial solutions such as Oracle Spatial, IBM Informix DataBlade, and Microsoft SQL Server make available spatial extensions to relational database management systems (RDBMS). PostGIS is an open source option based on PostgreSQL.

Spatial databases present a range of complex challenges related to long-term data preservation. The whole of these databases is often greater than the sum of the parts in that they are capable of storing not only a large set of datasets but also object relationships, behaviors, annotations, tools, and data models that may span or connect the stored datasets. Due to the complex nature of database structure and content, it may not be possible to extract and transfer individual components of this data into other systems without losing some information.[1] Some of these database systems provide special mechanisms for managing temporal versions of geospatial data forward in time—for example ArcSDE supports **Geodatabase Archiving**[2]—but these databases do not themselves constitute easily exchangeable archival objects.

[1] DPC Technology Watch Report. Preserving Geospatial Data. May 2009. http://www.dpconline.org/component/docman/doc_download/363-preserving-geospatial-data-by-guy-mcgarva-steve-morris-and-gred-greg-janee

[2] Geodatabase archiving was introduced in ArcGIS 9.2 as a means to capture static, historical versions of a Geodatabase in addition to the transactional versioning that was already supported. ArcGIS Resource Center, Desktop 10: Geodatabase Archiving. January 5, 2011. http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//002700000045000000.htm

Spatial Database File Formats

Spatial database file formats, which live in file form independent of a server-oriented RDBMS, are playing an increasingly prominent role in the distribution of data to end users. These formats also provide an option for transferring geospatial content to archives. Within the Esri community two Geodatabase formats have emerged since the release of ArcGIS 8.0: the **Personal Geodatabase (PGDB)**, which emerged as the original format for ArcGIS Geodatabases in 1999, and the **File Geodatabase (FGDB)**, which emerged in 2006. These formats are designed to support the full information model of the Geodatabase, including topologies, raster catalogs, network datasets, terrain datasets, and address locators, etc. Yet these file-based databases are designed to be edited by a single user and provide only very limited support for versioning.

The Geospatial Multistate Archive and Preservation Project (GeoMAPP) has been investigating a variety of issues related to ensuring long-term access to geospatial data. One area of focus concerns the long-term sustainability of Esri Geodatabase content in an archival context, and whether it would be better to convert these databases to open formats or retain them in a particular Esri Geodatabase format. This report will provide an initial assessment of the preservation challenges associated with the Personal Geodatabase and File Geodatabase formats.

Personal Geodatabases

Personal Geodatabases, released in 1999 with the arrival of ArcGIS 8.0, are stored in Microsoft Access (as an **.mdb** file). The database consists of nine system tables in addition to data stored in the following types of datasets: feature class, feature dataset, mosaic dataset, raster catalog, raster dataset, schematic dataset, table (non-spatial), and toolboxes. Feature datasets can contain feature classes as well as the following types of datasets: parcel fabrics, feature-linked annotation, geometric networks, network datasets, relationship classes, terrains, and topologies.[3] Personal Geodatabases are subject to a number of functional limitations, including:

- May only be used in the Microsoft Windows environment.
- An individual database cannot be larger than two GB in size, although the effective database size is smaller, somewhere between 250 and 500 MB, above which the database performance starts to slow down.
- Supports only single-user editing and does not support versioning.

The 2 GB ceiling on PGDB size limits the format's utility as an exchange mechanism. Esri now recommends using File Geodatabases instead, for their scalability, significantly faster performance, and cross-platform use; although Esri has indicated that they will continue to support PGDB for numerous purposes. Some users still prefer the PGDB because they like the table and attribute handling operations they can perform using Microsoft Access.[4]

[3] ArcGIS Resource Center, Desktop 10: What is a Personal Geodatabase? http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//003n0000007n000000.htm

[4] ArcGIS Desktop 9.3 Help: Types of Geodatabases. May 27, 2010. http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=types of geodatabases

Preservation Implications of Personal Geodatabases

Although the Personal Geodatabase will continue to be supported by Esri software for some undefined period of time, the format is problematic as an archival object for a number of reasons:

- It is characterized by a dual dependency on proprietary formats: the Personal Geodatabase format itself and the Microsoft Access database (.mdb) format in which it is stored.
- The 2 GB limit on database size limits the utility of the Personal Geodatabase as a means for exchange of Geodatabase content in archival transfers.
- The stated preference of Esri for the File Geodatabase format suggests that it will be necessary for archives to be vigilant about monitoring software support for Personal Geodatabases over time.
- Retroactive support of earlier versions of Personal Geodatabases by current software can be expected to have limits, requiring that data in older versions be upgraded.
- Databases that have been upgraded to the latest version may not be compatible with earlier versions of ArcGIS that are still in use, possibly requiring that a data provider make legacy versions available for use.

Since the Personal Geodatabase utilizes the .mdb database format it can be read using the OGR data conversion utility, which supports read access via ODBC without using any Esri middleware.[5] The single-file structure of the Personal Geodatabase also provides an advantage in terms of portability (subject to size constraints).

File Geodatabase

The File Geodatabase, released in 2006 in connection with the arrival of ArcGIS version 9.2, was created as a standalone database format that does not require a commercial back-end database. An FGDB consists of a file folder that contains various types of GIS datasets (and supporting components) with each dataset existing as a separate file on disk. This format is now recommended by Esri as the native ArcGIS format for data that is stored in file form.

Esri's goals with the File Geodatabase were to:

- Provide a portable Geodatabase that works across operating systems.
- Scale up to handle very large datasets (up to one terabyte per dataset, with 256 TB allowed for large imagery datasets).
- Improve performance and scalability, for example, to support individual datasets containing well over 300 million features and datasets that can scale beyond 500 GB per file with fast performance.
- Employ an efficient data structure that is optimized for performance and storage, using about one third of the feature geometry storage required by Shapefiles and PGDB, and

[5] OGR Simple Feature Library. Esri Personal Geodatabase. http://www.gdal.org/ogr/drv_pgeo.html

allowing users to compress vector data to a read-only format to reduce storage requirements even further.
- Outperform Shapefiles for operations involving attributes and scale the data size limits well beyond Shapefile limits.

File Geodatabase API

Until recently the internals of a File Geodatabase could only be accessed and manipulated using ArcObjects, the building blocks of licensed Esri software components. In June 2011 Esri released its new API (Application Programming Interface) to the File Geodatabase. The API provides a low level (non-ArcObjects based) means by which advanced developers can gain access to the database without an ArcObjects license, opening up the format to a limited extent to non-Esri applications. The API does not replace ArcObjects as the recommended approach for working with File Geodatabases.

The API makes it possible to create, open, and delete File Geodatabases, and perform a variety of reads, insertions, deletions, edits and queries. The API has a number of stated limitations with regard to support of various dataset types; and raster components "are not supported with the initial release". Spatial queries are limited to the envelope-intersects operator.[6]

Preservation Implications of the File Geodatabase

Some advantages of the FGDB format for preservation include:

- The full information model of the Geodatabase is supported, including features such as relationships, topologies, and data models that are not supported by the Shapefile.
- All information is stored in a directory of files that can scale up to one terabyte of size per dataset (or more for raster data), increasing portability in comparison with ArcSDE (Esri's widely implemented server-oriented spatial database system) and making the format more useful as a method for archival transfers of large data collections than the Personal Geodatabase.
- FGDB is Esri's preferred format for file-based storage of geospatial data and as such benefits from a high level of support in current and coming software products.
- The emergence of an API provides some hope of broader tool support for the format, including by non-Esri software, helping to increase the likelihood of longer-term usability and transparency of data stored in this way.

Some disadvantages of the FGDB format for preservation include:

- An open specification has not been provided, as is the case for Shapefiles, and so there is no benefit from broad software support.
- As a collection of binary files of unknown construction, the FGDB cannot be inspected in the way a Personal Geodatabase can as an .mdb file.

[6] ArcGIS Resource Center, Inside the Geodatabase: File Geodatabase API Details. December 13, 2010. http://blogs.esri.com/Dev/blogs/geodatabase/archive/2010/12/13/File-Geodatabase-API-details.aspx

- The FGDB API, released in June 2011, arrived with a number of well-documented limitations, particularly with regard to various dataset types as well as most raster-related database components.
- The directory structure of the format requires that some care be used when transferring data. Databases may be corrupted if not transferred properly or files are renamed.
- Metadata is not externally accessible without using Esri's ArcCatalog software.

The File Geodatabase shares these disadvantages with the Personal Geodatabase:

- Retroactive support of earlier versions of FGDB by newer versions of ArcGIS software might have limits, requiring that data in older versions be upgraded.
- Databases that have been upgraded to the latest version may not be compatible with earlier versions of ArcGIS that are still in use, possibly requiring that a data provider make legacy versions available for use.

Esri Geodatabase Conversion Options

Esri software offers some options to either export selected datasets or convert entire Geodatabases into other formats.

Options for Conversion of Datasets from Geodatabases

Shapefile

The Shapefile, which emerged in 1995, is commonly encountered as a distribution format. In 1998 Esri released a technical description white paper that provides detailed specifications for reading, writing, and updating Shapefiles, and many software vendors now provide Shapefile read/write options as part of their user interface.[7] The Shapefile is a very stable format, with functionality being added over the years through the creation of "sidecar files" that live outside of the core set of files making up the dataset. Individual feature classes (vector data layers) may be extracted as Shapefiles for archiving as a hedge against long-term failure of a Geodatabase, but limitations of the format may result in degradation or loss of data exported to Shapefile. Shapefiles only store geography and attributes for one set of features, so topological relationships among multiple feature classes cannot be established and stored within their structure. Other Shapefile limitations such as column name width limits enforced by the .dbf format, lack of support for true curves, and problems in field data type conversions may lead to data loss or degradation as a result of the conversion process.

Geography Markup Language (GML)

Individual feature classes may be exported as Geography Markup Language (GML) datasets. GML is an open standard of the Open Geospatial Consortium (OGC), developed as an XML

[7] ESRI Shapefile Technical Description: An ESRI White Paper. July 1998.
http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf

encoding standard for geographic information.[8] Support for Simple Feature GML (a simplified profile of GML) is available to all ArcGIS users, while the ArcGIS Data Interoperability extension provides support for several application schemas of GML (GML variations defined for specific communities of use). While GML and its profiles and application schemas are openly documented, more investigation is needed about the suitability of a particular flavor of GML to be used as a conversion target for archival purposes in the context of Geodatabases. GML exports may be useful as archival objects within particular communities with strong support for particular flavors of GML (a particular flavor of GML may be characterized by some combination of GML version, profile, and application schema).[9]

OGC KML

KML (formerly Keyhole Markup Language), an OGC standard since 2007, is an XML language focused on geographic visualization, including annotation of maps and images.[10] KML is not intended for data analysis and can only capture limited aspects of exported feature data (e.g., attribute information is lost), so the format is not a viable conversion target for preservation purposes.

Raster Formats

Raster datasets may also be extracted from a Geodatabase into ERDAS Imagine, JPEG, TIFF, Esri Grid, and MrSID formats.

Options for Conversion of the Entire Geodatabase

Geodatabase Upgrade

Personal Geodatabases and File Geodatabases may each be upgraded to format versions associated with new versions of ArcGIS. Personal Geodatabases may also be converted to File Geodatabase through mechanisms provided by ArcGIS. Conversions occurring step-by-step through incremental ArcGIS software versions may prove to be less subject to data loss and errors than conversions that involve skipping versions.

In an archival context one might also consider retaining pre-upgrade copies of a Geodabase, but software support for those earlier versions might erode over time.

[8] Open Geospatial Consortium. Geography Markup Language (GML) Encoding Standard. 2007. http://www.opengeospatial.org/standards/gml

[9] Examples of GML implementations for particular communities include the Aeronautical Information Exchange Model (AIXM-GML), developed to meet aeronautical information system requirements, and OS MasterMap, developed by the UK Ordnance Survey in support of national mapping efforts. Open Geospatial Consortium. GML Application Schemas and Profiles. http://www.ogcnetwork.net/node/210

[10] Open Geospatial Consortium. OGC KML Standard. 2007. http://www.opengeospatial.org/standards/kml

ArcGIS XML

Starting with ArcGIS version 9.0 an openly-specified XML schema and an associated XML export option became available for the Geodatabase, providing a means by which to implement interchange mechanisms to make Geodatabase content available in other technical environments. All of the items and data in a Geodatabase, such as domains, rules, feature datasets, and topologies, can be exported into the ArcGIS XML format and validated against the schema. According to Esri documentation, “the objective of the ArcGIS XML exchange tools is to represent the semantics of the Geodatabase constructs completely and unambiguously”. Three XML document types can be created in ArcGIS including:

- A *Workspace* document, which constitutes a complete export of a Geodatabase to XML.
- A *RecordSet* document, which involves export of a single feature class or table as simple features in a RecordSet document.
- A *Data Changes* document, which may be used in a disconnected editing environment to export (check out), modify, and resubmit (check in) changes to a parent database.[11]

The utility of ArcGIS XML as an archival option is uncertain since it is not clear what level of support there will be in future versions of ArcGIS for re-importing XML exports created from previous versions of the Geodatabase. In fact, limited testing has shown that data exported as XML from a version 9.1 PGDB could not be successfully imported to newer versions (see Test 1 below).

Esri Geodatabase Conversion Tests

Various Geodatabase file import, export, and conversion scenarios were tested in order to gain some insight on which of these processes could work successfully, and which processes would fail. The following specific conversion scenarios were tested using various versions of Esri’s ArcMap software, with the stated results:

Personal Geodatabase Tests

Test 1: Export of a PGDB version 9.1 to XML, and then import the XML to a PGDB version 10.

This represents a scenario of having created an XML export file from a PGDB, and needing to access the data several years later by importing the XML to a later, readily available version. The PGDB used is from Richmond, VA and was created October 3, 2003.

> Result 1: This process failed. Importing the same XML back into a 9.1 PGDB was successful, but importing the XML into all higher versions (9.2, 9.3, 9.3.1, and 10.0) failed.

[11] XML Schema of the Geodatabase: An ESRI Technical Paper. June 2008.
http://resources.arcgis.com/content/white-papers?fa=viewPaper&PID=43&MetaID=695

Test 2: Export of a PGDB version 9.2 to XML, and then import the XML to a PGDB version 10.

Result 2: This process was successful.

Test 3: Upgrade a copy of the same PGDB version 9.1 to version 9.3.1 by selecting the PGDB “Properties -> Upgrade Geodatabase.” Then, the upgraded PGDB was exported to XML, and the XML was imported back into a PGDB version 9.3.1. This process is more of a test of the XML export/import capability, since successful conversion outputs a PGDB of the same version as the original.

Result 3: This process was successful.

Test 4: Export a PGDB version 9.1 to Shapefiles in order to examine some specific loss of data.

Result 4: The conversion completed successfully, but sub-domain attribute text was lost in the conversion, leaving some of the attribute data un-usable.

Test 5**:** Convert a PGDB version 9.1 to an FGDB version 10.

Result 5: This process was successful.

File Geodatabase Tests

Test 1**:** Export of an FGDB version 9.2 (the first ArcGIS version that supported FGDB) to XML, and then import of the XML file to a version 10.0 FGDB. Similar to the PGDB Test 1, this represents a scenario of having archived the FGDB data in an XML file, and several years later, needing to import the data into a readily available FGDB version.

Result 1**:** This process was successful.

Test 2**:** Export of an FGDB version 10 to Shapefiles in order to examine some specific loss of data.

Result 2: This process is successful, except subtypes, domains, and relationship classes are not retained in the Shapefile outputs.

GML Tests

Additional tests were conducted of converting PGDB and FGDB to GML (Geography Markup Language) export files, and then importing the GML files back into Geodatabases, using the Data Interoperability Tools -> Quick Export and Quick Import tools with ArcToolbox. Esri does not provide a means to convert the entirety of a Geodatabase to GML. Initially, in the 2009 testing, an FGDB was exported and imported to and from GML, and the conversion failed, but the process was later repeated with success. Most noteworthy with GML conversions are the laborious requirements for setting up the export process: each feature class within a Geodatabase must be selected for conversion, which forces the user to make many mouse clicks. This is not an ideal design for an archivist needing to convert a large number of Geodatabases.

Testing Summary

Early export files in XML and GML formats created from Geodatabases seemed to introduce increased chances of failure when conversions to higher versions were attempted. Testing showed that version 10 of ArcGIS seems capable of importing and upgrading older Geodatabases for full access. Perhaps Esri has fixed bugs in the export process from earlier versions, but a slight lack confidence in relying on these intermediate exchange formats remains, even though they offer the advantage of having openly documented specifications.

Preservation Strategies for Esri Geodatabases

Near-term software support from Esri and other data translation software vendors such as Safe Software appears strong for all of the Esri Geodatabase formats, including for the process of upgrading File Geodatabases to higher versions. Since the nature of long-term software support for any particular data or database format will always be an unknown, archivists will need to plan on making format conversions in the future to whatever new data formats and architectures arise, so that the content being produced today will not be inaccessible and lost.

While the Shapefile is openly specified and much more widely supported than the File Geodatabase, the inability of the Shapefile to support the advanced features found in Geodatabases makes it inappropriate as a format for content that contains those advanced features, though Shapefiles may work as a stable, well-supported archival format in cases where advanced features are not present.

Keeping a Watch on Open Source Approaches

PostGIS, which adds support for geographic objects to the PostgreQSL object-relational database, is the predominate open source server-oriented geospatial database. An emerging open spatial database file format is SpatiaLite, which adds a spatial extension to the SQLite database construction, and offers the advantage of storing data within a single file.[12] Some

[12] SpatiaLite, a Complete Spatial DBMS in a Nutshell. http://www.gaia-gis.it/spatialite/

developers in the open source geospatial software community have embraced SpatiaLite as a database alternative. For example, QGIS allows full read/write access to SpatiaLite files, and other free stand-alone viewers and editors are available. Although Shapefiles can be imported and exported in and out of SpatiaLite files, there is unlikely to be any direct support of SpatiaLite from Esri software anytime soon.

Using the File Geodatabase API for Data Conversion

Community experience with the recently released File Geodatabase API is still limited, and the API comes with a number of well-documented limitations with regard to support of various dataset types (e.g., operations on raster data are not supported). Recently, as part of the Open Geospatial Consortium (OGC) OWS-8 interoperability initiative, Esri and OpenGeo partnered to test the use of the File Geodatabase API to do round-trip transfer of data between the File Geodatabase format and PostGIS.[13] Rather than create a conversion tool that worked directly with the PostgreSQL client API, a utility was created using an extra abstraction layer, the OGR simple features library, with the expected benefit that any work done to support better reading and writing of FGDB data would become part of the widely-used OGR software library.

Working within the well-documented constraints of File Geodatabase API functionality (e.g., limitations on data types supported), some of the issues encountered in this testing included:

- The API does not provide for exchange of explicit topology (although it does allow for the transfer of topology rules for feature classes in a dataset).
- Reading of compressed File Geodatabases is not supported, although there are apparently plans to support reading of compressed data in a future release.
- Reading of datasets in FGDB files was found to be slow at the FGDB-API level.

Accommodating the rich features and complex functionality of the Esri Geodatabase within open source alternatives may be challenging. During OWS-8 bulk transfer tests using OGR a number of conversion issues were surfaced, including:

- A mapping of the FGDB type model to the OGR type model resulted in having many FGDB types mapped to the same FGDB type, with the implication that fidelity will be lost.
- The OGR geometry model only supports the original Simple Features SQL (SFS) model for geometry, whereas the FGDB model supports additional Esri geometry types, including curved components.
- The FGDB table definitions include the concept of sub-types, enumerations that can restrict the values in a column, whereas the OGR abstraction includes no concept of sub-types or domains and does not handle these features.
- The FGDB includes the concept of relationships between tables, which the OGR abstraction does not address.

[13] An Engineering Report titled OGC OWS 8 Bulk Transfer with File Geodatabase is available at http://www.opengeospatial.org/standards/per

Some of the above limitations encountered in the OGR abstraction model may not apply if data were instead converted directly to PostGIS, making use of advanced features available in that environment. Yet these complications illustrate how difficult it can be to recreate complex FGDB components in external environments, raising questions about the viability of open source database formats or platforms as target environments for full archival conversion of File Geodatabase data at the present time.

For Further Investigation

Areas of further investigation related to preservation of Esri Geodatabase content include the following:

- Further evaluation of the archival utility of the File Geodatabase API. Community feedback can be monitored, and any documentation produced in the future will help to inform preservation-related investigations. Since the FGDB API is designed to read whatever XML metadata is provided, one opportunity is to create an archival tool for reading and extracting FGDB metadata.

- Explore best practices for File Geodatabase creation and distribution. The File Geodatabase conversion process involves optional operations which may have an impact on long term accessibility or quality of FGDB content. The disaggregated structure of FGDB objects also requires that special care is employed in transfer of data. Examples such as the British Columbia “File Geodatabase Standards for Data Creation, Publication, and Distribution” should be evaluated to identify possible best practices.[14]

- Compile a set of suggested improvements to the FGDB format to be presented to Esri. Possible topics might include: suggested improvements related to the FGDB API, better support for management and packaging of FGDB files, external manifest documents listing feature classes within an FGDB, and externally accessible metadata, including for the entire FGDB collection.

- Develop a set of recommendations for Esri Geodatabase preservation, as informed by current trends in format support, known strengths and weaknesses of different formats, and findings from conversion testing. (See Appendix I for an example)

[14] GeoBC File Geodatabase Standards for Data Creation, Publication, and Distribution. June 23, 2010. http://ilmb.gov.bc.ca/sites/default/files/resources/public/PDF/Standards/file_geodatabase_standards.pdf

Appendix I: Example Esri Geodatabase Preservation Strategy

In light of current trends in format support, known strengths and weaknesses of different formats, and findings from (limited) conversion testing, one approach to archival handling of Esri Geodatabases is outlined as follows. This approach reflects the recommendations of the Digital Preservation Coalition Technology Watch Report "Preserving Geospatial Data", which suggests a multi-pronged approach to preserving spatial databases.[15]

- In cases where the original production data is maintained in ArcSDE, temporal versions of the data should be managed forward in time within that environment, possibly utilizing tools such as Geodatabase Archiving. However temporal snapshots should also be captured in File Geodatabase format (and possibly *additionally* as Shapefiles) and retained in a separate archive. An example of the ArcSDE to File Geodatabase archival pathway has described for archival operations in the state of Kentucky.[16]

- Maintain data acquired in the File Geodatabase format in that format.

- Geodatabase content received in Personal Geodatabase, ArcGIS XML, GML export, or any other format should be immediately converted to FGDB in order to reveal any potential conversion errors.

- Export metadata into external files for ease of access.

- Upgrade File Geodatabase content to newer format versions as those new versions become established (it may be necessary in some cases to provide access to content in previous versions to support users of earlier ArcGIS versions).

- Optionally, as a hedge against future Geodatabase failure, export feature classes to the more stable Shapefile format. This approach should be considered *in addition* to not *instead of* retaining data in FGDB form, since the Shapefile conversion process may result in data loss.

Data received in Shapefile format will already have accrued any possible loss resulting from conversion, if the Shapefile has in fact been exported from a Geodatabase environment. If the Shapefile has been derived from a Geodatabase, a receiving archive may wish to pursue acquisition of the original Geodatabase data. If the data is in fact actively maintained in Shapefile format by the data producer, then the data may be retained in Shapefile format.

[15] DPC Technology Watch Report: Preserving Geospatial Data. May 2009. http://www.dpconline.org/component/docman/doc_download/363-preserving-geospatial-data-by-guy-mcgarva-steve-morris-and-gred-greg-janee

[16] Archival Procedures for Kentucky's Geospatial Data Assets. May 15, 2008. http://www.geomapp.net/docs/ky_geoarchives_procedures.pdf