Geospatial Multistate Archive and Preservation Partnership (GeoMAPP)

Emerging Trends in Content Packaging for Geospatial Data

December 2, 2011

Introduction

An individual geospatial data resource may be composed of a complex, inter-related set of data files as well as metadata and other supporting file objects, all of which need to be arranged in a certain fashion in order to be understood by the software and the humans that are involved in the exchange, management, and use of the data. In order to facilitate automated exchanges of complex data and avoid costly and error-prone human intervention, two organizing components are needed: a physical and/or logical package to encapsulate and structure data objects, and well-structured metadata or manifest information that is associated with that package. The objectives of this report are to: 1) characterize the role that content packaging is coming to play with regard to geospatial data management and access, 2) document emerging content package types that have appeared in the geospatial community, and 3) explore preservation challenges that may arise when these packages are expected to persist over time or when the packaging process itself results in changes to packaged data.

Characteristics of Content Packages

In addition to one or more data files, an individual geospatial data resource may be accompanied by supporting files such as the following: geo-referencing files, metadata files, display definitions (for symbolization, classification, etc.), licensing information, thumbnail images and other ancillary documentation or supporting files serving various functions in support of the data. Some of these objects may be "sidecar" files which are directly associated with a data format and which store this additional information in a specified manner (e.g., for Shapefiles, required files such as ".shx" and ".dbf" or optional files such as ".sbn" and ".sbx"). Other objects may be independent of the format yet store information that is needed to provide a context for use of the dataset (e.g., layer or project files which define data display).

Physical or Logical Packaging

Physical packaging ensures that required data objects are delivered together by putting them into one or more consolidated files, while logical packaging uses a syntax or document structure (often in XML or plain text) to express an organization of data objects. Physical and logical packaging can be used together or separately. Physical packaging, often embodied in the form of an atomic archive file, frequently in ZIP or TAR format, ensures that any required structure (e.g., subdirectories) for organization of the data is maintained during transfer. Individual physical packaging schemes may require the use of specific file and directory naming schemes and specific directory structures. Logical packaging often takes the form of complex XML wrapper formats that create associations between files (including with external resources that

are not physically present) and may incorporate additional metadata elements that are maintained outside of the referenced data objects.

Structured Metadata

Automated handling and interpretation of a metadata record included within a package may be impeded by ambiguous naming of the metadata record or by ambiguity about the schema and encoding used. In addition, it may be necessary to supplement an existing metadata record with additional technical, administrative, or descriptive metadata. In an archival context, for example, there may be a need to supply additional metadata (e.g., fixity, acquisition history, archival rights) that is not supported by core geospatial metadata standards. It also may be necessary to disambiguate information that already exists within the metadata record in order to streamline record handling. Additional or disambiguated metadata may be structured within a single archive package such as a ZIP file through the use of manifest files--typically text or XML files--the structure of which may follow a format-, software-, or community-specific convention. Structured metadata may also be expressed within a complex XML wrapper that encompasses datasets and associated objects.

Content Packaging in the Broader Community

While there is no standard scheme for content packaging within the geospatial community a variety of approaches have been put into place in other information domains, some of which have implemented content packaging standards involving complex, XML-based wrapper formats. Examples of content packaging specifications or standards include:

- XML Formatted Data Unit (XFDU) – for space data
- Metadata Encoding and Transfer Standard (METS) – for digital libraries
- MPEG-21 Part 2 Digital Item Declaration Language – for digital media
- IMS Content Packaging – for learning technologies
- Material Exchange Format (MXF) – for audiovisual content
- BagIt – for transfers associated with digital curation

The more elaborate content packaging schemes establish relationships between data objects, provide linkages to external resources, and provide a means of encoding different types of metadata associated with the object. XML-based wrapper formats can be quite complex, and any allowance for flexibility in definition of these wrapper objects will tend to lead to a corresponding decline in interoperability across systems.[1]

In practice, archive files (notably ZIP files) commonly function as a more rudimentary content package for exchange of multi-file datasets or groups of related datasets. While the archive formats don't themselves specify mechanisms for adding intelligence about file relationships and functions within a data package, such mechanisms have been defined as needed within specific communities of implementation.

[1] Jerome McDonough. Structural Metadata and the Social Limitations of Interoperability: A Sociotechnical View of XML and Digital Library Standards Development. Balisage: The Markup Conference. August 2008. http://www.balisage.net/Proceedings/vol1/html/McDonough01/BalisageVol1-McDonough01.html

The Use of Archive File Formats in Content Packaging

For convenience in the context of data transfer, **archive** formats (e.g., ".tar") may be used to bundle multiple files into one single file, while **compression** formats (e.g., ".gz") may be used to reduce file size. Archive and compression formats may also be used together (.e.g., ".tar.gz"). Some packaging standards or specifications may suggest the use of an archive format for serializing a set of objects into a single object for transfer without actually requiring that a specific format be used. For example, the BagIt File Packaging Format, a hierarchical file packaging format for the exchange of digital content, suggests but does not require the use of an archive file in a format such as TAR or ZIP, and provides several rules for how objects should be serialized within an archive file if such is used.[2] The BagIt format was used to test archival exchange of geospatial data as part of the Geospatial Multistate Archive and Preservation Partnership project in 2009.[3]

ZIP is one of several formats that implement both archive and compression functions. ZIP stores one or more files that are optionally compressed on a per file basis. The file extensions .zip or .ZIP are used and the format is associated with mime type *application/zip*. Zip supports several compression algorithms, although the most commonly used method is DEFLATE, which is described in IETF RFC 1951.[4] ZIP was originally created in 1989 and first implemented in PKWARE's PKZIP utility as a replacement for the earlier ARC compression format. One constraint imposed by ZIP is a 4 GB size limit. ZIP64 format extensions greatly expand capacity, although support is still somewhat limited and more often found in recent software releases.[5]

ZIP and Standards

The ZIP specification is actively maintained (with support from interested industry experts and users) and openly published by the firm PKWARE. While openly documented, parts of ZIP are covered by patents or pending patents. It often comes as a surprise that ZIP is not an open standard since it provides the basis for modern packing standards or specifications such as JAR, WAR, EAR, SCORM, ODF, OpenDocument, and Office Open XML. In order to ensure consistent use, these standards may define constraints on how ZIP may be used. For example, Office Open XML describes a detailed profile that requires use of DEFLATE compression and disables all of the advanced features of ZIP. ZIP-dependent standards reference the ZIP technical specification, the ".ZIP Application Note", maintained by PKWARE.[6] The specification

[2] California Digital Library. BagIT File Packaging Format v. 0.96. June 24, 2009. https://confluence.ucop.edu/display/Curation/BagIt

[3] Geospatial Multistate Archive and Preservation Partnership (GeoMAPP) Interim Report: 2007-2009. http://www.geomapp.net/docs/GeoMAPP_InterimReport_Final.pdf

[4] IETF Network Working Group. DEFLATE Compressed Data Format Specification version 1.3. http://tools.ietf.org/html/rfc1951

[5] ZIP64 format extensions have been in use for ESRI's LPK and MPK formats since ArcGIS version 10

[6] PKWARE. .ZIP Application Note. http://www.pkware.com/support/zip-app-note/

has evolved from the original APPNOTE.TXT file delivered with early versions of the PKZIP software.

In September 2010 ISO/IEC JTC 1 approved initiation of "a study period with the aim of establishing a firmer rationale for standardization of aspects of the ZIP format."[7] Such a standardization effort, if pursued, might result in definition of a ZIP-compatible format--perhaps involving only a subset of ZIP functionality--that is suitable for use with existing standards that build on top of ZIP.

Content Packaging Formats for Geospatial Data

Within the geospatial community archive formats such as ZIP or TAR commonly function as rudimentary content packages for multi-file datasets or groups of related datasets. These packages often lack any consistently structured information about file relationships and functions within the data bundle. However, formalized approaches to the use of ZIP files with geospatial data have emerged. This report addresses four such uses of ZIP as part of formalized archive file formats, each of which have been developed to address specific needs within a particular domain rather than to serve as a generalized solution:

- KMZ – for packaging a specific file format (KML)
- MEF– for packaging metadata and data in connection with specific geoportal software (GeoNetwork)
- LPK – for packaging data with display information within a suite of a specific vendor's software tools (various ESRI software packages or online tools)
- MPK – for packaging data and finished maps within a specific desktop GIS tool (ESRI's ArcGIS)

Although each of these formats addresses specific packaging problems within the geospatial domain, the examples provide some insight into preservation opportunities and challenges related to content packaging.

Example 1: KMZ

KMZ (KML-Zipped) are ZIP files containing one or more KML files which have been pulled together into a single compressed, archive file to make them easier to distribute and share with multiple users. In addition to KML files, a KMZ file may include ancillary files such as custom icons or images (such as those used in placemark descriptions or in overlays) so as to eliminate the need to link to those files through an internet connection. KMZ files may also include limited 3D model data exported from Google Sketchup or ArchiCAD as .SKP files.

File extension	.kmz (recommended) KMZ has a MIME type: application/vnd.google-earth.kmz

[7] ISO/IEC JTC 1/SC 34. Call for Participation in Study Period for "Zip" Format. September 11, 2010. http://www.itscj.ipsj.or.jp/sc34/open/1503.htm

Origination	Early 2000's by Keyhole, which created the product that eventually became known as Google Earth (Keyhole was purchased by Google in 2004).
Transparency	While KML is an open standard, the maintenance of which is governed by the Open Geospatial Consortium industry standards body, KMZ is not defined within the KML standard. KMZ however, builds on the openly documented ZIP format, and the KMZ format itself is documented on Google's website.[8]

Technical Description

A KMZ file consists of a main KML file and zero or more supporting files (additional KML files or ancillary files) that are packaged and compressed with a ZIP utility into one archive file. The compression ratio for the entire package depends on the nature of the content, with packages containing predominately KML (text) content achieving higher compression ratios than packages containing predominately images. KMZ files must be compatible with the DEFLATE compression method of ZIP or the package may not uncompress in some software packages that encounter the KMZ file.

The contents of a KMZ file include:

- A **single root KML document**, typically named "**doc.kml**", placed in the root directory of the ZIP package. Google Earth, the primary client for KMZ packages, will read the first KML file found in the root of the ZIP document and use that KML file as the root document regardless of file name. Since it may not be possible to control order of KML files in some ZIP creation tools, common convention is to place only the root KML document (commonly, though not by requirement, with the name 'doc.kml') in the root of the ZIP archive, with additional referenced KML files placed in subdirectories.

- Optionally, any overlays, images, icons, and 3D models referenced in the KML documents.

KMZ Use

KMZ files are used in connection with KML files in cases such as the following:

- When a number of KML and supporting files need to be packaged together for distribution.
- When file-size reduction through compression will be beneficial to data transfer.
- In order to create a self-contained package of content that will not depend upon an internet connection for content rendering.

[8] KMZ Files. Google. http://code.google.com/apis/kml/documentation/kmzarchives.html

Methods of KMZ Creation

- Within Google Earth it is possible to save a placemark or folder as KMZ File.
- Using various ZIP utilities.
- Various GIS and online mapping tools support creation of KML and KMZ files.

Methods of Interpretation

- A KMZ file maybe be opened and inspected by any software tool that can read ZIP files.
- A variety of geo-browser tools and GIS or online mapping environments support KMZ. For instance Google Earth will unzip a KMZ file, separating the main KML file and its supporting files into their original directory structure with their original filenames and extensions.

Preservation Implications

KMZ files which are entirely self-contained (i.e., do not rely on **network links** to reference external content across the internet) provide a stable means of capturing and retaining interconnected sets of files. To the extent that KML files within a KMZ file reference external files, the overall integrity of the KMZ file might be expected to degrade over time as externally referenced resources are relocated or removed from the network. Although KML is itself an international standard, the fact that the structure of KMZ files is not explicitly defined within that standard raises potential issues for KMZ compatibility across software environments and over time.

Resources

Google Code: KMZ Files
http://code.google.com/apis/kml/documentation/kmzarchives.html

Tutorial: Packaging Content in a KMZ File
http://earth.google.com/outreach/tutorial_kmz.html

Example 2: Metadata Exchange Format (MEF)

The Metadata Exchange Format is a ZIP-based file format designed to facilitate exchange of metadata and, optionally, associated data between different software and catalog environments. MEF was developed specifically for use with the open source GeoNetwork portal software in mind as a point of exchange, yet it could act as an interoperability format between any environments that support MEF. While the central function of MEF is to facilitate metadata exchange, the format also facilitates the exchange of associated data, supplementary content (e.g., preview or thumbnail images), and supplementary metadata of a technical or administrative nature (e.g., rights information). MEF provides an explicit structure for representation of the full metadata by standardizing the metadata file name ("metadata.xml"), requiring specification of metadata schema, and providing a mechanism for listing additional metadata as part of a manifest. The actual data and ancillary files are addressed less explicitly. These files are listed in the manifest and contained in separate subdirectories, but the relationships between these data objects are not addressed by the MEF format.

File extension	.mef (recommended)
Origination	2007, initially created by the UN Food and Agriculture Organization (FAO) for use with the GeoNetwork open source geoportal software.
Transparency	MEF is openly documented and has--by association with the GeoNetwork software--been moved under the auspices of the Open Source Geospatial Foundation (OSGeo), which now oversees GeoNetwork development.

Technical Description

An MEF file is a structured ZIP file which contains the following files and directories:

- **metadata.xml** (root directory file): Contains the metadata itself, in XML format, according to the encoding specified in the *info.xml* document.
- **info.xml** (root directory file): A manifest file which contains administrative, technical, and descriptive information that is related to the metadata but that cannot be stored in it.
- **Public** (directory): Stores the thumbnails images and other supporting files which are ancillary in nature and which are intended to be as publicly available as the metadata is. MEF does not define restrictions on the choice of image formats for thumbnails, though PNG, JPEG, or GIF are strongly recommended as a matter of practice.
- **Private** (directory): this is a directory used to store all geospatial data associated to the metadata. Files in this directory are *private* in order to allow a catalog environment to restrict access to data that is subject to restrictions on use. MEF does not define restrictions on the file types that can be stored in this directory.

Additional files or directories may be placed within MEF files, with the expectation that software would ignore those additional files unless support has been added by a specific community for custom extensions to MEF.

An MEF file can have empty public and private folders depending on the export format, which can be:

- **Simple** : both public and private are omitted (only metadata is included)
- **Partial** : only public files are provided (metadata and ancillary files included)
- **Full** : both public and private files are provided (metadata, ancillary files, and associated data included)

The info.xml file contains a variety of information related to the metadata record and associated data including, in brief:

- Metadata version information and a universally unique identifier (UUID)
- Date of creation and date of most recent change
- Information about metadata creator
- Indication of the metadata's schema (e.g., *iso19115*)
- Metadata rating and popularity information

- Category information
- Privileges (or rights) information and a variety of other administrative metadata elements
- Listings of files found in the *public* and *private* directories

The info.xml file may also be extended with custom attributes or sub-trees within the XML document if support for such extensions is developed within a particular community.

When to Use MEF

MEF was specifically designed to support the following functions for the GeoNetwork portal environment and supporting tools:

- Export a metadata record and associated data for backup purposes
- Import a metadata record and associated data from a previous backup
- Import a metadata record and associated data from a different GeoNetwork version to allow a smooth migration from one version to another

MEF has also come to provide an exchange format between some desktop software environments and GeoNetwork.

Methods of Creating MEF Files

MEF files may be generated by the GeoNetwork portal software or by any custom tool that can construct appropriately structured ZIP files. MEF files may be generated by some desktop GIS software environments, notably the ArcCatalog software of ESRI's ArcGIS desktop environment (using an available plug-in) or the gvSIG open source GIS desktop software.

Methods of Interpretation

MEF files may be read into GeoNetwork as part of data and metadata transfers. MEF files may also be examined using any tool that supports ZIP.

Preservation Implications

Although MEF was explicitly developed for use in connection with the open source GeoNetwork catalog environment, it presents an interesting model for using ZIP as the basis for a formalized packaging of geospatial metadata as well as associated data and ancillary components in a way such that automated exchange of metadata and associated data is supported.

Resources

Metadata Exchange Format v1.1 Documentation:
http://geonetwork-opensource.org/manuals/2.6.1/developer/mef/index.html

GeoNetwork Developer Manual v2.6.1:
http://geonetwork-opensource.org/manuals/2.6.1/developer/index.html

Example 3: ESRI Layer Packages

A layer package (LPK) is a file format introduced in ArcGIS 9.3.1 to encapsulate the data, cartography, and other properties of a layer that has been authored in ArcMap or ArcGlobe into one package that can be shared with other ArcGIS desktop users, ArcGIS Online, and ArcGIS Explorer. A layer package includes both the layer properties and the dataset referenced by the layer, making it possible to save and share everything about the layer, including symbolization, pop-up behavior, and labeling. "Notes" that have been created within ArcGIS Explorer may also be shared as layer packages. When saving a map layer, a layer author can choose to include its source data and other intrinsic properties, such as thumbnail, extent, and spatial reference, as part of a layer package.

File extension	.lyr
Origination	2009, by ESRI for use with ArcGIS 9.3.1 or higher and related ESRI software tools
Transparency	The LYR format is maintained by ESRI. As ZIP files, LPK files may be open and inspected using widely available tools, but associated LYR files are binary files stored in a format that is not openly specified.

Technical Description

An LPK file is a compressed ZIP file that contains a root file and two or more directories:

- **.lyr** (root file), a binary file which contains specifications for the presentation of datasets. Such specifications include color shading, naming, and label properties such as font, color, and placement.
- One or more data subdirectories. If the option to convert source data to File Geodatabase format is not checked then separate directories will be created. One directory named for the **File Geodatabase version number** is created to hold data already in File Geodatabase format, and another directory called **commondata** is created to hold other data sources. [*NOTE: There may be differences between version 9.3.1 and version 10.0 implementations of the data directory structure. There may also be differences in options for LPK construction for ArcObjects vs. ArcGIS interface-based creation of packages. This will require future exploration.*]
- **esriinfo** (subdirectory), which contains an item.pkinfo file and an iteminfo.xml file as well as an additional subdirectory containing a thumbnail image in PNG format.

The **iteminfo.xml** file contained within the esriinfo subdirectory includes a variety of information including: layer name, unique ID, descriptive information and keywords, bounding coordinates, and spatial reference information. This information is either automatically populated during the LPK authoring process or is added by the author.

The **item.pkinfo** file also contained within the esriinfo subdirectory is a brief XML file that provides information to be used in connection with the pkinfostylesheet.xsl style sheet hosted by ArcGIS Online. *[NOTE: this file appears to provide package name and location information to*

be displayed in connection with ArcGIS Online as well as providing some functionality in connection with ArcGIS File Handler utility. This requires additional exploration.]

Layer packages may be constructed for individual datasets or for **group layers**, which may be used when multiple data layers have been cartographically designed to work together.

When to Use Layer Packages

Layer packages are specific to ESRI software tools, and can be used to bundle data with associated representation information. This bundling allows users to add layers directly into their ESRI software tools for display without having to know how to access the associated data or define data representation.

Methods of Creation

LPK files are created through an authoring process in ArcMap. The authoring process uses existing symbolization (i.e., same cartography and legend as in ArcMap), and allows some customization in terms of dataset naming, selection of attribute fields to display, and changing of display properties. Layer packages do not support schematics or tool layers.

The package creation process includes a number of choices that will affect the nature of the data encapsulated within the resulting package:

- Data may be optionally converted to the File Geodatabase format (version optional).
- The spatial extent of the encapsulated data may be constrained to a specified area.

Other data conversions or changes that are forced in the package creation process include:

- When consolidating or packaging layers, the resulting layers will be renamed using a numbered sequence system.
- Personal Geodatabase data is always converted to File Geodatabase format.
- ADRG, CADRG/ECRG, CIB, and RPF raster formats will always convert to Geodatabase rasters.

Methods of Interpretation

LPK files may be read by a variety of ESRI software tools, including within the ArcGIS suite that includes ArcMap, ArcGlobe, ArcScene and ArcGIS Explorer. The ArcGIS File Handler utility allows an end user to set behaviors so that a particular application is always launched when an LPK file is launched. An LPK file may be opened and inspected using any tool that supports ZIP files, although the LYR file stored in the root of the package is a binary file (not supported by an openly documented specification) that may only be read with ESRI software.

Preservation Implications

The LPK format provides a means to capture and exchange cartographic representations associated with a dataset and as such may be beneficial to efforts to make cartographic representations more persistent, even through data exchange. However, the LPK packaging format is very new (originating in 2009) so there is no track record from which to assess

likelihood of consistent, long-term support for this package format.[9] The item.pkinfo (text) file and the iteminfo.xml file are openly readable but the LYR files at the root of the LPK files are binary and are not based on an openly documented format.[10] The associated data will be subject to whatever preservation risks are associated with that data format. Since LPK creation may involve required and optional data conversions (e.g., to File Geodatabase), the LPK's role in an archival context would be to preserve the representation, while preservation of the original data would need to be addressed separately. Ongoing evolution of both the LPK and File Geodatabase formats may create complications for support of package versions over time.

Resources

A Tutorial for Creating Good Layer Packages
http://blogs.esri.com/Info/blogs/arcgisexplorerblog/archive/2009/06/11/A-tutorial-for-creating-good-layer-packages.aspx

ArcGIS Resource Center, Desktop 10: Package Layer (Data Management)
http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/Package_Map/001700000q5000000/

Example 4: ESRI Map Packages

A Map Package (.mpk) is a packaging format introduced with ESRI's ArcGIS 10.0 to facilitate sharing of complete map documents by packaging an ESRI map document (.mxd) along with the data and models referenced by the layers that it contains into one compressed, portable file. Relative pathnames are used to ensure portability of data links within the MXD document.

File extension	.mpk
Origination	2010, by ESRI for use with ArcGIS 10.0 or higher
Transparency	The MPK format is maintained by ESRI. As ZIP files, MPK files may be opened and inspected using widely available tools, but associated MXD files are binary files stored in a format that is not openly specified.

[9] ESRI's online technical documentation indicates that the compression technology used by ArcGIS layer packages was upgraded in ArcGIS SP1 to remove certain internationalization and size limitations, resulting in some backward compatibility problems. In a technical session at the 2011 ESRI International Users Conference it was confirmed that the compression upgrade involved adding support for ZIP64 extensions.

[10] A Google search on the phrase "ESRI .lyr file" yields as first hit the NCSU Libraries Geospatial Data Formats web page rather than any ESRI website describing LYR.

Technical Description

An MPK file is a ZIP file containing the following directories:

- **esriinfo**, which is structured the same as the esriinfo directory of an LPK file (includes an item.pkinfo file and an iteminfo.xml file as well as an additional subdirectory containing a thumbnail image in PNG format.)
- One or more data subdirectories containing the MXD file and associated data. If the option to convert source data to File Geodatabase format is not checked then separate directories will be created. One directory named for the **File Geodatabase version number** is created to hold data already in File Geodatabase format, and another directory called **commondata** is created to hold other data sources. [*NOTE: There may also be differences in options for MPK construction for ArcObjects vs. ArcGIS interface-based creation of packages. This will require future exploration.*]

The **iteminfo.xml** file is similar to the iteminfo.xml file in an LPK file except that there are additional elements: “tags”, and “summary”.

The **item.pkinfo** file is similar to the item.pkinfo file in an LPK file except that instead of an <lpkinfo> element with nested layer names there is a <documenttypes> element which lists “mxd” as a document type.

When to Use ESRI Map Packages

To encapsulate data associated with an MXD map document file (project file) for purposes of exchange, or to create an archive of a particular map that contains a snapshot of the current state of the data used in the map.

Methods of Creating ESRI Map Packages

MPK files may currently only be created in ArcGIS Desktop 10. MPK package creation involves a set of required or optional data conversions (e.g., conversion to File Geodatabase format) that are similar in nature to those described in connection with the LPK creation process.

Methods of Interpretation

MPK files are read directly by ArcGIS version 10 (the latest version as of December 2011). As ZIP files, MPK files may be inspected with typical ZIP tools, though the MXD file within the package is a binary file that may only be read by ESRI software.

Preservation Implications

The ability of an MPK file to encapsulate and make portable the complex set of components such as data and object relationships that are necessary to maintain the viability of an MXD document suggests some possible utility for use of the MPK to make data representations and GIS project outputs more persistent. In fact, ESRI’s online help suggests the use of the format “to create an archive of a particular map that contains a snapshot of the current state of the data

used in the map."[11] The MPK packaging format is very new (originating in mid-2010) so there is no track record from which to assess likelihood of consistent, long-term support for this package format. The item.pkinfo (text) file and the iteminfo.xml file are openly readable, yet the MXD files stored within the MPK files are binary and are not based on an openly documented format. The associated data is encapsulated in the form from which the map was made and will be subject to whatever preservation risks are associated with that data format. Since MPK creation may involve required and optional data conversions (e.g., to File Geodatabase), the MPK's role in an archival context would be to preserve the representation, while preservation of the original data would need to be addressed separately. Ongoing evolution of both the MPK and File Geodatabase formats may create complications for support of package versions over time.

Resources

ArcGIS Resource Center, Desktop 10: Package Layer (Data Management)
http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//0017000000q4000000.htm

Ongoing Developments

A major challenge of geoarchiving lies in finding efficient means of reliably transferring large amounts of complex data between agencies. The issue of bulk data transfer and associated packaging needs was a key thread in the OGC Web Services Initiative, Phase 8 (OWS-8) testbed activity in 2011, a component of which was focused on advancing the state of geospatial data sharing and synchronization.[12] The scope of OWS-8 experiments included data exchange both via the network and using hard media via "sneaker net" methods. Key technical requirements for content packaging in the outlined experiments included: no data loss in translation, segmentation of the container to allow different components to be extracted or ignored, file-size minimalization (including by compression), and assurance of integrity in content transmission both at the container and component level. OWS-8 included an experiment involving data exchange using an approach called Geodata Bulk Transfer, which defines a container format that is ZIP compressed and contains feature data, schema, and topology as well as metadata.[13]

[11] ArcGIS Online: Common Questions. http://www.esri.com/software/arcgis/arcgisonline/common-questions.html

[12] Open Geospatial Consortium, Request for Quotation (RFC) and Call for Participation (CFP), OGC Web Services Initiative – Phase 8 (OWS-8): Annex B, OWS-8 Architecture.
http://portal.opengeospatial.org/files/?artifact_id=41689

[13] OGC OWS-8 Bulk Geodata Transfer Using GML: Engineering Report:
https://portal.opengeospatial.org/files/?artifact_id=46679

Conclusions

ZIP as a Physical Package

ZIP is widely used as a simple physical container for geospatial data exchange, either in an ad hoc matter or according to some community-, format-, or software-based specification of manifest information and folder structure. The absence of an open standard for some subset of ZIP format might raise long term problems for support of packages based on ZIP unless the package formats in question place restrictions on which ZIP features may used. By comparison with complex XML wrapper formats, ZIP-based packages tend to be relatively simple in nature, leaving any complexity in data relationships to be handled by data formats themselves. This simplicity may increase the likelihood of broad and persistent support for the physical package, especially in a domain such as geospatial that is characterized by great diversity in content types, by leaving preservation of the encapsulated data as a separate issue.

Packaging of Data Representations

Content packaging is being used to capture and make persistent cartographic representations and other data representations whether at the individual dataset level or the level of a data project encompassing many datasets. These packaging methods may help to exchange and preserve data representations in the near term, but the lack of transparency about how these package formats are specified raises questions about long-term sustainability. Since the creation processes of these packages may allow for or require the conversion of some data types, the package's role in an archival context would be to preserve the representation, while preservation of the original data would need to be addressed separately.

Proprietary Formats within Otherwise Transparent Packages

While the packages themselves may be transparent (such as in the case of using a widely supported subset of ZIP functionality), long-term viability of these packages depends upon ongoing software support for individual components--such as data files, cartography files, or project file--that might provide core functionality within the package.

External Data Dependencies

There may be dependencies on external resources (.e.g., network links in KML files, or data service connections in MPK files) that are ephemeral in nature. In cases where content packages are expected to be viable over a long period of time it will be necessary to make a concerted effort to limit the number of external dependencies. The authoring processes for some package formats provide an optional means to draw in and encapsulate data which might otherwise have remained external to the package. To the extent that content packages are created with the intent to support long term preservation, it may be useful to identify best practices for creation of packages which are intended to be of persistent value.

Validation of Data within Content Packages

None of the geospatially-focused packaging formats that were examined make use of checksums for validating data as part of a transfer, and general purpose archival software tools do not place checksum files directly into archival files.[i] Data validation may need to be enforced

through additional measures. In the broader community, one example of a content packaging scheme that incorporates checksums for validation is BagIt, which uses a manifest to list every file in the payload together with its corresponding checksum.

[i] During the OWS-8 Bulk Geodata Transfer experiments conducted by the Open Geospatial Consortium, program participants conducted an investigation and found that none of the archival software packages examined included a checksum internally in their zip files, although 7zip does allow the generation of checksums at a folder level. OGC OWS 8 Bulk Transfer with File Geodatabase. September 30, 2011. https://portal.opengeospatial.org/files/?artifact_id=45754