MULTIPLE REPRESENTATIONS WITH OVERRIDES, AND THEIR RELATIONSHIP TO DLM/DCM GENERALIZATION

Paul Hardy, Dan Lee
ESRI Redlands, California
phardy@esri.com, dlee@esri.com

KEYWORDS: Multiple Representation, Database Cartography, DLM/DCM/DKM, Overrides, Generalization

ABSTRACT

The introduction of new capabilities for multiple cartographic representations and overrides within a GIS database opens up alternative approaches to cartographic production. Many cartographic production organizations have expressed a strategy of building a Digital Landscape Model (DLM) in a central database, and from that deriving a range of cartographic products. To do this efficiently will require generalization tools and mechanisms for handling the Digital Cartographic Models (DCM) including multiple representations. It will also require a framework for controlling the flow of data from DLM to DCM, including updates.

This paper provides an update on a project presently underway at ESRI to implement high-quality cartography with multiple representations in the database, including override mechanisms that empower the cartographer to modify individual feature representations without affecting the master DLM data. It then relates that to ongoing development to provide efficient generalization processes and a robust framework for such DLM/DCM data derivation.

1 INTRODUCTION

1.1 Database, GIS and Cartography

The last few decades have seen a transition to using GIS tools to facilitate compiling and publishing maps, charts, plans, and other visual spatial products. Previous papers have discussed the relationship of GIS to cartography:

"Traditional map-centered cartography is a highly complex and wonderful process in which information is compiled from map sources and represented graphically to communicate some meaning to the map user. … GIS-based cartography separates this process into three parts: compilation of a geographic database according to database schema, processing tools which allow geographic information to be translated from one form to another, and rendering tools which translate objects in the geographic database into symbols on the map." [Morehouse 1995].

1.2 Industry Requirements

Many national and commercial mapping agencies have a strategic goal of using a common database and common environment for all map publishing, and a 'capture once and use many times' ethos [Lawrence 2004].This requires:

- A single software environment from capture to finishing
- Centered on database
- Supporting multiple representations for multiple products
- Handling generalization and incremental update as these advanced facilities mature

1.3 Solution

This paper describes the multiple representation and generalization project underway at ESRI to provide the framework and tools needed for database-centered map publishing using multiple representations with overrides, extending to cover generalization for automatic derivation of reduced-scale products. The early snapshot of this project was outlined in [Hardy 2004] and [Hardy et al 2005], but has evolved and progressed as described below.

The project adds new cartographic capabilities to ArcGIS. In particular it adds:

- Storage of multiple representations, closely associated with feature classes in the geodatabase
- Representation rules that define high quality cartographic visualizations of the features
- Override mechanisms that allow a cartographer to make exceptions to the rules.
- Intuitive editing tools for cartographic interaction, similar to those in desktop graphics packages

The following sections of the paper outline these capabilities in turn, and relate them to the overall framework for cartographic data derivation and generalization.

2 MULTIPLE REPRESENTATIONS WITH OVERRIDES

2.1 Cartographic representation

The conceptual basis for cartographic representation has been the subject of extensive academic analysis [MacEachren 1995], [Fairbairn et al 2001], but there has been continued difficulty in resolving the conflicting pressures of automation (rule-driven visualization) with those of cartographic clarity (freedom of expression). Giedre Beconyte in a recent paper on “Conceptual Models for Cartographic Representation” states “*Other than in the simplest cases, it is impossible to limit cartographic design to a single set of rules at all; hence thematic mapping can hardly be subject to automated processing functions*” [Beconyte 2004]. The Representations and Overrides system which is summarized in this paper and described more fully in [Hardy, Eicher, Briat, Kressmann 2005] unifies automation and freedom capabilities, and hence contradicts aspects of the above analysis. Its fundamental advance is to add minimal information to geographic feature classes in a GIS database to store representation rules and graphical overrides to individual features.

2.2 Representation and override example

Figure 1 below shows five stages of symbolization for linear road or ‘track’ features from a vector topographic GIS dataset (data copyright swisstopo). Traditional GIS-based mapping systems support only the first two stages.

In stage 1, default GIS symbology (a red line) is assigned to the linear features, and in stage 2 a dashed line symbol is applied, as this is the typical line pattern used in cartographic products for tracks. This GIS symbolization falls short of many cartographic requirements because the poorly symbolized line intersections and bends (highlighted by the red circles), lead to ambiguity as to where the tracks start and end.

Stage 3 shows how the new representation capabilities can automatically produce better symbology at line intersections by adjusting the line dash pattern to ensure intentional (half-dash) connections at the ends of all such line features. Stage 4 shows how using this improved representation as a starting point, the cartographer can further perfect it using a manual override of the sharp bend in the track in the northeast corner, manually marking the corner as needing to be in the center of a dash. This modification is stored in the database as an override to the representation geometry.

Stage 5 shows the ultimate graphical freedom and escape from the rules, where the cartographer has decided to change the color of some of the dashes, and to delete one dash from the other track. However, this ‘Free Representation’ is still closely associated with the original feature.

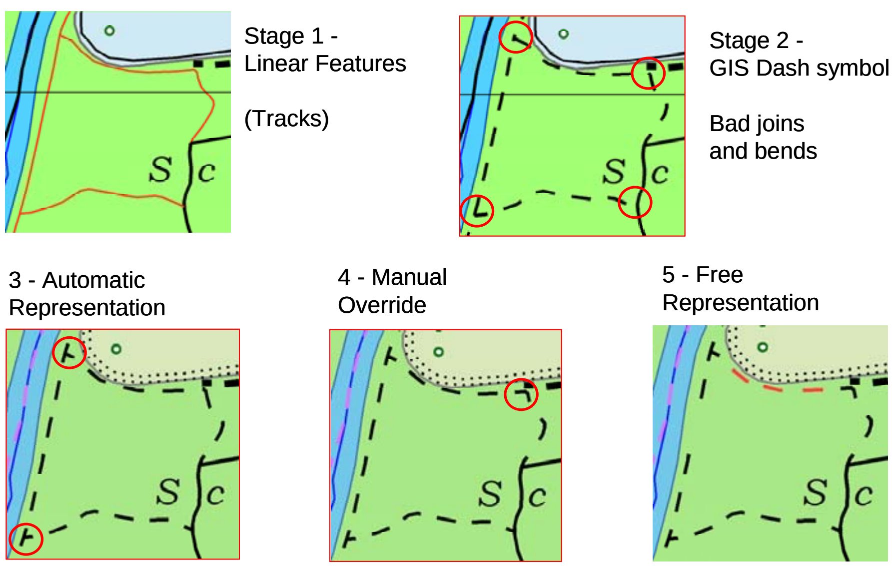

Fig 1. Five stages in symbolization

2.3 Representation Storage

A cartographic representation offers a superset of the capabilities of a GIS layer. One of a representation’s purposes is to encode information on how to categorize and symbolize features. Storing this representation information in the geodatabase contrasts with the setup that is most common in current systems, where layer information is typically stored in binary map document, or in separate layer files.

Physically, a cartographic representation adds additional fields to a standard ArcGIS feature class table. As in any vector GIS, the feature class stores point, line, or polygon geometries, as well as a set of additional attribute fields used for mapping, analysis, and data management. The added representation fields store data that defines the representation rule used to symbolize a feature. They also store cartographic overrides, which are exceptions to the representation rule for a given feature.

A basic representation system design premise was to avoid unnecessarily duplication of data. Therefore, the extra fields that hold cartographic representations and overrides are minimal in size, and wherever possible the representation information is derived dynamically from the existing GIS feature as it is needed. The structures used to hold overrides are highly flexible, and avoid the need for separate fields for individual overrides.

2.4 Representation Rules

Each cartographic representation added to a feature class can refer to different rules for subsets of features within the feature class. For example a roads feature class will typically have different rules for streets, first, second and third class highways, and for freeways. It may also have variant rules for highways on bridges, in tunnels, or for unique circumstances not normally part of the standard data model (such as a highway temporarily interrupted by a fair).

Rules are made up of one or more visual layers, each of which starts from the feature shape geometry and has an optional chain of 'geometric effects' (applied 'on-the-fly') before applying a basic symbol (marker, stroke, or fill).

Figure 2 shows a cartographic representation added to a GIS feature being symbolized. The shape field of the feature has a representation rule applied, which generates two visual layers, the first of which goes through two geometric effects (perhaps an offset to one side, then a dash pattern) before having a basic symbol (stroke) applied. The second layer has one geometric effect applied (perhaps a marker placement pattern) before the basic symbol (marker) is applied. Fig. 3 shows a typical visual result of such a rule.

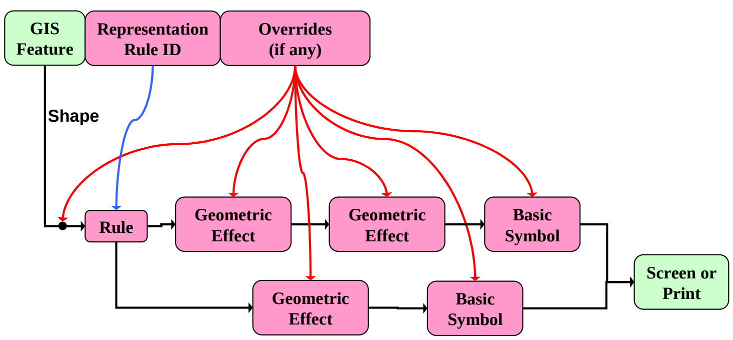

Fig. 2 - Drawing pipeline for representations with overrides.

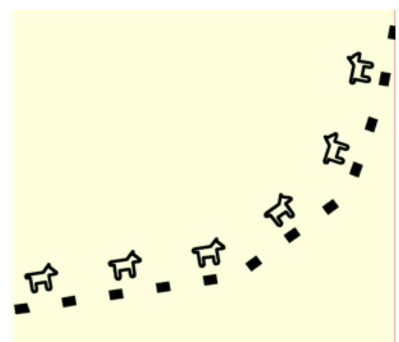

Fig. 3 - Result of Representation Rule.

2.5 Database attribute-driven representations

Rules can also be set up to use any existing field in the database as an 'Explicit Representation Field' to control the feature representation appearance. Such field values can be set by geoprocessing processes, which can use the full power of the GIS toolkit to determine the need and calculate the required result. A typical and powerful example is the use of the topology engine from within a geoprocessing tool to find all the cul-de-sac roads, and set a database field which is then used to control their line end style to be square rather than round ended (Fig. 4).

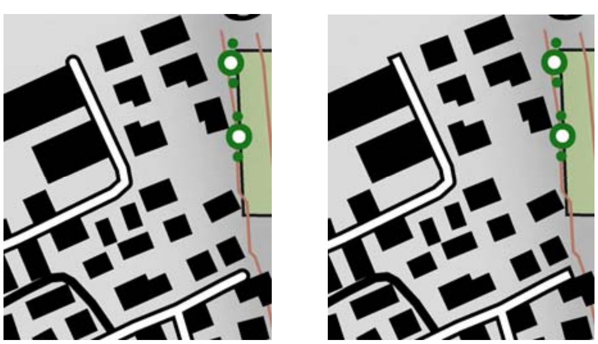

Fig. 4 - Before and after Cul-de-sac processing.

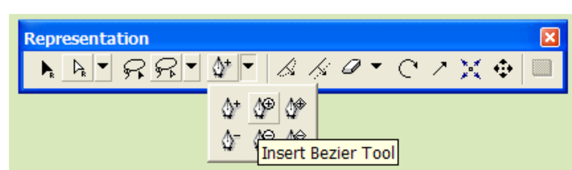

Fig. 5 - Cartographic Editing Tools.

2.6 Overrides and cartographic editing tools

Overrides allow the user to make exceptions to the rule while remaining within the data model. In Figure 2, the override field can modify the input shape, the parameters to the effects, or any of the graphic attributes of the symbols. A set of intuitive geometric and graphic attribute editing tools (Fig 5) are provided for defining and modifying overrides, based on tools and palettes familiar to a user of desktop graphics packages such as Illustrator or Freehand. Furthermore, many of the newly introduced tools are more efficient because they are designed with specific cartographic tasks in mind. Editing of representations takes place within the same versioned editing environment currently supported by ArcGIS for editing vector feature classes.

2.7 Free Representations

A further level of exception is provided by the ability to convert any representation into a 'Free Representation'. This makes an in-line copy of the rules affecting the particular representation, so that the rules can be changed for this one feature. This can include change of geometry type (area to point), adding new rules or symbol layers, or introduction of arbitrary new graphics. Being able to liberate a particular representation from the data model in this way gives freedom to successfully represent features with appearance too rich to model otherwise, such as a railway siding area where the representation should just show a typified subset of lines indicating 'there are lots of railway lines here'. It also allows repositioning or suppression of individual graphic elements of the symbolization, such as individual dashes of a road tunnel to avoid important features at ground level, as in Fig. 6 & 7.

Fig. 6 - Rule-based Representation of Tunnel.

Fig. 7 - Free Representation of Tunnel, with edited dashes.

2.8 Scope of Representations

It is important to note that the representation and override mechanisms are intended as only part of a solution to building and maintaining a multi-scale, multi-product database. Multiple representations of the type described above, work well for deriving and storing multiple products at similar scale from data at an appropriate resolution. As each representation is stored alongside a specific feature in the feature class table, it is not a solution for handling significant scale changes where aggregation of objects is necessary for each product. This task is instead in the scope of generalization, and is described later in the paper.

3 WORKFLOWS

The following sections lay out a series of data flows of increasing complexity, indicating the various ways that the representation mechanisms described in the previous sections can fit into cartographic production workflows.

3.1 Simple case

In the simplest case, an organization has existing GIS data in a feature class, and wants to use it to produce a high-quality cartographic product. In this case, they add a cartographic representation to the feature class to support the single product (see Fig 8).

3.2 Multi-product Case

In the next case, an organization has existing GIS data in a feature class, and wants to use it to produce more than one high-quality cartographic product at similar scales. In this case, they add a cartographic representation for each product to the same feature class (see Fig 9).

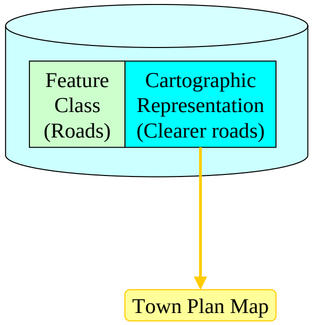

Fig 8. Simple case – Existing Feature Class

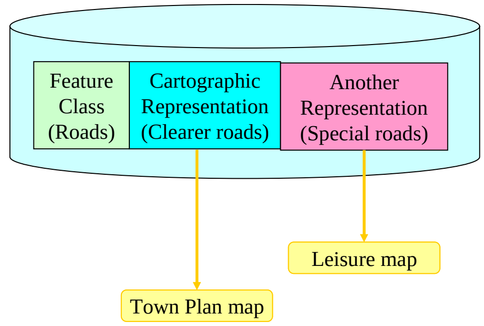

Fig 9. Multiple Cartographic Representations

3.3 DLM/DCM Case

In the third case, an organization has a master landscape model data (DLM), and wants to use it to produce more than one high-quality cartographic product at different scales, as well as non-cartographic products (such as navigation routes for an in-car voice guidance system). In this case, they require the extraction of requisite data from the DLM by selection and generalization into a Digital Cartographic Model (DCM), which can then be enhanced with multiple representation capabilities as before (see Fig 10).

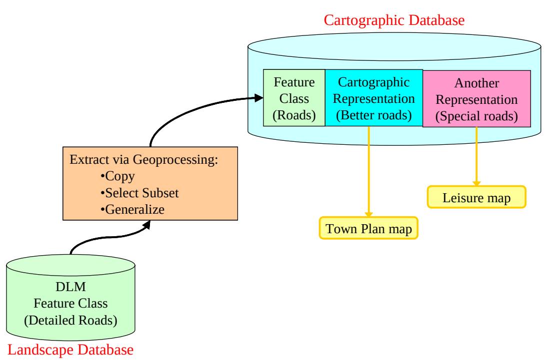

Fig. 10 - DLM/DCM. case

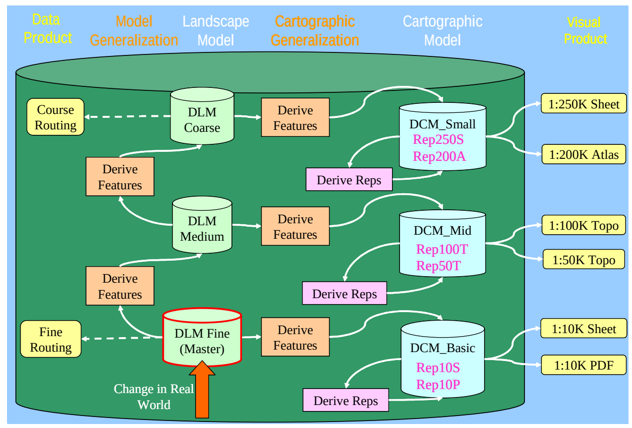

Fig. 11 - Multi-scale DLM/DCM Data Flows.

3.4 Enterprise Case

Figure 11 extends the enterprise DLM/DCM case to where multiple products at different scales are to be produced. Here, the data flow encompasses both ‘model generalization’ (deriving landscape model features at coarser resolution by selection, aggregation and simplification), and ‘cartographic generalization’ (deriving visually appropriate features by applying displacement, exaggeration and typification).

The concepts of DLM-DLM model generalization (statistical generalization) and of DLM-DCM cartographic generalization were laid out in [Brassel & Weibel 1988], and conceptual models for generalization are discussed further in [McMaster & Shea 1992 pp21-26]. Normally, model generalization is applied first, in order to derive a set of reduced-resolution landscape models. Starting from each DLM, cartographic generalization is applied, to produce a digital cartographic model for a particular ‘scale band’.

Once cartographic data appropriate to the scale band has been derived, the multiple representation and override capabilities can then be applied to handle the geometric differences and to satisfy the symbolization and cartographic clarity requirements for various products.

3.5 Update Data Flows

Once the DLM/DCM work flow has been established, and the various products generated in their initial editions, then the data flows for update should follow the same paths, starting with capture into the master DLM, flowing through the reduced resolution DLMs, and through the cartographic model and the representations to the products.

For maximum efficiency, just the delta changes should be propagated through the framework, which requires incremental generalization – a concept which cannot be fully implemented until a second stage of development, learning from experience of initial generalization. A prerequisite for incremental generalization is the creation and maintenance of ‘parentage’ attribution for all derived features.

In the interim, the practical workflows will assess the accumulated amount of change, and determine for each product whether it is better to reapply generalization starting from the current state of the DLM, or to apply minor edits at the DCM stage. In either case, the versioning capabilities of the geodatabase will be used to control the time dependencies often required for cartographic publishing.

Note that rule-based representations (other than the very small minority where overrides have been made) automatically apply any changes made to the source feature. This minimizes the effort in propagating change through to products.

4 GEOPROCESSING AND GENERALIZATION

4.1 Geoprocessing

ArcGIS [ESRI 2004] has a versatile and powerful framework and toolbox for 'Geoprocessing' – applying bulk processes to geographic data. Geoprocessing tools (Fig 12.) are grouped for availability into toolboxes, and can be simply combined into sequences, either by scripting languages such as Python, or using a visual ModelBuilder. The system is extensible, and user-written tools can easily be added as scripts or executables. Tools (whether core or user-written) can be invoked via dialogs, scripts, models, or by command lines.

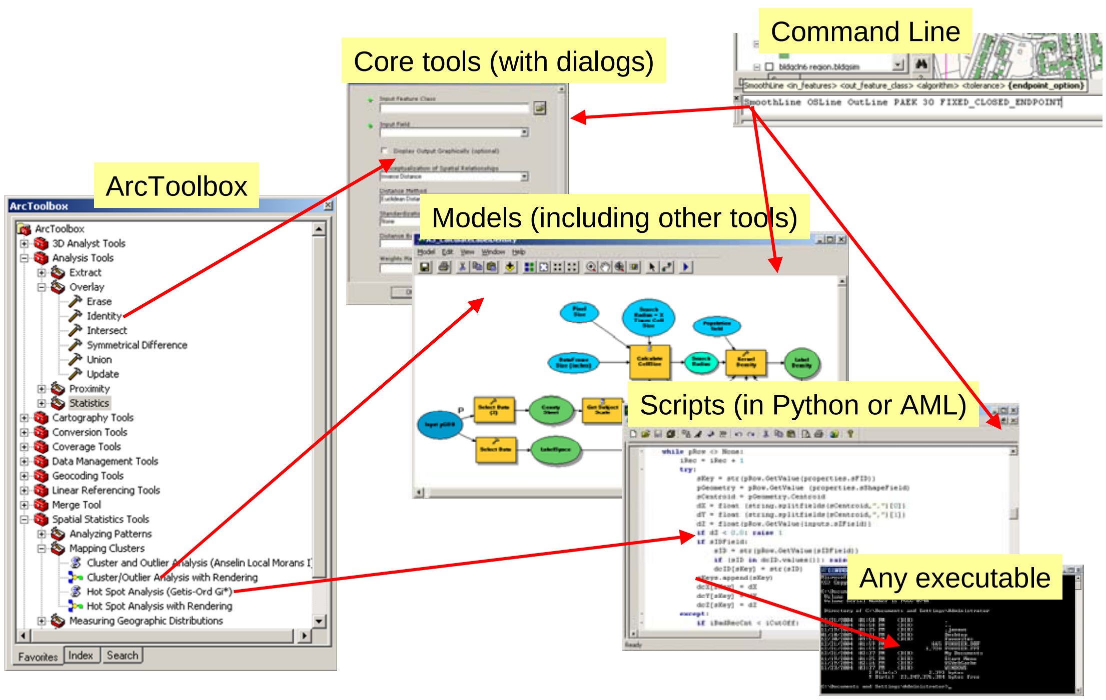

Fig 12. Geoprocessing framework, models and tools

The substantial investment in development of the geoprocessing framework which became available in ArcGIS 9.0 provided the primary infrastructure on which an efficient and flexible generalization environment could be based. In the same recent timescale, the underlying ArcObjects components acquired the necessary level of functionality and richness of data structure to underpin efficient and robust generalization algorithms.

4.2 Generalization algorithms

Within the geoprocessing framework, there are a growing set of generalization tools, and more are being researched and developed for subsequent releases of the software. Figure 13 shows results from a variety of these algorithms.

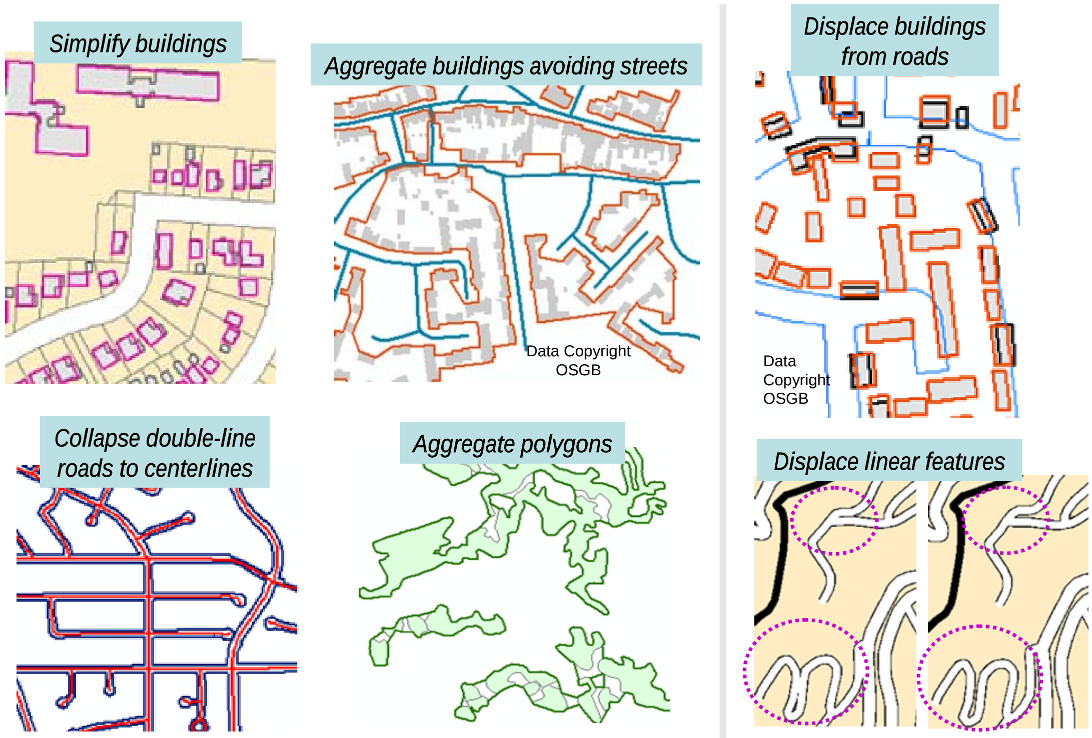

Fig 13. Generalization Algorithms

Recent engineering work has concentrated on providing geoprocessing tools that correspond to improved versions of the generalization commands available in Workstation ArcInfo - Simplify Polygon, Simplify Building, Aggregate Polygon, and Collapse to Centerline. These will join the existing tools in the Generalization toolset (under Data Management Tools), which include Dissolve, Eliminate, Simplify Line, and Smooth Line.

In all cases, the new tools will be more robust and generic than the old coverage tools, as they make use of the better data structures and libraries available in ArcGIS, such as TINs and topology. Many other tools can be useful for generalization when used in process sequences. Examples include tools in the Analysis toolbox for data extraction, overlay, and proximity computations, tools in Spatial Statistics toolbox for analyzing patterns, clusters, and distributions, and tools in Features toolbox and Feature Class toolbox for converting, merging, and managing features.

4.3 Contextual Generalization

Good generalization is not an easy task, whether done by a computer or a human, or by a combination as described as “Amplified Intelligence” in [Weibel 1991]. It requires understanding and characterizing the geography of the mapped area, and involves finding patterns in the data, and abstracting them. In particular:

- One can’t do it a feature at a time – It involves relationships with neighbors
- One can’t do it a layer at a time - Relationships are with features of many classes
- One can’t do uniformly - Even features of one class have different surroundings
- One can’t do it across whole map at once - Must localize or partition
- One can’t do it just on geometry – It needs understanding of topology and of attributes

An earlier paper [Lee 2004] analyzed and illustrated in detail what geographic and cartographic contexts should be considered in generalization. The multiple representation and generalization project is moving on to extend the framework to include contextual generalization. The project team will gain from the knowledge of other ESRI development team members who participated in the European AGENT project [Lamy et al 1999], and in other leading cartographic functionality, such as the Maplex text placement system.

The resultant design and architecture will exercise the concepts of partitioning mapped areas into geographic zones, recognizing patterns and distributions, setting rules and priorities that guide the generalization analysis and decisions, controlling and assessing the generalization status and quality, and supporting post processes and representation refinement.

The design will learn where appropriate from the AGENT experience. However, the concept of individual features becoming self-aware software agents with a rigid lifecycle is unlikely to be applied, as experience by the primary author during and after the AGENT project indicates that this introduces unnecessary complexity and computational overheads for the achieved returns.

Instead, the resultant system will make extensive use of the geoprocessing framework, and will flow data through an adaptive series of steps, such as those outlined in figure 14.

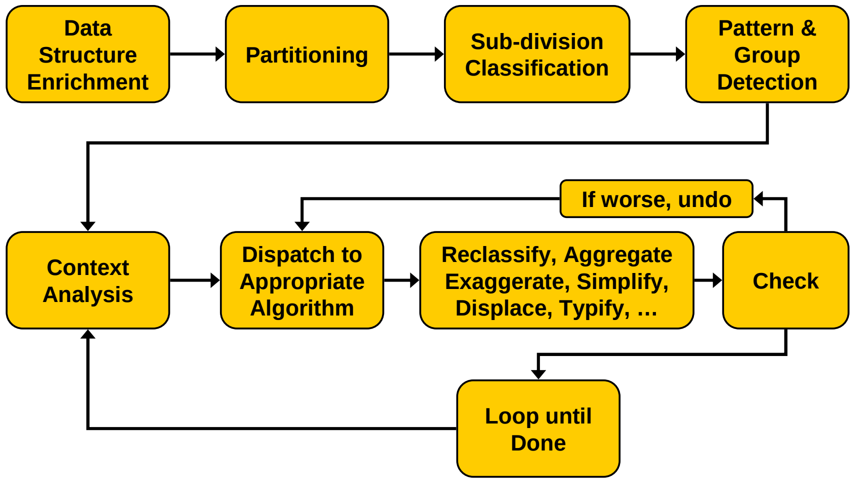

Fig 14. Possible sequencing for contextual generalization

- Data structure enrichment involves making explicit many of the spatial relationships that are present implicitly in the data. Generalization is a computationally intensive task, and benefits from rapid access to nearest neighbor, topological connectivity and other such spatial relationships.
- Partitioning involves identifying homogenous geographic regions of the map, such as separating urban from rural areas, and flat areas from hilly areas.
- Sub-divisions are relatively enclosed areas that can be generalized in isolation, such as urban blocks (keep turning right round the street network).
- Patterns are important to generalization – this stage can identify alignments of trees, clover-leaf road intersections, or configuration of structures, such as L, H, T or C-shaped sets of farm buildings.
- Context analysis looks at what is within each spatial division, and dispatches the individual features or groups of features to the appropriate algorithm
- Algorithms modify features or create new features based on feature combinations
- Check, loop and undo stages act as sanity checks, and allow the exploration of alternative approaches.

The above stages are example of the type of steps that can be implemented within a generalization geoprocessing model – future papers will provide more detail as the design and development continues.

4.4 Model Generalization and Cartographic Generalization

Although model generalization emphasizes data integrity and geographic information, and cartographic generalization focuses on symbolization and legibility in map space, they share similar analysis and management, and will be built in a common environment. The system will apply specifications and alternatives according to the needs of the model generalization or cartographic generalization being performed.

For example, during model generalization, the operations of reclassification, simplification, and aggregation are often needed and the one-to-one or many-to-one relationships and feature attribution must be maintained. In contrast, during cartographic generalization, the operations of exaggeration, displacement, and refinement considering full symbolization are more essential to the final products. The database cartography framework will incorporate both model and cartographic generalization processing.

4.5 Constraints and Rules

Early aspects of the architecture include designing a generic system for defining and applying, where appropriate, the common rules and tolerances which control generalization behavior. This includes rules for things like:

- minimum sizes for classes of feature
- minimum spacing between different types of feature
- whether one class of feature can overlap another
- Priorities for aggregation and displacement

It is not intended to create a single repository for all the constraints on generalization, as many of them are better handled as parameters to the individual generalization processes, within the overall models that will control the process sequencing. The design of the rules mechanism will learn from the prior experience of the Maplex text placement engine, which has similar requirements for controlling contention for white space on the map.

Other inputs into the design covering constraints and rules will include the experience from the AGENT project, plus appropriate academic studies, such as [Beard 1991].

5 CONCLUSIONS

- New capabilities for multiple representations and overrides within a commodity GIS and relational database are nearing completion and will provide a powerful and flexible environment for cartographic publishing.
- This visualization environment forms part of a larger system of data models and data flows, to cope with the concepts of DLM and DCM, and the generalization processing that links them.
- Generalization functionality is increasing, and design and development is under way on a framework and tools for contextual generalization.

6 NOTES

1. The sample data used in Figures 1, 4, 6, & 7 is swisstopo VECTOR25, copyright Swiss Federal Office of Topography, and urban data for Figure 13 is courtesy Ordnance Survey GB, Crown Copyright.

2. Some contextual matter in this paper is revised and updated from other documents jointly authored by members of the ESRI team working on representations and DLM/DCM flowlines. In particular Cory Eicher, Marc-Olivier Briat, Thierry Kressmann and Edie Punt are thanked for their contributions.

3. This paper is a forward-looking research document, and the capabilities it describes are still under development. As such, it is intended to give guidance as to likely future direction and should not be interpreted as a commitment by ESRI to provide precise capabilities in specific releases.

7 REFERENCES

- Beard K. 1991, “Constraints on Rule Formation” In: Buttenfield, B.P. and McMaster, R.B. (eds.). Map Generalization: Making Rules for Knowledge Representation. London: Longman, 121-135.
- Beconyte G, 2004, “Conceptual Models for Cartographic Representation”, Geovizualization 2004, Kolymbari, Crete http://www.kc.gf.vu.lt/Publications/ConcptModelling2004.doc
- Brassel, K.E. and Weibel, R. 1988, „A Review and Framework of Automated Map Generalization” Int. Journal of Geographical Information Systems, 2(3): 229-244.
- ESRI 2004, “What is ArcGIS”, ESRI Press, ISBN 1-58948-090-2, 2004, http://store.esri.com/esri/showdetl.cfm?SID=2&Product_ID=109&Category_ID=28
- Fairbairn D, G. Andrienko, N. Andrienko, G. Buziek, J. Dykes, 2001, “Representation and its relationship with cartographic visualization: a research agenda” - Cartography and GIS, Vol. 28, Nr. 1
- Hardy P. 2004, “Database-Driven Cartography From A Digital Landscape Model, With Multiple Representations And Human Overrides”, ICA Workshop on Generalisation and Multiple Representation, Leicester, UK, August 2004 - http://ica.ign.fr/Leicester/paper/Hardy-v2-ICAWorkshop.pdf
- Hardy, Eicher, Briat, Kressmann 2005, “Database-stored Representations and Overrides, Supporting Automated Cartography with Human Creativity”, Auto-Carto2005, Las Vegas, USA, March 2005.
- Lamy et al 1999, “AGENT Project: Automated Generalisation New Technology”, 5th EC-GIS Workshop, Stresa, Italy, June 1999. http://agent.ign.fr/public/stresa.pdf
- Lawrence V 2004, “The changing role of national mapping organisations: A case study of Ordnance Survey”, Map India, http://www.gisdevelopment.net/policy/international/mi04003pf.htm
- Lee D. 2004, “Geographic and Cartographic Contexts in Generalization”, ICA Workshop on Generalisation and Multiple Representation, Leicester, UK, August 2004 - http://ica.ign.fr/Leicester/paper/Lee-v2-ICAWorkshop.pdf

- MacEachren A, 1995, “How Maps Work. Representation, Visualization and Design”. New York-London: The Guilford Press.
- McMaster, R. B. and K.S. Shea, 1992. “Generalization in digital cartography”, Washington, D.C., U.S.: Association of American Geographers.
- Morehouse, S. 1995 “GIS-based map compilation and generalization." GIS and Generalization: Methodology and Practice. Eds. J.C. Muller, J.P. Lagrange, R. Weibel. Bristol: Taylor & Francis, 21-30.
- Weibel R. 1991, “Amplified Intelligence and Knowledge-Based Systems” In: Buttenfield, B.P. and McMaster, R.B. (eds.). Map Generalization: Making Rules for Knowledge Representation. London: Longman, 172-186.

[Original 2004-02-23, Revised 2005-06-07]