University of Redlands

**A Prototype Method for Storing Symbols for Multiple Maps in a Single
Geodatabase Using ArcGIS Cartographic Representations**

A Major Individual Project submitted in partial satisfaction of the requirements
for the degree of Master of Science in Geographic Information Systems

by
Peter Michael Kasianchuk

Aileen R. Buckley, Ph.D., Chair
Mark Kumler, Ph.D.
Charlie Frye, M.A.

May 2007

A Prototype Method for Storing Symbols for Multiple Maps in a Single Geodatabase
Using ArcGIS Cartographic Representations

Copyright © 2007
by
Peter Michael Kasianchuk

The report of Peter Michael Kasianchuk is approved.

Mark Kumler, Ph.D.

Charlie Frye, M.A.

Aileen R. Buckley, Ph.D., Chair

April 2007

ACKNOWLEDGEMENTS

I would like to acknowledge my wife Colleen, for her support and patience during the time it took to complete this work. She nursed both me and our second daughter through this period (my weaning has taken longer), and acted as a single parent to Ava and Darcy for far too many bedtimes and weekends. I hope I can be as supportive when she resumes her academic career.

I would like to thank Dr. Karen Kemp for her strong support in allowing me the opportunity to complete this program on a part-time basis. Without her initial and enthusiastic “yes”, I would still be just a lowly Bachelor of Arts. Theresa Ellis was invaluable in helping me navigate the mysterious intricacies of completing the program requirements while a member of five different cohorts.

I would like to thank colleagues at ESRI for their feedback and consultation: Tim Daly, Paul Hardy, Robert Jensen, Thierry Kressmann, Clint Loveman, Tim Ormsby, Edie Punt, and Natalie Vines.

I would like to thank Dr. Aileen Buckley and Charlie Frye for their patience throughout the lengthy gestation of this project. Their insight and perspective were extremely valuable to my learning, and it was a great privilege to work with them.

Finally, I cannot fully express my gratitude to my parents for their unfailing support and encouragement. “Here’s to an early completion” was a toast used many times over the many months it took me to complete the program. I would like to thank my father for providing me with the goal I used as my main motivation when the work became challenging and the end seemed out of sight: the image of him posing with his young family at his Ph.D. graduation from UC Berkeley in 1968. I was able to reproduce that image with my own family in Redlands in 2007.

ABSTRACT

A Prototype Method for Storing Symbols for Multiple Maps in a Single Geodatabase Using ArcGIS Cartographic Representations

by
Peter Michael Kasianchuk

ArcGIS 9.2 software, released in late 2006, introduced a new way for ESRI users to store symbology in the geodatabase. This new method, called cartographic representations, presents new challenges for those individuals involved in producing high-quality maps from the GIS. These challenges include developing new workflows which incorporate the new technology.

The project methodology used an existing geodatabase and a test set of hard copy maps as a base from which to develop a prototype methodology to implement cartographic representations.

The main purpose of the project was to discover how feature symbols for multiple map products could be stored within a single geodatabase. In the course of the research, new techniques and functionality available with cartographic representations were evaluated against the standard ArcMap symbol management tools.

Table of Contents

1.	Introduction.....	1
1.1.	Client.....	1
1.2.	Cartographic Representations defined	2
2.	Project Background.....	5
2.1.	Problem Statement	6
2.2.	Scope of Work	7
2.3.	Project Deliverables	7
3.	Project Methodology.....	9
3.1.	Identify test maps.....	9
3.1.1.	National Park Service (NPS) map.....	9
3.1.2.	Digital Raster Graphic (DRG) map	11
3.1.3.	TOPO map	11
3.1.4.	TOPO_OLD map.....	13
3.2.	Construct maps with ArcGIS	15
3.3.	Compile test geodatabase.....	15
3.4.	Visually evaluate maps for use with cartographic representations	16
3.5.	Implement cartographic representations	16
3.6.	Evaluate results	16
3.7.	Consolidate project geodatabase.....	17
3.8.	Produce final maps.....	17
4.	Implementing Cartographic Representations.....	19
4.1.	NPS map	19
4.1.1.	Gradient line symbol for park boundary	20
4.1.2.	Multiple marker symbols for park facilities.....	24
4.1.3.	Single marker symbols for park facilities	25
4.1.4.	Dashed line intersections for trail symbols.....	29
4.1.5.	Cased line symbol for roads.....	30
4.1.6.	Polygon fill pattern for bogs	33
4.1.7.	Control points for trail dashed line symbol.....	35
4.1.8.	Rotation angle for hydrologic marker symbols	38
4.1.9.	Background raster layers.....	43
4.1.10.	NPS map summary	43
4.2.	DRG map	44
4.2.1.	Building markers aligned to roads	44
4.2.2.	Masking layer for road symbols	46
4.2.3.	New PLSS line symbol	49
4.2.4.	Background vector polygon layer for vegetation cover.....	53
4.3.	TOPO map	55
4.3.1.	Remove selected road feature representations from display.....	56
4.3.2.	Warp tool for bathymetric contour lines	57
4.3.3.	New marker symbols for park facilities.....	59
4.3.4.	New park boundary symbol.....	60
4.3.5.	Background vegetation cover layer	62
4.4.	TOPO_OLD map.....	62

4.4.1.	New masking layer for roads	62
4.4.2.	Remove road features from display	63
4.4.3.	Raster hillshade	64
4.5.	Annotation.....	64
5.	Summary of Cartographic Representations and Final Geodatabase.....	67
5.1.	Summary of Cartographic Representation techniques used in the project	67
5.2.	Final geodatabase summary	69
6.	Discussion	71
6.1.	Advantages of implementing cartographic representations	72
6.2.	Final geodatabase design	72
6.3.	Feature class attributes for cartographic representations	72
6.4.	Cartographic representations in the multi-user geodatabase environment	73
6.5.	Data capture to support cartographic representations	74
7.	Conclusion	75
8.	References.....	77
	Appendix A: Valid Field Types for Cartographic Representation Properties	79

Table of Figures

Figure 1-1 - Schematic illustration of cartographic representation geodatabase table relationships.	3
Figure 3-1 - NPS map	10
Figure 3-2 - DRG map	11
Figure 3-3 - TOPO map	12
Figure 3-4 - TOPO_OLD map	14
Figure 4-1 - Index map of NPS use cases for cartographic representations.	20
Figure 4-2 - Original NPS Park Boundary buffered polygon symbol.	21
Figure 4-3 - Park Boundary layer representation properties for stroke symbol layers.	22
Figure 4-4 - NPS Park Boundary with cartographic representations.	23
Figure 4-5 - Cartographic Representation stroke symbol with gradient fill.	24
Figure 4-6 - NPS map font and symbol errors computer.	25
Figure 4-7 - Marker without background (left) and with background (right).	26
Figure 4-8 - Representation marker editor, showing original font-based glyph.	26
Figure 4-9 - Final representation marker as drawn on the map.	27
Figure 4-10 - Unique representation rules for ‘stacked’ markers	28
Figure 4-11 - Representation marker layers used to symbolize the three park locales.	29
Figure 4-12 - Standard trail line symbols: line ends have dashes of varying length.	30
Figure 4-13 - Trail feature representations: black lines end with a half dash.	30
Figure 4-14 - Cased line caps with the default setting to use Butt line caps.	31
Figure 4-15 - Cased line caps with Round line caps.	31
Figure 4-16 - Gap in data exposed with Butt line cap.	32
Figure 4-17 - Gap in data closed with Round line cap.	32
Figure 4-18 - Source map excerpt showing original bog marker symbol.	33
Figure 4-19 - Bog feature drawn with a simple marker fill pattern	34
Figure 4-20 - Final bog representation symbol.	35
Figure 4-21 - Mount Scott trail symbol prior to editing feature representation.	36
Figure 4-22 - Boat launch trail symbol prior to editing feature representation.	36
Figure 4-23 - Mount Scott trail with feature representation control points added.	37
Figure 4-24 - Boat launch trail after control points added.	38
Figure 4-25 - Geographic, arithmetic, and graphic rotation values.	39
Figure 4-26 - Spring markers with incorrect rotation.	40
Figure 4-27 - Spring markers with correct rotation using new graphic rotation values. ..	40
Figure 4-28 - Default settings for ‘Falls’ marker symbol layer.	41
Figure 4-29 - Angle parameter is now set to an attribute field	42
Figure 4-30 - Spring marker with selected cartographic representations removed.	43
Figure 4-31 - Non-rotated building representation marker symbols.	45
Figure 4-32 - Building representation markers after using the ‘Align Marker to Stroke or Fill’ geoprocessing tool.	46
Figure 4-33 - Initial road line symbol.	47
Figure 4-34 - Road mask layer created with Feature Outline Masks geoprocessing tool drawn in grey.	48
Figure 4-35 - Layer mask applied with Data Frame Advanced Drawing Options.	49
Figure 4-36 - PLSS polygon boundary overprint	50

Figure 4-37 - PLSS lines with incorrect line caps.	51
Figure 4-38 - USGS topographic symbols for PLSS features	52
Figure 4-39 - PLSS lines with Cut Curve geometric effect applied.	53
Figure 4-40 - DRG map vegetation cover drawn from raster data source.....	54
Figure 4-41 - DRG map vegetation cover drawn from vector data source.....	55
Figure 4-42 - Mazama Village detail showing roads with no representation rule.....	57
Figure 4-43 - Mis-aligned bathymetric contours	58
Figure 4-44 - Use of the Representation Warp Tool.....	59
Figure 4-45 - Park Boundary symbol as stored in TOPO.style file	61
Figure 4-46 - Park Boundary symbol imported as representation rule.....	61
Figure 4-47 - Mazama Village detail showing roads with no representation rule.....	63
Figure 4-48 - Rim Village detail showing roads with no representation rule applied.....	64
Figure 5-1 - ArcCatalog Trees showing changes to project geodatabases.	70

List of Tables

Table 5-1 - Summary of cartographic representations used on all test maps.	67
---	----

List of Acronyms and File Extensions

BLOB	Binary Large Object
CD	Compact Disk
ESRI	Environmental Systems Research Institute, Inc.
DRG	Digital Raster Graphic
.gdb	ESRI File Geodatabase Format
GIS	Geographic Information System
.jpg	Joint Photographic Experts Group Image File Format
.lyr	ESRI Layer File
.mdb	ESRI Personal Geodatabase for Access Format
.mxd	ESRI Map Document
NPS	National Park Service
PLSS	Public Land Survey System
.sid	LizardTech MrSID Image File Format
.style	ESRI Style File Format
USGS	United States Geological Survey

1. Introduction

ArcGIS 9.2 software, released in late 2006, contains a geodatabase model enhancement and software tools for defining and storing feature symbology that are collectively called “cartographic representations”. One of the key benefits of this new technology is that it provides ESRI users with the potential to store symbology for multiple maps within a single geodatabase. The client in this project identified a need for an evaluation of this new functionality in an applied context in order to support ESRI users in implementing this technology into their existing geodatabase design and map production workflows. The research question for this project is: how can ESRI’s cartographic representation technology enhance map production of multiple map products from a single data source?

Cartographic representations provide users with the ability to maintain cartographic data as an integral part of their GIS data holdings rather than as a separate set of file-based objects that must be explicitly managed by the user. This functionality allows multiple users to access a single data source from which they can create a variety of different map products without remaining reliant upon file-based symbol storage formats, thus reducing data management tasks.

The client provided a personal geodatabase containing data for mapping the Crater Lake National Park, Oregon and its surrounding area, and a set of four test maps, three of which were available only in digital image formats. The project involved converting these data sources which are stored in geodatabase feature classes, raster data formats, ESRI style files, and fonts, into a single geodatabase that stored all of the vector feature symbology and supporting raster files for the project maps. The key point was that the single geodatabase was to store the symbology for multiple maps.

A number of cartographic representation techniques were evaluated, and most produced final results which either reproduced or enhanced the original map symbols. The project achieved two main goals: a) a successful prototype methodology for implementing cartographic representations into an existing cartographic geodatabase, b) all of the feature symbology to support four different maps was stored within a single geodatabase.

1.1. Client

The client for this project is Charlie Frye, Senior Cartographic Researcher at ESRI. As a software developer and vendor, ESRI is concerned with providing clients with comprehensive support resources for using ArcGIS desktop software. The development of software requires a theoretical underpinning which must be at least loosely tied to practical, applied GIS. The client identified the need for the development of a prototype methodology that would assist GIS cartographic specialists and database designers in understanding the requirements for implementing cartographic representations in order to

store multiple map symbologies within a single geodatabase. The project prototype was to be based on a robust sample data source for which an existing set of hard copy maps already existed.

ESRI's current product documentation requires augmentation in the area of cartographic data modeling, specifically as it pertains to storing symbologies for multiple maps within a single geodatabase. The client, who reports directly to the most senior level of ESRI product development, may elect to use the results of this project to support the future development of software tools, product documentation, and geodatabase data models.

1.2. Cartographic Representations Defined

This section contains an overview of how cartographic representations are structured within the ESRI geodatabase data model. The ArcGIS 9.2 Desktop Help (ESRI, 2007) documentation describes how the tools contained within the desktop applications are used. Therefore, this information will not be discussed herein. The information in this section is solely intended to provide sufficient background for this report, and it is not presented as a comprehensive review of the technology.

Cartographic representations are stored as a property of a feature class and can be created and managed through ArcCatalog, ArcMap, and with a number of representation-specific geoprocessing tools. The technology was developed to allow users to automate symbol creation and management tasks, with the intention of embedding a level of feature symbol intelligence, with the result being a "smarter map" (Punt *et al*, 2006). Both ArcCatalog and ArcMap contain user interfaces through which representation feature symbol properties can be defined.

Cartographic representations are stored within a series of attribute fields that relate to a number of geodatabase tables that normalize the storage of the symbology. A feature class representation is comprised of two attribute fields which are added to the existing feature class table when the representation is created. By default, these two fields are named "RuleID", which stores an integer data type, and "Override", which stores a BLOB (Binary Large Object) data type. The default names can be changed by the user; this is particularly necessary if a feature class stores more than one representation. The two fields are used to store feature symbol information for each row within the feature class.

The feature class representation can store any number of different symbol definitions. These are called representation rules. Representation rules are comprised of a set of feature symbol layers (i.e. "marker", "stroke", "fill"), which can be used in any number or combination to define feature symbology. The user defines how the rule is defined by setting display parameters for each symbol layer. The creation, definition, and editing of

representation rules and their properties are discussed in detail in Section 4. Upon creation, each representation rule is assigned an integer value, unique for the specific feature class in which it is stored. This value is stored in the RuleID field for an individual feature. The RuleID field integer value is related to a geodatabase system table, typically hidden from the user, named GDB_ExtensionDatasets, which stores the full symbol definitions within a BLOB data type field. This reflects a normalized database design schema for storing multiple symbol definitions within a single geodatabase: a single representation rule's symbol definition is stored once in the GDB_ExtensionDatasets table, but is drawn many times for the requisite feature class. This single system table thus stores the cartographic representation feature symbol information for the entire geodatabase, and removes the need to store multiple symbols within a feature class.

In contrast to the RuleID field, the purpose of the feature class Override field is to store any changes made to an individual feature's representation rule within the feature class attribute table. The BLOB data type stored in this field contains the entire symbol definition for the affected feature itself, rather than referring to the GDB_ExtensionDatasets table. Examples of the types of changes stored in the Override field include edits made to the location of the feature representation symbol for enhanced cartographic clarity or altering a feature representation's symbol for color, line width, or the like. Figure 1-1 shows a schematic view of the relationships between these tables and fields:

Figure 1-1 – Schematic illustration of cartographic representation geodatabase table relationships.

In order to draw this new symbology, ArcMap contains new display functions for rendering feature representations on the map. When a feature class containing representations is first added to a map, ArcMap draws the features with the standard default simple symbol (i.e., single line, round point marker, or polygon fill symbol) and a single random color. The user must then access the layer's properties to change the symbology and select from the list of feature class representations available for that layer.

Subsequently, should the map document or layer file be saved, the software will automatically recognize and draw the representation symbology the next time the file is opened.

Representation symbol properties are not limited to being stored only in the two feature class fields described above. The default fields are designed to provide support for full feature symbol definitions, but the user may elect to apply existing attribute values to define representation properties, e.g., line width, color, rotation angle, marker size, etc. In this manner, feature symbol properties are exposed to the user within the feature class table, rather than being hidden within the RuleID or Override fields; thus they can be edited directly using standard ArcMap editing tools. However, cartographic representation symbol properties require that attribute fields store data with specific data types (i.e., integer, floating point, double precision) for the available symbol parameters. For example, a marker rotation angle field must be stored as a “float” or “double” data type. The complete list of valid data types required for use with cartographic representations is attached in Appendix A.

2. Project Background

Cartographic representations were developed by ESRI for the purpose of improving high-quality cartographic symbology. This section describes three major reasons which led to the development of cartographic representations. The major goal for the development of cartographic representations was to store feature symbology within the geodatabase.

Prior to the release of ArcGIS 9.2, map production with ArcGIS desktop software stored map feature symbol definitions in several different files. Feature symbol definitions for a specific map could be stored in a map document (.mxd), a layer files (.lyr), or within ESRI style files (.style). Some symbols were also dependent on the characters in a font, which are files that are stored and managed by the computer's operating system. These files were used individually or in combination by organizations or individual cartographic specialists. A by-product of symbol definitions being stored in these files required that symbol standards be defined, stored, and maintained independently of the data used for maps. This variety of file-based symbol storage locations often resulted in maps not retaining consistent feature symbology across networks or from one computer to another. For example, when original data sources may have been moved or renamed, a broken link occurs between map documents or layer files and their source data. Further, font files not copied to the "destination" computer would result in symbols on the map and in styles being drawn incorrectly or not at all. Finally, the simple demands of ensuring that these various sources for symbol definitions were well managed was often time-consuming for the cartographic specialist.

An additional problem was that cartographers often needed to use multiple software applications for final map finishing tasks, e.g., creating a special symbol for a single map feature. A common map workflow would entail working with ArcMap to symbolize features in a basic map layout. That layout would then be exported to a graphics application such as Adobe Illustrator or Corel Draw, to add graphic effects that are not available in ArcMap to finish the map. This resulted in workflows that required additional time and broader skillsets, and converting the GIS data into a graphics format, which could not accommodate the attribute data. This reflects the historical development of GIS software as a spatial analysis tool, rather than as a cartographic tool. The development of cartographic representations is an incremental step of trying to develop more robust support for high quality mapping in the GIS (Buckley & Hardy, 2007).

Finally, the issue of creating cartographic data within a GIS has typically required that separate data sources be created solely for map-based priorities. Recent work by the ESRI Cartographic Research Group suggests that this rather limited approach to geodatabase design is more a reflection of users' lack of understanding of ArcGIS capabilities rather than a limitation of the software (Frye, 2006). Through most of the history of GIS-based cartography, the original source data has driven the map, rather than the map defining the geodatabase design – which is the optimal solution for GIS cartography (Buckley, Frye, Battenfield & Hultgren, 2005). Spatial data is usually

collected for use in GIS analysis or data inventory, which reflects the reality that many datasets are collected for a single purpose, which is usually not cartography (Buckley & Frye, 2007). As a consequence, the creation of high-quality cartographic products has required the development of separate geodatabases containing new data sources, or the enhancement of existing data to support map feature symbology that meets the appropriate standards for cartographic clarity (Arctur & Zeiler, 2004). Often, this meant editing the original feature geometry, thereby compromising the topological and spatial integrity of the data.

One use of cartographic representations is intended, in part, to provide cartographers who use ArcGIS the ability to alter feature symbology without also having to alter the underlying feature geometry while still using the same geodatabase source for data. This functionality eliminates the need for cartographers to create separate feature classes for solely cartographic purposes. This has the benefit of reducing data storage needs by maintaining a dynamic relationship between map feature symbology and underlying feature geometry in a geodatabase.

The ESRI literature describes the development of cartographic representations as a significant first step toward the solution to these problems in map production workflows and cartographic data management (Hardy, Eicher, Briat, & Kressmann, 2005; Hardy & Kressmann, 2005).

2.1. Problem Statement

The main purpose of the project is to discover how feature symbols for multiple map products can be stored within a single geodatabase. In the course of the research, new techniques and functionality available with cartographic representations will be evaluated against the standard ArcMap symbol management tools.

The ArcGIS Desktop Help system contains comprehensive information describing the general purpose and use of individual buttons, toolbars, and geoprocessing tools specific to cartographic representations. As this technology is new, there is currently a lack of published information regarding how these tools might be used in a “real world” applied context – that is, “best practices”. The client would like to use the results of this project to augment this use case information.

The project will use an existing map-based work flow to evaluate how cartographic representations might be implemented in a production environment. This project attempts to achieve three goals:

1. develop a prototype map production methodology for applying cartographic representations to an existing geodatabase from which four test maps can be created;

2. evaluate how cartographic representations may enhance map feature symbology;
3. create a single geodatabase from which multiple maps can be produced.

2.2. Scope of Work

The ESRI Cartographic Research Group maintains a large data set for Crater Lake National Park, Oregon and its surrounding area. This was selected as the test material for the project. The source data supplied by the client, in late October 2006, was the project snapshot; new spatial data sources were digitized or converted as required. Ancillary information, such as fonts, ESRI styles, and original map images were supplied by the client, when available and as required.

The main purpose of the project was to evaluate cartographic representations, which are applied only to geodatabase vector data. However, it was necessary to make adjustments to raster and geodatabase annotation data sources to complete the final maps. Work on annotation alone could have become quite time-consuming, as each of the test maps contained very different typographic standards and map scales. It was instead decided early in the project to limit the work on the map annotation and to use only that which was supplied with the original geodatabase, and make only minimal changes to this default data for each map.

2.3. Project Deliverables

The goal of the project was to provide the client with a geodatabase that supports the production of four maps, as well as a document containing a description of potential best practices (i.e., this report). As a final project deliverable for mapping, the client received a file geodatabase, four map documents, and two sets of four hard copy maps: one set of original maps, and one set of maps created with ArcGIS using cartographic representations.

3. Project Methodology

The methodology described below suggests an initial set of “best practices” for implementing cartographic representations to support multiple maps in a single geodatabase. These were initially outlined through informal discussions between the client, the academic advisor, and the researcher, and revised as the work progressed. The summary is as follows, with each step described in detail below:

1. Identify test maps
2. Construct maps with ArcGIS
3. Compile test geodatabase
4. Visually evaluate maps for use with cartographic representations
5. Implement cartographic representations
6. Evaluate results
7. Consolidate project geodatabase
8. Produce final maps

3.1. Identify test maps

The ESRI Cartographic Research Group has been using a robust data set for the Crater Lake National Park, located in southwest Oregon, to investigate new cartographic techniques. At the client’s request, this dataset was used as the subject area for this project. In addition to collecting the GIS data, the ESRI Cartographic Research Group had assembled a series of hard and soft copy reference maps of the Crater Lake park area, which represented a temporal and stylistic range of cartographic techniques. Four of these maps were identified as test maps for the project. Reduced-size images of these maps are shown in Figures 3-1 through 3-4, with the map naming conventions adopted for this project.

3.1.1. National Park Service (NPS) Map

The original source for this test map was a published brochure available from the U. S. National Park Service (NPS) whose use is intended for a general audience of park visitors who are primarily traveling by car (USNPS, 2007). This map was re-created by Dr. Aileen Buckley, using ArcGIS to investigate the mapping capabilities of the software. This map was the most “ArcGIS-ready” as it was stored as a map document (.mxd) and contained well-defined feature symbology. This map was used as the initial source for the project geodatabase, as it was the most recently published and data-complete.

3.1.2. Digital Raster Graphic (DRG) Map

The original source for this test map was a series of hard copy United States Geological Survey (USGS) topographic quadrangle map sheets for the area surrounding Crater Lake National Park. Scanned images of these maps were combined as a single mosaic image and stored as a MrSID (.sid) georeferenced raster image file.

Figure 3-2 - DRG map.

3.1.3. TOPO Map

The source for this test map was a modified copy of an NPS map of uncertain date, provided as a scanned image and stored in Joint Photographic Experts Group (.jpg) format (USGS 2007a).

Figure 3-3 - TOPO map.

3.1.4. TOPO_OLD Map

The source for this map was also an excerpt from a USGS topographic map, but from an earlier decade than the TOPO map. This map was provided as an image file stored in the .jpg format.

Figure 3-4 - TOPO_OLD map.

The image resolution for each of the three file-based map sources (DRG, TOPO, TOPO_OLD) was recognized as a potential challenge for the accurate capture of feature symbology for mapping with cartographic representations.

These maps represent a variety of USGS and NPS mapping styles for the Crater Lake National Park and its surrounding area. Each map was unique in some way, and differed in typography, symbology, style, and production date. It was the differences in feature symbology between the maps that this project attempted to capture and integrate into a single geodatabase through the application of cartographic representations.

3.2. Construct maps with ArcGIS

As discussed in Section 2.3, one project goal was to re-create each map with ArcGIS Desktop software and produce a single map document for each test map, all drawn from the same geodatabase with the fewest number of feature classes possible. The design approach for both the geodatabase schema and the map symbology was to replicate the original test maps. This approach follows the one advocated by the client, whereby the final map product is used as the template to design the underlying geodatabase and map production workflow (Frye & Eicher, 2003). As this project is using newer functionality, there is an adaptive, iterative component to the development of both the methodology and the final geodatabase design. This approach differs from that of the more common “traditional” geodatabase design, which is typically concerned with data accuracy, precision, resolution, completeness, fitness-for-use, and other priorities which support GIS analysis or spatial data warehousing (Arctur & Zeiler, 2004). Often mapping is not a priority, and in some cases it may not be considered at all (Frye, 2006).

It should be noted here that differences in data capture techniques can produce different results for both GIS mapping and analysis. For example, much of the data collected for the project geodatabase was digitized directly from a cartographic product. As such, the resolution and accuracy of the road and stream lines, for example, is the product of data previously generalized to support a hard copy map at a specific scale.

The first task was to create a new ArcMap map document for each test map. The main purpose of this task was to define the broad parameters from which the geodatabase design and further map refinements would derive. These parameters included map output, page size, map scale, and the initial set of vector and raster data layers.

3.3. Compile test geodatabase

After the completion of the initial map document for each map, the next step was to create a new geodatabase which contained the minimum number of data sources required to support all of the maps. Original vector format source data supplied by the client was

stored in Personal Geodatabase for Access (.mdb), with supporting image and raster information stored in a number of file directories and data formats. Ancillary information was stored within a variety of map documents, style files, and fonts. A major goal of the project was to create a single, integrated data source which relied on no external file-based data sources.

This part of the project consumed a significant amount of time. The source geodatabase contained a number of redundant and unused feature classes, raster layers, and attribute fields, most of which were the result of data conversion. The ArcGIS 9.2 software release introduced a new geodatabase storage format, the File Geodatabase (.gdb). One of the main enhancements available with the file geodatabase is its ability to store raster information as a database table, rather than as a referenced file-based image, as is the case with the Personal Geodatabase for Access format (.mdb). It was determined that the file geodatabase format would be used for this project in order to store raster information and to take advantage of the enhanced data compression available.

3.4. Visually evaluate maps for use with cartographic representations

A pair of maps was printed from the original data sources and the new map document for each test map. Each map pair was visually evaluated to determine where cartographic representations could best be used. Effective evaluation of the maps required that the investigator possessed a comprehensive knowledge of the basic functions of cartographic representations. A thorough understanding of how map features depicted on the hard copy maps could best be symbolized or enhanced by cartographic representations was a mandatory precursor to the success of the project. This prior knowledge of the software capability reflects a more general requirement for effective use of GIS for cartography (Buckley & Frye, 2007; Buttenfield & Frye, 2006; Brewer & Frye, 2005). The results of this step were used to compile the list of use cases for the implementation of cartographic representations.

3.5. Implement cartographic representations

The results of the visual evaluation step for each of the maps provided the substance for the majority of the project work. The implementation of the cartographic representations is the subject of further discussion and description in Section 4.

3.6. Evaluate results

This step resulted in a series of map iterations. Once the initial implementation of cartographic representations was complete, weekly meetings between the researcher, the client, and the academic advisor helped to refine the maps. The results of this work are contained within the project data CD as the final project map documents.

3.7. Consolidate project geodatabase

Upon completion of the final map documents, as well as the creation of cartographic representations within the same geodatabase, a number of feature classes were no longer required or were able to be consolidated. The goal was to create a geodatabase with the fewest number of feature classes and raster files.

3.8. Produce final maps

The maps drawn using cartographic representations were visually compared by the client and the academic advisor. At this step, final decisions were made with regard to map scale and page size.

4. Implementing Cartographic Representations

This section describes the specific instances where visual examination of the hard copy source maps identified the need for the use of cartographic representations to resolve a feature symbol problem, enhance existing font-based or other symbology, or create an effect not possible or difficult to create with standard ArcMap symbol management tools. For each test map, a discussion of how each of these issues was resolved (or not) through the use of cartographic representations is presented, along with an initial evaluation of their efficacy in providing an enhanced cartographic outcome. A summary discussion of the use of cartographic representations symbol tools is presented in Section 6, with references to the use cases discussed below.

4.1. NPS map

The NPS map was created as a general reference map for visitors to the park. The original map brochure was published in 2002 by the National Park Service at a map scale of 1:63,360 (USNPS, 2002). At that scale, the map is meant to be used for driving throughout the park, but it is not intended for hiking or back-country camping. The project commenced with this map for three main reasons: a) the map symbology was already well defined and stored in a number of ArcGIS-related file formats, b) the source geodatabase was complete and had been compiled primarily from NPS and USGS data sources as well as select feature classes digitized by the client, c) the client had already created the map in ArcMap and supplied a map document from which to start. In addition, the map supplied by the client was itself a re-creation of an original brochure-sized NPS map (USNPS, 2007). The map in the brochure was created without the use of GIS technology and was the subject of previous research by the ESRI Cartographic Research Team as a source document from which to develop new cartographic techniques with GIS. The original map brochure provided an additional source against which to evaluate the final map output from this project.

As described in Section 3.4 of the Project Methodology, visual inspection of the map revealed that the NPS map had the largest number of unique use cases for which the use of cartographic representations showed the most promise. An overview map of these use cases is shown in Figure 4-1.

1. Gradient line symbol for park boundary
2. Multiple marker symbols for park facilities
3. Single marker symbols for park facilities
4. Cased line road symbol
5. Bog polygon marker fill
6. Dash line control points
7. Marker rotation

Figure 4-1 – Index map of NPS use cases for cartographic representations.

4.1.1. Gradient line symbol for park boundary

The park boundary symbol included in the map document consisted of a series of twenty buffered polygons each 5 meters wide, and symbolized with a green color ramp, as shown in Figure 4-2. As shown in the ArcMap Table of Contents, the data contained in this feature class (*park boundary buffer*) was created explicitly for cartographic purpose, and copied from the original USGS park boundary data (*park_bndy_USGS*), which was also stored in the source geodatabase.

Figure 4-2 - Original NPS Park Boundary buffered polygon symbol.

Two different cartographic representation techniques were evaluated to re-create the original effect: a series of offset stroke symbol layers, and a “Donut” geometric effect which created an outline polygon with a specific width to which a gradient fill was applied. These are described in greater detail below. Each of the two techniques was stored as a separate feature class representation within the *park_bndy_USGS* feature class.

The series of offset stroke symbols were stored as a representation rule in the “NPSbndy” feature class representation. The number of symbol layers was reduced from the original 20 to 10, primarily as a symbol layer management issue. Twenty representation symbol layers became difficult to manage within the layer properties interface, and 10 layers

were able to achieve a similar effect at the final map scale. Initially, each of the 10 stroke lines was rendered with a 0.5 point line width. Each stroke layer was offset from the one “inside” it in draw order by a margin of 0.5 points, such that the stroke lines abutted one another. The original green color ramp colors then were applied to the stroke symbol layers (Figure 4-3).

Figure 4-3 - Park boundary layer representation properties for stroke symbol layers.

The resulting effect closely matched the one on the client’s map. One benefit of converting the symbology to a cartographic representation was that the original separate feature class (*park_boundary_buffer*) was no longer required in the geodatabase. This small reduction in database storage has implications for a large data warehouse, where similar removal of redundant data could provide significant reductions in disk storage needs. Furthermore, having a single data source for both geometry and feature symbology allows dynamic changes to feature geometry to automatically be populated to any maps which use that data source. In this specific example, should there be any future alterations of the park boundary line features, the symbology will remain dynamically linked and current. This would not be the case with the original buffered polygons; the buffers would have to be re-created from an updated park boundary feature class and then re-symbolized.

The benefits noted above were somewhat offset by a notable difference in the speed with which the feature representation symbols were rendered: they drew much more slowly than the original polygon buffers. Moreover, when the first check plot for this map was printed, there was a noticeable gap of approximately 2 points line width between the individual stroke symbol layers which did not preserve the smooth gradient fill

appearance. This was likely due to the rasterization applied to vector lines by the ArcMap printing output engine, which attempts to differentiate between individual line features. This problem was alleviated by increasing the line width of each stroke symbol layer to 0.75 points, in effect creating an overprint. The final result is shown in Figure 4-4.

Figure 4-4 - NPS park boundary with cartographic representations using offset stroke symbols.

Repeated experiments with different settings for the fill symbol layer failed to produce a satisfactory output, as shown in Figure 4-5. In addition, the rendering speed for this technique proved to be even slower than for the stroke symbols layer representation.

Figure 4-5 – Cartographic Representation stroke symbol with gradient fill.

4.1.2. Multiple marker symbols for park facilities

One of the more common problems with sharing maps between computers and different software applications is that font-based symbols and text may not translate correctly. This is typically because any unusual or unique fonts may not be installed on the recipient computer. This was the case with the original data supplied by the client for this project. The NPS map used a number of special font-based symbols, which were not embedded within the map, or installed on the test computer. The result is shown in Figure 4-6.

Figure 4-6 - NPS map font and symbol errors due to the lack of required fonts on the computer.

Note that the errors illustrated above were of two types: feature symbols (e.g., springs shown with a blue “1/2”, falls shown with an “=” sign, and campsites shown with the ç and £ symbols), and those used for descriptive mimetic symbols used in the “Rim Village” and “Munson Valley” labels. This problem was easily corrected by installing the appropriate fonts, but it provided a good example of how cartographic representations could be used to permanently alleviate this common problem. There are two main benefits to converting font-based markers to representation marker symbols: maps and feature symbols are no longer strictly reliant upon a specific font and, once converted, representation markers can easily be edited.

The NPS map contained feature-based font symbols of three types: feature symbols (e.g., hydrologic features), mimetic symbols (e.g., campground), and “stacked” labels (e.g., Munson Valley). Much of the work in managing these symbols focused on the *cultural_pts* feature class, which contained the information for park facilities, such as campgrounds, picnic areas and overlooks, the discussion below reflects this focus. The markers were added to the map as labels which were defined with different label expressions for each specific point feature. Additional discussion regarding the symbolization of hydrologic features is included in Section 4.1.7.

4.1.3. Single marker symbols for park facilities

The initial approach to reproducing the NPS map font-based marker symbols was to convert each unique symbol into a separate representation rule within the feature class representation (“NPSrep”). Each of these mimetic symbol types (i.e., camping, picnic,

etc.) consisted of a single black-colored glyph. When originally drawn, this allowed the map background to show through the “empty” area of the glyph (Figure 4-7, left).

Figure 4-7 - Marker without background (left) and with background (right).

The map standard required that each glyph be drawn against a white background, such that the “empty” space within the glyph was white and did not allow the underlying map features to be displayed (Figure 4-7, right). The cartographic representation management tools include a marker symbol editor; this functionality was used to add the white background to the marker symbol.

The marker symbol editor was also used to create a new symbol required for this map. A glyph depicting a backcountry campground was available from an existing font, but required the removal of the border and the retention of the white background. Figure 4-8 shows the original marker symbol, and Figure 4-9 shows the appearance of the edited representation marker as drawn on the map.

Figure 4-8 - Representation marker editor, showing original font-based glyph.

Figure 4-9 – Final representation marker as drawn on the map.

The symbol editing task was simple to perform with the representation marker editor. It should be noted here that learning to use the Marker Editor required a relatively short time (i.e., approximately two hours). This learning curve was made easier by the fact that the software interface had intentionally been designed to operate in a manner similar to other well-used graphics applications (i.e., Adobe Illustrator). The ability to edit marker features within ArcMap 9.2 provides a more streamlined workflow option than the alternatives, which include the use of an external font editor application.

Standard ArcMap symbol editing tools are not able to exactly reproduce this effect, as they do not provide the user with the ability to edit the individual vertices which comprise the marker symbol. A similar effect could be produced with standard ArcMap symbol tools by creating a multi-layer symbol combined with masking effects; however, this would require a significantly more complex and time-consuming workflow.

4.1.3.1. Markers with stacked labels for Mazama Village, Rim Village, and Park Headquarters (Munson Valley)

Three point symbols required the use of marker symbols within a stacked label, which is a label that is shown on more than one line of text. The first technique tested to resolve this issue with cartographic representations was to convert each point feature, specifically, the location of Mazama Village, Rim Village and Munson Valley, into a Free Representation. A Free Representation provides the user with the option to completely detach an individual feature representation from the representation rule base. A Free Representation symbol essentially becomes its own unique rule. This option was not implemented for two reasons: the feature symbology was not so complex as to require the detachment of each feature from the feature class representation rule base, and a simple and more effective method was subsequently discovered, namely, a unique representation rule for each of the three points. The “stacked” symbols were constructed for each location as a set of individual symbols with a cumulative x-axis offset value applied to each individual symbol within the string. The point features for each location were then assigned their respective representation rule in a subsequent edit session. The final appearance of these representation markers is shown in Figure 4-10.

Figure 4-10 - Unique representation rules for stacked markers at Rim Village and Munson Valley.

An advantage of converting the initial font-based symbols to representation markers was that the symbols could be moved independent of their associated features, similar to annotation features, which provided enhanced control over marker placement. Using the representation symbol tools provides a more user-friendly graphical interface through which to manage these symbols than with standard label placement tools. For example, the original label expression used to define the annotation feature was:

```
Munson Valley<FNT name='ESRI NPS 1' size='11'>
<CHR spacing='25'><LIN leading='-12.75'><CLR red='255'
green='255' blue='255'>%%></CLR>
<CLR red='0' green='0' blue='0'>VñkÇ-e</CLR></LIN></CHR></FNT>
```

Note the inclusion of the ‘%’ and ‘VñkÇ-e’ characters embedded within the expression; Figure 4-6 shows the result of using these in the label string when the necessary font (i.e., “ESRI NPS 1”) is not installed. Contrast the expression above with the visual interface provided by the representation symbology tools shown in Figure 4-11.

Figure 4-11 - Representation marker layers used to symbolize the three park locales with multiple amenities shown with the marker symbols.

The representation symbol management tools were easier to use than tools for geodatabase annotation for this specific use case, and provided greater control over the placement of each element in the final stages of cartographic refinement for the map. The marker editor interface provides direct access to the computers' font library character maps, which enables the manual selection and management of individual glyphs. Map publishing is enhanced because the symbols are now stored as a geodatabase property rather than as a file on disk.

4.1.4. Dashed line intersections for trail symbols

One advantage of using cartographic representations for dashed line feature symbology is that the user can control how dashed line patterns start, stop, and intersect. For the trails feature class (*trail_lines_dissolved*), converting the trail lines from standard symbols to cartographic representations provided enhanced control over how the beginnings and endings for the dashed lines would appear. The default setting for lines to end with a half pattern was retained. No significant change to this setting was required to conform to the map standard, as shown in Figures 4-12 and 4-13.

Figure 4-12 - Standard trail line symbols: line ends have dashes of varying length.

Figure 4-13 - Trail feature representations: black lines end with a half dash.

4.1.5. Cased line symbol for roads

The ability to control cased line symbols for linear features is available with standard ArcMap symbol tools. The problems that can occur with this type of symbol are easier to resolve with the representation symbol management tools. This is largely due to having the individual parameter controls exposed within a single interface rather than through a series of dialog boxes and tabs, as with the standard ArcMap symbol management tools.

Two problems were encountered when the cased road symbols were converted to cartographic representations: road corners did not “wrap” correctly where the corners occurred at a line’s end, and one location exposed a gap in the underlying feature geometry. Figures 4-14 and 4-15 illustrate how the change in the line caps type from “butt” to “round” resolved the problem. The second problem was also solved with the symbology change, as shown in Figures 4-16 and 4-17.

Figure 4-14 - Cased line caps with the default setting to use Butt line caps.

Figure 4-15 - Cased line caps with Round line caps.

Figure 4-16 – Gap in data exposed with Butt line cap.

Figure 4-17 – Gap in data closed with Round line cap.

4.1.6. Polygon fill pattern for bogs

This use case presented one of the more difficult challenges for mapping with cartographic representations. The single marker symbol used for the layer did not include the second element of the fill pattern as drawn on the original NPS brochure map, i.e., short dashes randomly distributed around the “marsh” marker symbol, as shown in Figure 18.

Figure 4-18 – Source map excerpt showing original bog marker symbol.

The polygon marker fill pattern supplied by the client was acceptable, but it is not an optimal solution. Figure 4-19 shows the outline of the polygon features to which this representation rule was applied. The outline does not appear on the final map but it is included here to illustrate the issue with using the random marker placement within a “donut” polygon: the markers are not drawn inside the non-bog polygon.

Figure 4-19 - Bog feature drawn with a simple marker fill pattern and random marker placement.

The first approach to include both the marsh marker and the randomly placed dashes was to add a second representation stroke symbol layer with random placements and experiment with various offset spacings. Despite considerable experimentation, it was not possible to find a combination of settings to apply to both the marker symbol layer and the stroke symbol layer such that the symbols did not overlap.

The second approach was to edit the marker symbol. As described earlier in Section 4.1.3, the Representation Marker Editor provides enhanced controls to customize feature symbology. In this case, there were no font-based markers from which to convert, so the result was a symbol unique to this map, as shown in Figure 4-20. While this solution is acceptable, it is not the optimal desired outcome, as the polygon is not filled with the randomly placed dashes. Another option which was considered for this specific map symbol was to convert the feature representation to a Free Representation. This would have detached the single selected feature from the feature class representation rule and allowed the user to edit the individual symbol elements independently. Each “marsh” marker symbol and dash line would thus be available for manual editing through the Representation toolbar. However, the cost of this fine-grained editing functionality is that the feature is permanently detached from the feature class representation rules base, and the Free Representation symbol is not available for use with other maps. The main goal of the project was to create feature representation rules that could be stored once and used for multiple maps. Thus, the final appearance as shown in Figure 4-20 was deemed acceptable for the purposes of this use case.

Figure 4-20 - Final bog representation symbol.

4.1.7. Control points for trail dashed line symbol

The map contained a number of locations where the combination of the trails dashed-line pattern and convoluted-line geometry resulted in cluttered and/or ambiguous symbology, primarily where switchbacks were located. This use case illustrates one of the key benefits of cartographic representations: the ability to edit feature symbology for cartographic refinement without having to also edit the underlying feature geometry. Up to this point in the discussion of this map, none of the cartographic refinements required any explicit editing of a feature representation's location.

In the two use cases illustrated here, the symbol ambiguity was compounded by the fact that the dashes are, by default, placed according to their relationship to the underlying geometry, in this case as line segments defined by the originally digitized vertices. Figure 4-21 shows how the trail on Mount Scott originally appeared. This example, as well as the trail on Wizard Island and the boat launch trail on the north shore of Crater Lake (Figure 4-22), required the use of tools located on the Representation toolbar within an ArcMap edit session. To correct this display using representations, either the existing feature vertices were converted to representation control points, or control points were manually added where required.

Figure 4-21 - Mount Scott trail symbol prior to editing feature representation control points.

Figure 4-22 - Boat launch trail symbol prior to editing feature representation control points.

Figure 4-23 - Mount Scott trail with feature representation control points added.

Figure 4-24 - Boat launch trail after control points were added. Note the full dashes at the switchbacks.

Feature representation control points allow the user to manually define where feature representation symbol elements start and end. This user-defined symbol property is illustrated in Figures 4-23 and 4-24, where the trail corners are drawn with a full dash, clearly showing the curve of the trail.

4.1.8. Rotation angle for hydrologic marker symbols

As with the marker symbols for the *cultural_pts* feature class, discussed in Section 4.1.3, the default symbols for the *hydro_pts* feature class were originally font-based. Conversion of the original symbols to representation marker symbol layers was straightforward. After being converted to feature representations, the symbols were initially placed with the default horizontal orientation (i.e., zero degrees rotation, without specifying an override field). The map symbol specification required that the spring tails be oriented toward the direction of flow on their respective streams or slopes and the fall symbols be placed perpendicular to the stream line on which it was located. The *hydro_pts* feature class contains an attribute field (“Angle”) which stores the specific angle of rotation for each individual feature.

This use case exposed an omission in the ArcGIS Desktop help system which does not describe the way representations use rotation angles (ESRI 2007). Cartographic representations use graphic rotation for most rule parameters and geometric effects in order to conform to common graphic software standards. Graphic rotation is a new rotation method applied within ArcGIS 9.2 software, whereas previous versions supported only geographic and arithmetic rotation. Graphic rotation starts with zero degrees at the top of the map, as does geographic rotation. However, graphic rotation rotates values counter-clockwise, whereas the geographic rotation is clockwise. Arithmetic rotation starts with zero degrees at the right and rotates values counter-clockwise. Figure 4-25 illustrates the differences between these three rotation methods. It should be acknowledged that the terms “geographic”, “arithmetic”, and “graphic” rotation conform to ESRI’s definition and use of these terms within their documentation (ESRI 2007).

Figure 4-25 - Geographic, arithmetic, and graphic rotation values.

In order to draw the falls and springs symbols with their correct orientation, two data manipulations were required. First, the original angle attribute values were converted from an integer data type to a float data type because the representation marker parameter requires that input data be stored as either a float or double data type (see Appendix A). Therefore, the original attribute values were copied into a new attribute field. However, the symbols were still drawn incorrectly, as shown in Figure 4-26, because the original attributes were stored as geographic angles but drawn by the representation renderer as graphic rotation. This required a second data manipulation to re-calculate the rotation angles. The solution was to subtract the original rotation values from 360; this was performed by using the ArcMap Field Calculator in an edit session.

Figure 4-26 - Spring markers with incorrect rotation due to using geometric angle values.

Figure 4-27 - Spring markers with correct rotation using new graphic rotation values.

The *hydro_pts* marker symbols are an example of how cartographic representations provide the user with the ability to dynamically link a feature representation property with an attribute stored in the feature class table, in this case the rotation angle. As

described above, the original point geometry data contained an angle field which required some data processing in order to be used with the cartographic representation renderer. Shown below is the method by which this value was applied. Figure 4-28 shows the fields that may potentially be used for representation marker symbol layer properties (marker, size, angle, x offset, and y offset). Note that, by default, all these fields are mapped to “override field”. This is the BLOB field (“Override”) created as part of the feature class representation schema specifically to store parameters such as the ones discussed here. However, the user may opt to relate a field to an existing feature attribute.

Figure 4-28- Default settings for “Falls” marker symbol layer.

As shown in Figure 4-29, the representation marker “Angle” field was changed to the “angle_fl” field. Through the use of this function, changes made to feature attributes at the geodatabase level are automatically reflected in the feature representation symbol.

Figure 4-29 - The Angle parameter is now set to an attribute field so that the value is automatically applied to the representation marker.

4.1.8.1. Remove selected hydrologic features from display.

A final step was required to improve graphic clarity by reducing the number of features drawn. As shown in Figure 4-27, there were a number of locations where the many spring feature symbols overlapped, resulting in a cluttered appearance. ArcMap Representations tools were used within an edit session to change the representation rule value for selected features from “Spring” to “no rule applied”. The result was that the selected features were not drawn. Standard ArcMap tools provide a similar solution: create an attribute field which stores a value (i.e., on/off) which is then used in a definition query to define which selected features will be displayed. The benefit of using this technique through the cartographic representation properties was simply that no new attribute fields were required, thereby reducing the size of the feature attribute table, and retaining the initial geodatabase schema. The final appearance of the springs in this section of the map is shown in Figure 4-30.

Figure 4-30- Spring marker with selected cartographic representations removed.

4.1.9. Background raster layers

Each of the four test maps contained various map layers drawn from raster sources. In the case of the NPS map, the ESRI Cartographic Research Group had created a variety of raster layers to depict topographic and land cover information (Dr. A. Buckley, personal communication, October 2006). While these data sources did not present a germane use case for cartographic representations, they are noted here as they comprise much of the map's graphic content.

Two raster layers used in this map, *DEM* and *Bump_map*, were drawn with their original display properties as supplied by the client. The *DEM* raster layer contains the data used for the hypsometric tinting, while the *Bump_map* raster layer contains the 'textured' information. The original map document also contained three additional raster layers which defined the Crater Lake bathymetry. These were combined, or "flattened", by exporting a single image file (JPEG) from ArcMap comprised of only those three layers, and importing it into the project geodatabase as a single georeferenced raster layer. This process reduced the size of the geodatabase by combining three raster files which stored elevation and hillshade values and converting these original raster cell values into color values which were added to the map with no requirement for additional symbol manipulation.

4.1.10. NPS map summary

The NPS map provided the most extensive set of use cases for cartographic representations, consequently it took longer to complete than the other maps. Most of the feature classes used to make this map were used to make the other maps, as there now existed a cartographic representation rules base for new map symbology that could be

augmented or modified. As a result of the more intensive work on the NPS map, each successive map in the project series required less time to implement cartographic representations.

4.2. DRG map

The USGS Digital Raster Graphic (DRG) map mosaic was the second map to be evaluated. The cartographic representation use cases evaluated with this map were:

1. Building markers aligned to roads
2. Masking layer for road symbols
3. New PLSS line symbol

A benefit of using this map as a target was that the feature symbology conformed to published USGS symbol standards (USGS 2007b). Consequently, feature symbol parameters such as line width, color, and dash pattern were simple to determine. As was the case with all three of the image-based map sources (DRG.sid, TOPO.jpg, and TOPO_OLD.jpg), direct sampling of the map for symbology was a problem due to limitations of the image format pixel size and color range. That is, the digital images of these maps were dithered, resulting in combinations of variously colored pixels to give the impression of a single color or symbol over a larger area. For example, a simple line depicting a stream feature was, upon close examination of the scanned image, comprised of a number of pixels with different blue colors. Much of this type of manual symbol sampling was alleviated because the client supplied a number of the USGS symbols in an ESRI style file.

One goal of working with the three remaining USGS-based test maps was to re-use as many of the same data sources from the NPS map as possible, so as to reduce the size of the final geodatabase and gain efficiencies in feature class management. Thus, the USGS symbols were combined with the existing NPS feature geometry.

4.2.1. Building markers aligned to roads

This example used the *building_pts* feature class which contained point features for which simple building markers were drawn on the map. A new feature class representation was created with a set of representation rules which each contained a single marker layer for each of three building classes. The marker symbols were initially drawn with a horizontal orientation, as shown in Figure 4-31.

Figure 4-31 - Non-rotated building representation marker symbols.

The map specification required that these marker symbols be oriented to align with the road line feature closest to them, as shown in Figure 4-32. The solution to this problem was easy to achieve by using the Align Marker to Stroke or Fill geoprocessing tool. This geoprocessing tool is new with the release of ArcGIS 9.2 (contained within the Cartographic Tools, Symbolization Refinement toolset), and engineered to process only feature representation input and output. The output result was that the symbols were automatically rotated, and the rotation values stored within the Override field. An added benefit was that the new rotation values could be modified directly from the geodatabase, if necessary.

Figure 4-32 - Building representation markers after using the Align Marker to Stroke or Fill geoprocessing tool.

This use case is a good example of a map production workflow enhancement available with cartographic representations. Workflows used to achieve this solution in earlier versions of ArcGIS were extremely time-consuming, as each symbol required a manually adjusted edit per feature. An ESRI software product called the Production Line Tool Set (PLTS) can also be used for this, but at a potentially prohibitive cost for smaller organizations.

4.2.2. Masking layer for road symbols

As illustrated in Figures 4-31 and 4-32, the road symbols for this map, both solid (paved roads) and dashed lines (unimproved or dirt roads), were drawn on the final map with a cased symbol separated by a transparent line. However, the original symbol was a solid line, as shown in Figure 4-33.

Figure 4-33 – Initial road line symbol.

Cartographic representations provide the user with the ability to create a dynamic relationship between feature representations and an individual feature mask, implemented through a geodatabase relationship class. The workflow defined in the ArcGIS documentation recommends that individual masks for selected feature representations be created within an edit session through the use of the Mask Tool on the ArcMap Representation Toolbar. The representation masking function was evaluated as a solution for use with the entire roads feature class representation. The representation masking function could have been implemented as designed for use on a feature-by-feature basis; however, it would have required that each individual road feature be masked through a time-consuming edit session in which each feature was manually traced.

The solution for this use case was achieved through the creation of a new road mask feature class created with the Feature Outline Masks tool, which did not require the use of cartographic representations. This functionality has been available with ArcMap since version 9.0. The Feature Outline Masks geoprocessing tool was used within ArcMap, with the *road_line_NPS_dissolved* feature layer used as the input. The geoprocessing tool created an outline mask based on the input feature representations' symbology rather than the simple line geometry, resulting from the input layer already having had the feature representations applied, as shown in Figure 4-34.

Figure 4-34 – Road mask layer created with Feature Outline Masks geoprocessing tool drawn in grey.

The final appearance of the roads, drawn with the masking layer selectively applied to only the roads feature class, is shown in Figure 4-35. This use case provided an example of where the combination of standard ArcMap methods and geoprocessing tools offered a more effective solution than cartographic representations.

Figure 4-35 – Layer mask applied with Data Frame Advanced Drawing Options.

4.2.3. New PLSS line symbol

The DRG map included Public Land Survey System (PLSS) lines and associated annotation which were not drawn on the NPS map. The data source for this information was available as a polygon feature class stored within the original geodatabase.

The first cartographic representation solution evaluated was to create a new feature class representation for the *OR_PLSS* polygon layer, and to add a representation rule consisting of a single stroke symbol layer with a dashed pattern. However, as both the coincident boundary lines of adjacent polygons were drawn, this produced overprinting of the dash pattern, which often resulted in a solid red line and inconsistent dash phasing, shown in Figure 4-36.

Figure 4-36 – PLSS polygon boundary overprint

To resolve this problem, the *OR_PLSS* polygon feature class was converted to line geometry and added to the map as *OR_PLSS_lines*. A new feature class representation was created and the line symbol was re-applied to the line features. The result was satisfactory because the line feature representations were consistently drawn with the correct dash pattern. However, this solution produced a new problem. As shown in Figure 4-37, the PLSS lines were initially drawn with their line ends meeting with a cross-, or T- shaped join.

Figure 4-37 - PLSS lines with incorrect line caps.

This symbology has a specific meaning when depicting PLSS information on USGS maps. Cross-like symbols are used to depict actual “found” or surveyed section corners, “T”s depict “found” closing corners, as shown in Figure 4-38.

LAND SURVEYS	
<i>Public land survey system</i>	
Range or Township line	—————
Location approximate	- - - - -
Location doubtful	- - - - -
Protracted	*****
Protracted (AK 1:63,360-scale)	—————
Range or Township labels	R1E T2N R34W T4S
Section line	—————
Location approximate	- - - - -
Location doubtful	- - - - -
Protracted	*****
Protracted (AK 1:63,360-scale)	—————
Section numbers	1 - 36 1 - 36
Found section corner	—+—
Found closing corner	—+—
Witness corner	—+— WC
Meander corner	—+— MC
Weak corner*	—+—
<i>Other land surveys</i>	
Range or Township line	*****
Section line	*****
Land grant, mining claim, donation land claim, or tract	—————
Land grant, homestead, mineral, or other special survey monument	■
Fence or field lines	—————

Figure 4-38 – USGS topographic symbols for PLSS features (USGS, 2007b).

In this case, these symbols were incorrect, so the representation rule required further enhancement. The solution was to apply the “Cut Curve” geometric effect to the representation symbol layer. The Cut Curve geometric effect creates a representation line that is shorter on one or both ends based on a user-defined distance. In this case a *cut* value of 1 point resulted in the correct symbol appearance, as illustrated in Figure 4-39.

Figure 4-39 - PLSS lines with Cut Curve geometric effect applied, note gaps at line ends.

The use of cartographic representations in this example provided a solution not available with standard ArcMap symbol management tools.

4.2.4. Background vector polygon layer for vegetation cover

As discussed in Section 2.2, the inclusion of raster information on the DRG map was not an issue for cartographic representations. However, this map was initially created using a single raster layer symbolized with a single solid fill color to represent vegetation cover. The information contained within the raster layer (*veg_cov*) was not a continuous or highly variable surface and, therefore, could as easily be shown as a polygon with a simple fill. Further, the original raster cell size was 25 meters, which when drawn on the map resulted in a significant amount of the jagged-edged boundaries typical of low-resolution raster data, as shown in Figure 4-40.

Figure 4-40 – DRG map vegetation cover drawn from raster data source, showing excessive pixel size for map scale.

To resolve this problem and also to try keeping the size of the geodatabase to a minimum, the raster data was converted to a vector polygon feature class. This was achieved through the use of the ArcGIS ArcScan extension, which contains tools to control the amount of smoothing in the output vector polygons. The raster-to-vector conversion was processed using the default settings for the Vectorize tool. As a result, the output polygon feature class contained numerous small polygons which were derived from a single 25-meter pixel. Additional experimentation with the ArcScan vectorization settings to reduce the number of these small raster-to-polygon artifacts was not undertaken, as the appearance was deemed acceptable for the purposes of the test map. The final DRG map was drawn using only vector format data sources. The result is shown in Figure 4-41.

Figure 4-41 – DRG map vegetation cover drawn from vector data source.

4.3. TOPO map

The cartographic representation techniques applied to the TOPO map were:

1. New road line symbols
2. Warp tool for bathymetric contour lines
3. New marker symbols
4. New park boundary symbol

This map and the TOPO_OLD map each presented a design challenge, as the source maps were scanned .jpg images. This data format supplied a poor quality image from which to capture feature symbology, thus it was difficult to appropriately overlay GIS data on the image. The image supplied by the client had been georeferenced by the data provider, but because the original document had been folded and stretched over time, the resulting georeferencing was only an approximation (USGS 2007b).

Work on the TOPO map proceeded quickly, as most of the feature classes used symbology very similar to that of the DRG map. The main differences between these two maps were that the TOPO map contained generally more simple feature symbology and fewer mapped features. As described below, the limited number of vector feature classes required for this map already contained cartographic representations and representation

rules for the previously defined maps. These symbol sources were leveraged to reduce the amount of work required to define the new feature representations for this map.

4.3.1. Remove selected road feature representations from display

The feature class representation for the road symbols for this map was created by copying the existing feature class representation already created for the DRG map. The new feature class representation was required to contain the simple change required, namely to reduce the line width for both representation rules.

This use case provided another example of the technique described in Section 4.1.8.1: the removal of a feature representation being displayed on the map by editing the RuleID value to “No rule applied” in an ArcMap edit session. For this map, historical changes in the actual constructed roads in Crater Lake National Park meant that the newer maps (NPS, DRG) showed roads not yet built when the TOPO map was printed. However, the *road_line_NPS_dissolved* feature class contained temporally current information. In order to maintain the historical accuracy of the TOPO map, selected roads were edited to have no representation rule applied, so they would not be displayed on the map. Representation layers have a setting, turned on by default, which displays feature representations with invalid or null RuleIDs. These are shown in Figure 4-42 with a thick red line (these features might also be drawn as point or polygon boundary geometries, as appropriate). This reflects the software design assumption that if a feature does not have a representation rule applied, there may be an error. However, as shown here, it may be intentional that a feature has no representation rule. The feature layer representation properties allow the user to select which features not to display, while retaining the data completeness of the source feature class geometry.

Figure 4-42 - Mazama Village detail showing roads with no representation rule applied, drawn in red.

This feature representation management technique offers enhanced control of the display of individual features over standard ArcMap symbol management tools. In using this function, the feature exception is stored as a property of the feature class, rather than within a layer file or map document.

4.3.2. Warp tool for bathymetric contour lines

The dataset supplied by the client did not contain data for the bathymetric contours, therefore, the TOPO map required heads-up digitizing to create a new data source. As the features on the map were fairly small in number and spatial extent, the creation of these data was a simple task. The original .jpg file was used as the source from which the new features were traced. However, upon digitizing the features, a problem became apparent. The source image was the product of a scanned hard copy, and as a result warping in the paper from where the map had been folded and stretched was present in the .jpg image. Consequently, the boundary of the lake on the map did not align with the vector polygon data source provided in the client geodatabase, which meant that the digitized bathymetric contours were not properly placed either. These errors having been identified, the features were nonetheless digitized from the original source, with the intention of correcting the errors after converting the features to representations, as explained below. A new annotation feature was created which stored the text for the isobaths after the new *lake_bathy_contours* feature class was digitized.

As shown in Figure 4-43, the bathymetric contours were out of alignment at the north-western edge of the lake. The ArcMap Representation toolbar contains a Warp tool whose function is suited to the repair of feature mis-alignment such as in this use case.

Figure 4-43 - Mis-aligned bathymetric contours are a result of digitizing a folded and stretched paper map that was scanned.

Correcting these errors required that the relative relationships between the separate bathymetric lines be preserved; simply moving one line would result in the line being too close to the next one, and so on. The Warp Tool provides the user with the ability to select multiple feature representations, or only selected segments, and manually move them. The feature representations can be from one or multiple feature class representations. As shown in Figure 4-44, only the bathymetric line sections most in error were selected and moved, i.e., the three line feature representation segments were displaced simultaneously.

Figure 4-44 - Use of the Representation Warp Tool to move selected bathymetric contour feature representations. The original location of feature representations are shown in blue as selected lines and vertices. The black lines show the new location of the feature representations.

Standard ArcMap editing tools could have been used to achieve a similar solution to the cartographic problem, but would have required the separate editing of each individual feature's geometry, and would not have accurately preserved the original position of the bathymetric contour lines relative to one another. The use of representations to repair this error provided benefits in both efficiency of data management and the preservation of the original data. It is important to note here that a prior understanding of the capabilities of the ArcMap representation editing tools was fundamental to the success of this approach to the data management workflow.

4.3.3. New marker symbols for park facilities

The TOPO map contained a simple set of point marker symbols similar to those already used in the NPS and DRG maps, but stored in two separate point geometry feature classes, each of which contained a feature class representation with a single representation rule. It was at this point in the project that an opportunity for data consolidation became apparent. In keeping with the overall project goal of producing the most concise resultant geodatabase, it was obvious that rather than storing two separate

point feature classes with similar symbology (*cultural_pts*, *building_pts*), data management efficiency could be gained by combining them into a single feature class which stored multiple feature class representations. It was decided that the point features from the DRG map *building_pts* feature class would be appended to the more extensive NPS map *cultural_pts* feature class.

The workflow was performed in four steps:

1. all layers were added to ArcMap, and the representation renderer was applied;
2. the Add Representations geoprocessing tool was used to copy the *building_pts* rules to the *cultural_pts* feature class as new representations – the tool option to assign rule IDs was left unchecked;
3. the Append geoprocessing tool was used to add the *building_pts* point features to the *cultural_pts* feature class; and
4. the appropriate representation rules were applied to the new features.

The result was a single point feature class with two representations, derived from separate data sources. This workflow represents a suggested practice for consolidating data in a cartographic database where separate data sources may currently be maintained solely for cartographic symbol management. This may be helpful where point geometry features need to be used for mapping across a broad range of map scales.

This use case may be viewed as experimental, as the move to consolidate the data was driven by database management considerations, rather than by cartographic production priorities. Combining two feature classes may be viewed as “de-normalizing” the database by grouping similar geometries by theme rather than retaining the discrete feature classes as originally derived from different maps.

4.3.4. New park boundary symbol

The addition of a new representation to the *park_bndy_USGS* feature class for the TOPO map was straightforward, but is worthy of mention here as the source symbol was copied into the representation rule directly from an ESRI style file. Cartographic representation functionality supports seamless interoperability between file-based ESRI symbol storage and the geodatabase. ESRI users who have already made an investment in defining and storing custom symbol libraries in style files may directly import those symbols into a representation rule by using the representation feature class’ layer properties. Alternatively, users who have created symbols stored as representation rules may export these to an ESRI style file, so that they can be used by other ArcGIS users who may not have implemented cartographic representations or are using older versions of the software.

For this use case, the client supplied a TOPO.style file, as shown in Figure 4-45, which contained a number of symbol definitions based on USGS standards. The new representation rule was added as “USGS bndy” and compared to the “TOPObndy” rule, which had earlier been created from scratch.

Figure 4-45 - Park boundary symbol as stored in TOPO.style file

Figure 4-46 - Park boundary symbol imported as representation rule.

This procedure was a simple way to create a new feature representation symbol definition, and illustrates the seamless method through which symbol definitions stored in a style file can be easily converted to geodatabase information.

4.3.5. Background vegetation cover layer

This map used the same data source for the background vegetation cover as that created for the DRG map, as previously discussed in Section 4.2.4. The only change required was to apply a different color to the polygon fill in order to match the background on the source TOPO map.

4.4. TOPO_OLD map

As this map was the final one compiled for the project, much of the work had been completed from previous maps (e.g., rotated building markers from the DRG map and bathymetric contours from the TOPO map). It was also the map with the fewest number of required map layers, as well as the most simple feature symbology. The only unique use case for representations on this map involved the road symbols.

1. Remove selected road feature representations from display

4.4.1. New masking layer for roads

As first discussed in Section 4.2.2, the use of a feature masking layer for the road symbols was required for this map, similar to that applied on the DRG map. One difference was that the mask was not required for all of the main road features, i.e., those drawn with a solid line, but only to a selected subset of campground roads within the Mazama Village and Munson Valley areas, which were drawn with a cased line. As the number of features which required masking was small and limited to two map areas, they were first manually selected, and the Feature Outline Mask geoprocessing tool was used with just the selected set of features as input. The result is shown in Figure 4-47.

Figure 4-47 - Mazama Village detail showing roads with no representation rule applied for TOPO_OLD map. Compare this with the same area shown in Figure 4-42.

4.4.2. Remove road features from display

Since the TOPO_OLD map was the oldest of the four in the project series, the roads shown on the map represented the earliest “as-built” state of the Crater Lake Park infrastructure, and were thus the fewest in number. To illustrate this, compare Figure 4-42 to Figure 4-47 and note the additional feature representations removed from the map. As previously discussed, a feature representation edit was used to remove selected features from display. The same technique was also applied at the Rim Village area, as shown in Figure 4-48.

Figure 4-48 - Rim Village detail showing roads with no representation rule applied.

4.4.3. Raster hillshade

The TOPO_OLD map was the second map in the project series which required the use of a raster data source for a significant portion of its content, in this case for the hillshade effect. As with the NPS map, no geoprocessing was performed on the source data, and only minimal experimentation was required to change the color ramp from the default grey scale to that shown above, as on the source map.

4.5. Annotation

The final step for the completion of each map required the addition of text elements in the form of geodatabase annotation. As discussed in Section 2, it was recognized early in the project that the creation of unique annotation for each map was beyond the scope of the project. Therefore, it was decided that the annotation created for NPS map, already contained within the original Crater Lake geodatabase, would be used as the default set for all four maps, recognizing that the scale of the annotation would not be correct across all four test maps. However, some solution was required to include at least minimal textual information on each map.

The original annotation feature classes from the NPSreps.mxd were exported as a Group Layer and added to each of the other three maps. Subsequently, each of the other maps' annotation settings were minimally altered using the symbol substitution functionality available with each annotation feature layer. Two new annotation feature classes were required for the PLSS range and township lines and the bathymetric contour lines. These

new annotation feature classes were added to the existing annotation group layer for each map.

5. Summary of Cartographic Representations and Final Geodatabase

The project test maps provided a wide range of use cases from which to evaluate cartographic representations. While the maps themselves were varied in content and provided a range of feature symbologies, the data model from which the maps were derived was quite simple. This section will summarize the cartographic representations described in Section 4, and review the evolution of the geodatabase through the project.

5.1. Summary of Cartographic Representation techniques used in the project

The test maps provided a number of opportunities to evaluate a comprehensive range of cartographic representation functionality but did not require the use of all the available desktop application toolbar buttons, representation geometric effects, or geoprocessing tools. The symbol specifications for most of the use cases identified in the test maps were not complex, the bog symbol being the exception.

Table 5-1 lists the vector feature classes for which cartographic representations were evaluated, with a summary overview of the techniques discussed in Section 4. Raster layers, road masking feature classes, and annotation feature classes also stored in the *CraterLakeReps.gdb* project geodatabase are not described here.

Table 5-1 – Summary of cartographic representations used on all test maps.

Feature Class Name (vector data only)	Representation Name Per Map			
	NPS	DRG	TOPO	TOPO_OLD
building_polys	None	None	None	None
CLNP_ownership_lines	NPSowner: 1 representation rule with enhanced line dash pattern	None	None	none
contours_USGS	None	None	None	None
cultural_pts	NPSrep: 6 representation rules, three with 'stacked' marker symbols, 3 with single markers	DRGrep: 3 representation rules for simple marker symbols; aligned to closest line feature with new GP tool; also used this rule for TOPO_OLD	TopoRep: 3 representation rules for single markers	Used DRGrep
DRGveg_poly	None	None	None	None
hydro_pts_USGS	NPSHydroP: 2 representation rules, assigned angle attribute field to marker symbols	None	None	None
lake_bathy_contours	None	None	TOPOreps: 1 representation rule, used Warp Tool to edit multiple feature representations	

lake_polys_NPS	None	None	None	None
OR_plss_lines	None	DRGplssL: 1 representation rule, enhanced line dash pattern, assigned Cut Curve to line ends	Used DRGplssL	Used DRGplssL
park_bndy_USGS	GradBdy: tested but not used NPSbndy: 1 representation rule	None	USGSbdy: tested but not used TOPObdy: symbols imported from USGS.style	Used TOPObdy
parkbndy_buffer	None	None	None	none
physiog_polys	NPSphys: 1 representation rule for enhanced bog marker symbols	None	None	None
road_lines_NPS_dissolved	NPSroads: 2 representation rules with enhanced line caps	DRGroads: 2 representation rules with enhanced line dash patterns; road masking layers applied	TOPORoad: copied DRGroads, narrowed line widths, removed selected lines from display as "No rule applied"	TOPOold Roads: copied TOPORoad, narrowed line widths,
stream_lines_dissolved	NPSstreams: 2 representation rules, enhanced line dash pattern	None	None	None
stream_polys_NPS	None	None	None	None
survey_pts	None	None	None	None
trail_lines_dissolved	NPStrails: 2 representation rules with enhanced line dash patterns	DRGtrails: simple dash line pattern	Used DRGtrails	Used DRGtrails

A total of 19 vector feature classes were required for all four test maps - the 17 listed above, plus two masking layers for the road features on the DRG and TOPO_OLD maps. Three raster layers are also contained in the final geodatabase: *bathy_dem* and *bump_map* for the NPS map, and *topo_hillshade* for the TOPO_OLD map. As can be seen in Table 5-1, seven of the vector feature classes did not require the use of cartographic representations. Of those seven, only one, *parkbndy_buffer*, is an instance where the application of cartographic representations failed to produce a satisfactory enhancement in the feature symbology. Therefore, of the 11 feature classes for which the application of cartographic representations provided a potential benefit, 10 were successfully implemented.

It should be noted that these results are specific to the data and test maps used for this project, and should not be extrapolated as a benchmark for implementing cartographic representations for other mapping projects. Moreover, although the use cases described above represent the use of a substantial portion of the ArcGIS representation management tools, not all of the functionality was evaluated. For example, the Create Overpass

geoprocessing tool, which provides the user with a single operation to create a masking layer, relationship class, and optional bridge parapet feature class, was not required. Further, the ArcGIS geoprocessing tools are designed for automating data management workflows through the use of ModelBuilder and scripts. As this project used a relatively small set of feature classes within a single file geodatabase, these types of geoprocessing tools were not required.

5.2. Final geodatabase summary

The main goal of the project was to create a single geodatabase which would support the production of multiple maps. As first discussed in Section 3.3, the decision to use the file geodatabase format for the final project deliverable was made for two main reasons: its ability to store raster information, and its higher compression ratio than the original geodatabase format. A general aim of the best practices in managing any data on a computer is to reduce the amount of disk space required to store information. In this regard, the project was successful in reducing the amount of storage space required to support the four test maps.

The original geodatabase, named “Crater_Lake.mdb” in Figure 5-1, was supplied by the client with both “managed” and “unmanaged” (these terms are specific to ESRI data management) raster files, as well as a number of experimental feature classes. The total size of the data on disk, including rasters, was 2.1 gigabytes. By comparison, the first test geodatabase, named “CraterLake.gdb” in Figure 5-1, occupied 1.79 gigabytes. The final project geodatabase, named “CraterLakeReps.gdb” in Figure 5-1, occupies 79.8 megabytes of disk space. As a final comparison, the identical geodatabase schema and data was converted back to the Personal geodatabase for Access format; the total size, including managed rasters, was 145 megabytes.

Figure 5-1 – ArcCatalog Trees showing changes to project geodatabases.

6. Discussion

The project was successful in providing the client with a set of four map documents based on a single geodatabase. A prototype methodology for implementing cartographic representations from an existing geodatabase was defined following the completion of the maps. These components provided the client with qualitative results from which to evaluate the efficacy of implementing cartographic representations to enhance map production and feature symbology.

As discussed in Section 3, the overall design approach for the test maps was to mimic *finished* map products. This method conforms to the methodology suggested by the ESRI literature and product documentation with regard to implementing cartographic representations (ESRI 2007). That is, once the final maps were created, they were re-evaluated for conformance to those special feature symbol problems that were correct in the original maps and may only have been resolved through the use of cartographic representations. However, subsequent research indicates that there are more effective methods for implementing cartographic representations which do not require that the map is “finished” prior to their application. The results of this project suggest that this design approach can be augmented to include alternatives. Broadly, these alternatives may be defined in three ways:

1. Create cartographic representations for all feature classes within a geodatabase, regardless of symbol complexity. In so doing, the feature representations are permanently available to all users who have access to the data, and they are not reliant upon file-based information contained within map documents, layer files, style files, or fonts to define feature symbols. New maps may be created at any time and will retain consistent symbols that conform to map standards.
2. Create cartographic representations from an original source dataset, but export the representation rules to a style file. This detaches the feature symbology from an explicit data source for use by other cartographers across other computer networks or geodatabases. Again, new maps may be created with consistent map symbol standards, but they remain independent of fixed and explicit data sources. This facilitates workflows that are disjointed, either temporally or physically, typically for information security reasons.
3. Create data collection methods and design the geodatabase feature class schema to capture and store comprehensive feature symbol attributes, e.g., line width, color, rotation angle, etc., which will subsequently be used to define representation rule properties.

Although this project used aspects of all of these approaches to some degree, the use of raster layers for two of the maps (NPS and TOPO_OLD) and annotation for all of the maps required that map documents be included with the final deliverables.

6.1. Advantages of implementing cartographic representations

In two use cases, cartographic representations provided a solution that was not possible to achieve with standard ArcMap tools: a) automatically rotating marker symbols to align with the nearest line (DRG map); b) converting font-based marker symbols to feature attributes (NPS map). In addition, the conversion of the *lake_bathy_contours* to representations provided editing functionality which was significantly easier to manage than using default geometry editing tools; the underlying feature geometry was unaltered, but the map symbology was enhanced. The majority of the data used in the test maps was line geometry, thus the main benefit of converting standard symbols to representations was in the enhanced control of line ends and intersections, e.g., the ability to apply the Cut Curve geometric effect to the PLSS lines.

6.2. Final geodatabase design

The main purpose of the project was achieved, namely to supply the client with a single geodatabase that contained the vector symbology for all four of the test maps, as well as the minimum necessary raster and annotation data sets. As discussed in Section 4.3.3, additional reduction in the number of original feature classes was achieved through the consolidation of multiple feature classes into a single feature class with multiple representations. This represents a considerable improvement for data management and offers other users the ability to create maps whose symbology will appear identical to those created here directly from the final project geodatabase.

6.3. Feature class attributes for cartographic representations

The project approach to create representation rule properties for features on each of the maps was to define the symbols manually, in most cases by first defining the basic feature symbols with standard ArcMap tools and then converting those to representations. The NPS map was supplied by the client along with an existing ArcMap map document containing complete symbol definition properties, thus representation rules were simple to generate from the original data source. As the other map sources were scanned image files of the original paper maps, the representation rules for these maps had to be created from scratch or modified from those which were created for the NPS map.

The full representation rule definitions for almost all of the maps are contained within the RuleID or Override fields for each individual feature class representation, as opposed to being defined by existing feature class attributes, which did not exist for most of the data. The exception is the *hydro_pts* feature class, which contained an attribute field for a rotation angle. As discussed in Section 4.1.8, the use of this field to dynamically place each feature representation marker symbol with its correct rotation required additional data processing. The simple markers for the *cultural_pts* buildings, which were rotated to align to the closest road, used a rotation angle stored within the Override field. Thus, the ability to explicitly store feature symbol attributes within the feature class attribute table was not required for most of the data. Other than *hydro_pts*, the original data did not contain attributes which could be used for symbol definitions, nor was there sufficient

complexity within the symbol range to require that any of these attributes be exposed for future use.

The ability to associate existing feature attributes with representation properties gives the user the ability to edit an attribute value and have that edit dynamically alter the feature representation. This functionality will require updated approaches to geodatabase design and data capture methods in order for ESRI users to realized the full potential of cartographic representations.

6.4. Cartographic representations in the multi-user geodatabase environment

This project was specifically designed and implemented for use with the file geodatabase storage format for a single user/editor, which can be considered a typical architecture for most ESRI clients for whom high-volume cartographic production is not a core business requirement. This system architecture is suitable for organizations with low-volume map production (i.e., one at a time) or a cartographic editing environment with very few individual users (i.e., three or fewer). The file geodatabase format was adopted primarily for its ability to store raster information, but has the added benefit of enhanced compression over the .mdb geodatabase format.

An alternative computer system architecture that supports data distribution across networks is the Relational Database Management System (RDBMS). The use of multiple RDBMSs such as Oracle or SQL Server to store geodatabases is widely implemented throughout ESRI's customer base through the installation of the Spatial Database Engine (SDE) technology. It is certain that cartographic representations will be implemented by those ESRI users who store their cartographic information using SDE within an enterprise-level RDBMS. This system architecture supports multiple concurrent editors, robust data storage capacity, and all the database administration functionality available with the native RDBMS tools. However, the ESRI product documentation provides only a general overview of how cartographic representations may be implemented within a multi-user, multi-editor ("versioned") environment (ESRI, 2007).

Consultation with the ESRI Professional Services Group suggests that the best practice for managing cartographic representations in a versioned geodatabase is to first create the feature class representations and all required representation rules in the *default* (i.e., original master copy) geodatabase instance (C. Loveman, personal communication, March 21, 2007). At present, the creation of efficient workflows, and the methods by which users will manage potentially conflicting edits, representation rule definitions, or changes to the underlying geometry remains only partially defined. The technology is new, and more time is required to gain sufficient real-world application experience. These issues offer significant opportunity for future research.

6.5. Data capture to support cartographic representations

As noted in Section 2, GIS data are typically collected to support spatial analysis or data warehousing. An area of current research involves the capture of raw data, or the conversion of existing data sources with the intention of their use in cartographic production (Buckley *et al*, 2005; Frye, 2006). The specific attribute data type requirements for cartographic representations presents a need for additional geodatabase design and data collection planning. For example, as discussed in Section 4, the rotation angle attributes used for the *hydro_pts* feature representation needed to be converted to a float data type. The original attributes had been collected as an integer data type.

The creation of efficient and comprehensive workflows for data capture in support of cartographic representations will demand the combination of the system designers' knowledge of the required valid feature attribute values, as well as the specific data types required by cartographic representations. A table listing the valid field data types required for mapping feature class attributes to representation rule properties is contained in the Appendix A.

Additional research into this area may provide benefits in two main areas: the enhancement of enterprise geodatabase design; and the development of efficient workflows in support of high-quality cartographic production.

7. Conclusion

This project began with the premise that the new cartographic representation functionality released with ArcGIS 9.2 software would provide enhancements to a map production workflow by allowing the user to store feature symbology for multiple maps within a single geodatabase. The methodology defined in Section 3 is the result of an iterative process which began with a single geodatabase, four maps, and a library of feature symbols stored within a variety of file-based objects. The project scope of work involved converting all of the feature symbology into feature class attributes. The project approach was to emulate a real-world workflow for a typical ESRI desktop user using standard ArcGIS 9.2 ArcInfo desktop tools and a local hard drive. Prior knowledge of the capabilities of cartographic representations was a fundamental requirement for the success of the project. This point emphasizes the importance of acquiring comprehensive knowledge of the software capabilities in order to effectively define an efficient map production workflow.

This work will be useful to those ESRI users who are experienced cartographers but may be unfamiliar with the potential of cartographic representations, database administrators who need to understand how symbol storage in the database will impact geodatabase schema and disk space allocation, and any ESRI users who are new to ArcGIS desktop software and need to understand its capabilities. Adoption of this new technology will require changes in methodology for cartographic geodatabase design and data management in order to fully exploit its potential. Fundamentally, cartographic production with ArcGIS can now be considered part of data modeling and geoprocessing workflow, rather than as layout-based graphic data manipulation.

Two main areas for further research which would build on or enhance the results of this project are: managing cartographic representations in a multi-user enterprise-level geodatabase, and managing annotation in conjunction with cartographic representations.

The research described in this report represents a practical, real-world evaluation of a new technology, and as such contributes to the larger body of knowledge pertaining to the use of ArcGIS Desktop software.

8. References

- Arctur, D., & Zieler, M. (2004). *Designing geodatabases: Case studies in GIS data modeling*. Redlands, California: ESRI Press.
- Brewer, C., & Frye, C. (2005). *Comparison of GIS and graphics software for advanced cartographic symbolization and labeling: Five GIS projects*. Proceedings of the International Cartographic Association, A Coruña, Spain.
- Buckley, A., & Hardy, P. (2007). Cartographic software capabilities and data requirements: Current status and a look toward the future. *Cartography and Geographic Information Science*, 34(2) 155-157.
- Buckley, A. R., & Frye C. (2007, August). *Lessons learned in cartographic data modeling in support of GIS-based map production*. Proceedings of the International Cartographic Association, Moscow, Russia.
- Buckley, A. R., Frye C., Battenfield, B., & Hultgren, T. (2005). An information model for maps: Towards cartographic production from GIS databases. *AutoCarto 2005*, Cartography and Geographic Information Society.
- Battenfield, B., & Frye, C. (2006, June). *The fallacy of the "Golden Feature" in MRDBs: Data modeling versus integrating new anchor data*. Proceedings of the International Cartographic Association Commission on Map Generalisation and Multiple Representation, Vancouver, Washington.
- ESRI (2007). *ArcGIS Desktop Help* (Version 9.2) [Computer software]. Redlands, California: Environmental Systems Research Institute, Inc.
- Frye, C. (2006, June). A product-driven approach to designing a multi-purpose multi-scale GIS base map database that supports high quality mapping. *AutoCarto 2006*. Cartography and Geographic Information Society. Retrieved April 23, 2007, from <http://www.cartogis.org/publications/autocarto-2006/frye.pdf/view>
- Frye, C., & Eicher, C. (2003, August). Modeling active database-driven cartography within GIS databases. *21st International Cartographic Conference*. The International Cartographic Association.
- Hardy, P., & Kressmann, T. (2005, April). Cartography, database, and GIS: Not enemies, but allies! *AutoCarto 2005*. Cartography and Geographic Information Society.
- Hardy, P., Eicher, C., Briat, M., & Kressmann, T. (2005, July). Database-stored representations and overrides, supporting automated cartography with human creativity. *22nd International Cartographic Conference*. The International Cartographic Association.
- Punt, E., Jensen, R., Hardy, P., & Daly, T. (2006). Smarter symbols: Smarter maps. *AutoCarto 2006*. Cartography and Geographic Information Society. Retrieved April 23, 2007, from <http://www.cartogis.org/publications/autocarto-2006/~puntjensenhardydaly.pdf/view>

- USGS Cascades Volcano Observatory (2007a). *Crater Lake, Oregon Topographic Map*. Retrieved April 23, 2007, from http://vulcan.wr.usgs.gov/Volcanoes/CraterLake/~Maps/map_crater_lake_topo.html
- USGS (2007b). *Topographic Map Symbols*. Retrieved April 23, 2007, from <http://erg.usgs.gov/isb/pubs/booklets/symbols/>
- USNPS (2002). *Crater Lake National Park* [Map Brochure]. Crater Lake, OR.

Appendix A: Valid Field Types for Cartographic Representation Properties (ESRI, 2007).

Numeric Value	Float, Double
Drop Down List	Short Integer, Long Integer
Check Box	Short Integer, Long Integer, Text
Color Swatch	Blob

DEFAULT SYMBOL LAYERS			
RULE SYMBOL LAYER TYPE	PROPERTY	FIELD INPUT	FIELD TYPE
Markers	Marker	Element	Blob
	Size	Numeric Value	Float, Double
	Angle	Numeric Value	Float, Double
Strokes	Width	Numeric Value	Float, Double
	Caps	Drop down list	Short Integer, Long Integer
	Joins	Drop down list	Short Integer, Long Integer
	Color	Color Swatch	Blob
Fills	Color	Color Swatch	Blob

GEOMETRIC EFFECTS			
POINTS	PROPERTY	FIELD INPUT	FIELD TYPE
Buffer	Buffer Size	Numeric Value	Float, Double
Buffer - Gradient option	Color 1	Color Swatch	Blob
	Color 2	Color Swatch	Blob
	Algorithm	Drop down list	Short Integer, Long Integer
	Style	Drop down list	Short Integer, Long Integer
	Intervals	Numeric Value	Short Integer, Long Integer
	Percentage	Numeric Value	Float, Double
	Angle	Numeric Value	Float, Double
Buffer - Hatch pattern	Color	Color Swatch	Blob
	Width	Numeric Value	Float, Double
	Angle	Numeric Value	Float, Double
	Step	Numeric Value	Float, Double
	Offset	Numeric Value	Float, Double
Radial	Angle	Numeric Value	Float, Double
	Length	Numeric Value	Float, Double

LINES	PROPERTY	FIELD INPUT	FIELD TYPE
Add Control Points	Angle tolerance	Numeric Value	Float, Double

Buffer	Buffer Size	Numeric Value	Float, Double
Cut curve	Begin cut	Numeric Value	Float, Double
	End cut	Numeric Value	Float, Double
Dashes	Pattern	Numeric Value	Text
	Endings	Drop down list	Short Integer, Long Integer
Enclosing polygon	Method	Drop down list	Short Integer, Long Integer
Offset curve	Offset	Numeric Value	Float, Double
	Method	Drop down list	Short Integer, Long Integer
	Simple	Check Box	Short Integer, Long Integer, Text
Reverse curve	Reverse	Check Box	Short Integer, Long Integer, Text
Simplify	Tolerance	Numeric Value	Float, Double
Smooth	Flat Tolerance	Numeric Value	Float, Double

POLYGONS			
	PROPERTY	INPUT TYPE	FIELD TYPE
Buffer	Buffer Size	Numeric Value	Float, Double
Cut curve	Begin cut	Numeric Value	Float, Double
	End cut	Numeric Value	Float, Double
Dashes	Pattern	Numeric Value	Text
	Endings	Drop down list	Short Integer, Long Integer
Donut	Donut Width	Numeric Value	Float, Double
	Simple	Check Box	Short Integer, Long Integer, Text
Enclosing polygon	Method	Drop down list	Short Integer, Long Integer
Simplify	Tolerance	Numeric Value	Float, Double
Smooth	Flat Tolerance	Numeric Value	Float, Double

MARKER PLACEMENT STYLES			
POINTS	PROPERTY	FIELD INPUT	FIELD TYPE
On point	X Offset	Numeric Value	Float, Double
	Y Offset	Numeric Value	Float, Double

LINES	PROPERTY	FIELD INPUT	FIELD TYPE
Along line	Step	Numeric Value	Text
	Endings	Drop down list	Short Integer, Long Integer
	Control points	Drop down list	Short Integer, Long Integer
	Angle to line	Check Box	Short Integer, Long Integer, Text
At extremities	Extremity	Drop down list	Short Integer, Long Integer
	Position	Numeric Value	Float, Double
	Angle to line	Check Box	Short Integer, Long Integer, Text
Decorations	# of markers	Numeric Value	Short Integer, Long Integer
	Begin position	Numeric Value	Float, Double
	End position	Numeric Value	Float, Double
	Flip all	Check Box	Short Integer, Long Integer, Text
	Flip First	Check Box	Short Integer, Long Integer, Text
	Angle to line	Check Box	Short Integer, Long Integer, Text
On line	Position	Numeric Value	Float, Double
	Relative to	Drop down list	Short Integer, Long Integer
	Angle to line	Check Box	Short Integer, Long Integer, Text
Randomly along	Step	Numeric Value	Float, Double
	Endings	Drop down list	Short Integer, Long Integer
	Control points	Drop down list	Short Integer, Long Integer
	Randomization	Drop down list	Short Integer, Long Integer
	Skew effect	Check Box	Short Integer, Long Integer, Text
Variable size	Step	Numeric Value	Text
	Endings	Drop down list	Short Integer, Long Integer
	Control points	Drop down list	Short Integer, Long Integer
	Minimum zoom	Numeric Value	Float, Double
	Maximum zoom	Numeric Value	Float, Double
	Zoom Number	Numeric Value	Short Integer, Long Integer
	Method	Drop down list	Short Integer, Long Integer
	Offset	Numeric Value	Float, Double
POLYGONS	PROPERTY	FIELD INPUT	FIELD TYPE

Along lines	Step	Numeric Value	Text
	Endings	Drop down list	Short Integer, Long Integer
	Control points	Drop down list	Short Integer, Long Integer
	Angle to line	Check Box	Short Integer, Long Integer, Text

Decorations	# of markers	Numeric Value	Short Integer, Long Integer
	Begin position	Numeric Value	Float, Double
	End position	Numeric Value	Float, Double
	Flip all	Check Box	Short Integer, Long Integer, Text
	Flip First	Check Box	Short Integer, Long Integer, Text
	Angle to line	Check Box	Short Integer, Long Integer, Text

Inside polygon	X step	Numeric Value	Float, Double
	Y step	Numeric Value	Float, Double
	Grid angle	Numeric Value	Float, Double
	Shift odd rows	Check Box	Short Integer, Long Integer, Text
	X Offset	Numeric Value	Float, Double
	Y Offset	Numeric Value	Float, Double
	Clipping	Drop down list	Short Integer, Long Integer

On line	Position	Numeric Value	Float, Double
	Relative to	Drop down list	Short Integer, Long Integer
	Angle to line	Check Box	Short Integer, Long Integer, Text

Polygon center	X Offset	Numeric Value	Float, Double
	Y Offset	Numeric Value	Float, Double
	Method	Drop down list	Short Integer, Long Integer
	Clipping	Drop down list	Short Integer, Long Integer

Randomly along	Step	Numeric Value	Float, Double
	Endings	Drop down list	Short Integer, Long Integer
	Control points	Drop down list	Short Integer, Long Integer
	Randomization	Drop down list	Short Integer, Long Integer
	Skew effect	Check Box	Short Integer, Long Integer, Text

Randomly inside polygon	X step	Numeric Value	Float, Double
	Y step	Numeric Value	Float, Double
	Clipping	Drop down list	Short Integer, Long Integer

		Integer
--	--	---------

Variable size

Step	Numeric Value	Text
Endings	Drop down list	Short Integer, Long Integer
Control points	Drop down list	Short Integer, Long Integer
Minimum zoom	Numeric Value	Float, Double
Maximum zoom	Numeric Value	Float, Double
Zoom Number	Numeric Value	Short Integer, Long Integer
Method	Drop down list	Short Integer, Long Integer
Offset	Numeric Value	Float, Double