Design and Experience of Generalization Tools

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

Abstract

While research is underway at ESRI on solutions for adaptive and contextual generalization, the development of bulk generalization tools under the geoprocessing framework in ArcGIS has continued. One of the essential design aspects is to extend and redefine the scope of automation. Another is how to support the evaluation and further optimization of the output.

It is obviously important to use the most effective approaches and techniques to maximize the automation, so the new tools take advantage of the topology functions and TIN functions available in ArcGIS to preserve shared geometry and to derive generalized features. It is equally important that the tools provide feedback about the quality of the automated output, together with hints and tips to support further processing to complete the generalization tasks and hence increase productivity.

This paper introduces the existing and upcoming generalization tools, discusses major design decisions, and illustrates how these tools, along with other geoprocessing tools, can be used in various scenarios for model generalization and for cartographic generalization. It covers how the data can be generalized, but also the inspection and follow-up processes.

1 INTRODUCTION TO GEOPROCESSING

ArcToolbox is the geoprocessing framework of ArcGIS. It is the environment where batch processes are set up and executed to manipulate spatial data. Each geoprocessing tool takes a required input stream (e.g. a feature class or a selection of features) together with any controlling parameters, and it produces output (such as a new feature class). The Geoprocessing framework combined with cartographic representation facilities form the necessary infrastructure for automated multi-product cartography (Hardy & Lee, 2005).

1.1 Generalization tools

Generalization is the abstraction, reduction and simplification of features, for change of scale or resolution. A set of automated generalization tools has been implemented in the ESRI ArcGIS software family and centralized into the ArcToolbox geoprocessing framework (Lee, 2003). The Arc generalization commands, previously available in Workstation ArcInfo, have been integrated into the Generalization toolset under the Coverage toolbox – Data Management toolset, still operating on topologically structured coverage data. A new Generalization toolset has been added under the Data Management toolbox containing tools operating on the new generation of geographic data - geodatabase features.

The first phase of growing the new Generalization toolset aims at developing tools that provide similar functionality to the coverage generalization commands, with equivalent or better performance and quality. This toolset in ArcGIS 9.1 includes four tools: Dissolve, Eliminate, Simplify Line, and Smooth Line. Four more tools have been developed for the ArcGIS 9.2 release – they are: Aggregate Polygons, Collapse Dual Lines To Centerline, Simplify Building, and Simplify Polygon.

1.2 Data structures for geoprocessing and generalization

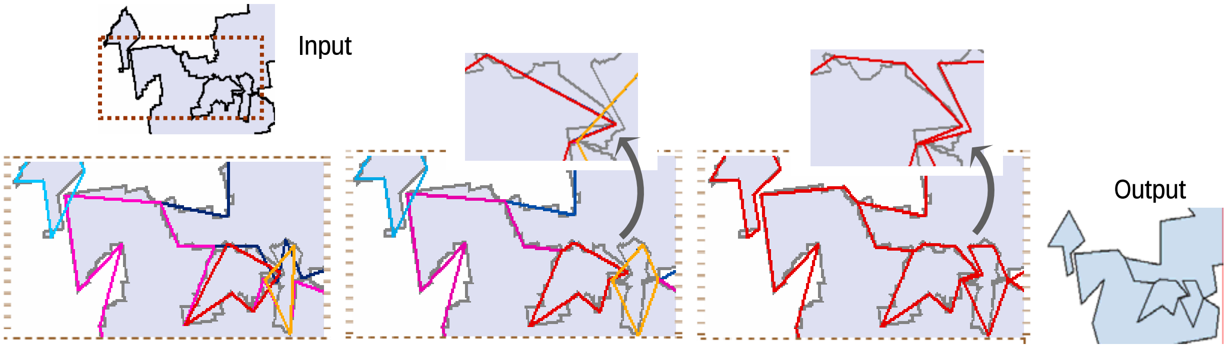
There are opposing pressures on GIS implementations for simplicity and complexity of data structures (Hoel et al, 2003). Increasing standardization of use of commodity relational databases for persistent storage pushes the GIS towards simple tabular data structures allowing rapid query and retrieval. The geodatabase data model of ArcGIS is a move away from the complex persisted topology structures of earlier GIS implementations. However, generalization and many other spatial analysis processes rely on frequent checks on the neighboring features and topological relationships. For example, one of the popular requirements for generalization is to preserve shared geometry – common boundaries and connectivity; and to do such tasks at speed requires the building of temporary topology which makes explicit those spatial relationships.

To achieve good performance and high quality output, modern data structure traversal techniques have been incorporated in the new generalization tools. An in-memory Triangular Irregular Network (TIN) is built on the fly to support the various analyses, such as proximity and adjacency.

2 TOPOLOGY IN GENERALIZATION

The design of generalization tools needs to satisfy different levels of applications – some demand fast reduction of data without concerning topology; others require to maintain feature integrity for further use and analysis. Two of the key objectives in maintaining feature integrity during generalization processes are to analyze and preserve the topological relationships embedded in the input features and to discover any violations of topological rules introduced by the processes.

The Simplify Line and Simplify Polygon tools currently allow three options for increasing levels of topology analysis and handling (Figure 1):



NO_CHECK PRESERVE_SHARED RESOLVE_ERRORS

Figure 1. Simplify Polygon with three options

- NO_CHECK - the features will be simplified using the specified algorithm and tolerance; no checking on topological relationships or violations.
- PRESERVE_SHARED - the embedded topological relationships in the input features will be analyzed and preserved, particularly the knowledge that two or more features share a common segment. Typical violations of topology rules, such as features changed by the process to be intersecting or overlapping will be flagged.
- RESOLVE_ERRORS - the embedded topological relationships in the input features will be analyzed and preserved; typical violations of topology rules, such as features changed by the process to be intersecting or overlapping will be detected and fixed.

2.1 Preserving embedded topological relationships

2.1.1 Connectivity and adjacency

Geodatabase features, whether lines or polygons, are stored as completely independent geometries, rather than as topological segments as in the previous Arc/Info Coverage format. Hence they may not have explicit vertices indicating the connectivity (where they are touching or intersecting) and the adjacency (where a shared portion of a line or polygon boundary begins and ends). In order to preserve the implicit topology, the design of the Simplify Line and Simplify Polygon tools involves the use of a standard ArcGIS topology function (Feature To Line). This segments feature geometry at these locations so that the simplification will be applied only to the segments without altering the endpoints. The segments that form each original input feature are easily identified and the entire feature is reconstructed after the simplification.

2.1.2 Grouping

Another kind of topology analysis is needed when a generalization process is to be performed on a group of adjacent features.

> **Example one**: both Simplify Polygon and Simplify Building (footprint) tools allow a minimum_area parameter so that an isolated feature or a group of adjacent features with the total area smaller than the minimum_area will be excluded from simplification and output. This is not an option in the standard selection by area and elimination.
>
> **Example two**: the Simplify Building tool will process a group of buildings connected in relatively simple formation as an entity. This involves finding the outer boundary of the connected building group, simplifying the shape of the outer boundary, and then restoring the interior boundaries (walls).

In both cases, the adjacent features and the groups they belong to are discovered on the fly to support the desired simplification processes.

2.2 Handling new spatial conflicts

Although the input features may contain intersecting lines and overlapping polygons, any of the simplification algorithms may introduce new topological errors – lines collapsed on top of each other or across one another. They are considered as spatial conflicts, and therefore undesired. These errors must be detected separately from any existing topological relationships in the input features. The Simplify Line and Simplify Polygon tools segment and analyze the line work both before and after the simplification, so any new coincident or crossing lines can be detected.

2.2.1 Flagging the involved features

All geodatabase features carry unique feature IDs and these are preserved during the segmentation process. They can therefore be used after detecting introduced errors to sort out which features are in conflict and assign a flag value to their output records. Flagging the introduced conflicts is an option in all three simplification (line, polygon, and building) tools and also in the Smooth Line tool. Further enhancement is under consideration to store the flags on the local segments instead of the entire features for easier visualization and post-processes.

2.2.2 Resolving the errors using a progressive approach

Resolving the introduced errors has been made available for Simplify Line and Simplify Polygon. This is an enhancement from the Workstation ArcInfo command. The input features are first simplified using the specified tolerance. The error detection routine then locates the involved segments of features. A reduced tolerance (half of the original) is applied to re-simplify these segments. This detection and re-simplification with a reduced tolerance (half of the last used) repeats until no more errors are found. More details about this approach were discussed and illustrated in a previous paper (Lee, 2004).

3 TIN-BASED GENERALIZATION

There have been many research papers and implementations on TIN-based generalization, including feature aggregation, skeleton approximation, displacement, and so on (Jones, Bundy, and Ware, 1995; Peng, 1997; Ware and Jones, 1998), although other techniques have been examined in similar areas (Christensen, 1996; Hojholt, 2000; Thom, 2005). TIN-based generalization has been determined at ESRI to be one of the most effective approaches to use. An efficient library of TIN functions has been made accessible in ArcObjects (the development platform for ArcGIS). The TIN-based approaches have been utilized in developing Aggregate Polygons and Collapse Dual Lines To Centerline tools, as described below.

3.1 Aggregation of polygon features

The new Aggregate Polygons tool is designed to combine area features within a specified distance to each other into one area feature. A TIN structure is built using the input polygons. Triangles covering the polygon areas are given tag values equal to the polygon IDs. Other triangles covering the space between features will have the tag value of 0. Through the triangle edges it is quite easy to analyze where features are close enough to belong to a cluster (a group). The tool traces the edges bordering a group of features and the connecting edges to form a new polygon and, if necessary, any polygon hole(s).

The tool works for two types of features - non_orthogonal features which have natural-shaped boundaries, such as vegetation; orthogonal features which have mostly square corners in their shapes, such as buildings (Figure 2).

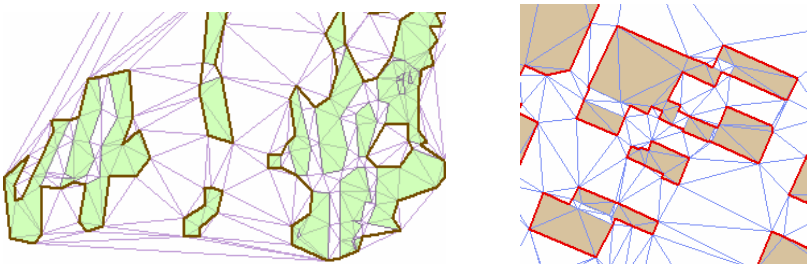


Figure 2. Aggregation of non_orthogonal (left) and orthogonal (right) features

3.2 Derivation of road centerlines

The Collapse Dual Lines To Centerline tool is aimed at large-scale road casing lines (not polygons) as input and derives centerlines following the “middle axis” of pairs of casings that fit the specified range of widths. A TIN structure is built using the input lines. Any closed street blocks are identified using the Feature To Polygon function which assembles polygons from lines. They will then be excluded from centerline derivation by locating a seed triangle inside each polygon area in the TIN and populating the triangles enclosed by the polygons with a special tag value (Figure 3). Open areas (surrounded by incomplete street blocks or skirting the data areas) are also identified and excluded (Figure 3). The remaining spaces are where to derive centerlines.

Refinement of the tool implementation is still in progress. The general strategy is to first identify road junctions by “junction triangles” (a junction triangle touches three road casings; none of its three edges is part of a road casing). These initially identified junctions will be further analyzed – they may or may not be merged into larger junctions with three or more road entries; some may be reduced to pseudo junctions (each with two road entries) or dead-ends (each with only one road entry). Centerlines are derived wherever the road width is within the specified range and are connected at junctions wherever possible (Figure 4). Some junctions may remain unconnected due to the complexity of their configuration (see section 4.2.2).

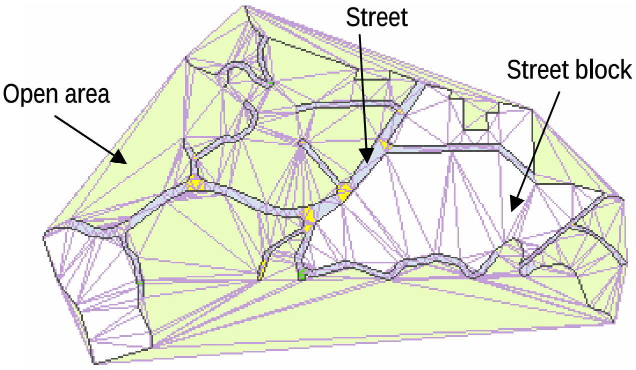


Figure 3. Street blocks and open areas are excluded

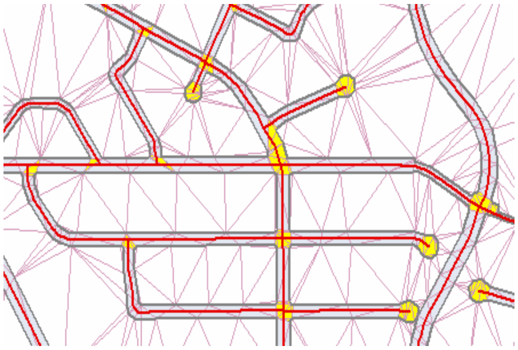


Figure 4. Centerlines and junctions

4 QUALITY AND STATUS INFORMATION

The design of generalization tools not only addresses how the automation is done, but also involves decisions on what may not be fully solved automatically and how such facts are communicated to the user. Examples will be given below on the quality and status information provided with the output which can be examined and used as feedback to further processes.

4.1 Tracking feature conflicts

Generalization processes may introduce feature conflicts due to the change of geometry. It has been shown to be helpful to provide a tool option to flag features involved in the conflicts. When such an option is enabled, an attribute field will be added to the output table to carry a value of 0 for non-conflict features and 1 for conflict features. The following tools have this option: Simplify Line, Simplify Polygon, Simplify Building, and Smooth Line.

A high percentage of features flagged could hint that:

- the tolerance may be too large (consider a smaller tolerance)
- too many features are in close proximity (consider elimination of small ones or aggregation)

The flagged features can be selected by a query, and passed to further inspections and processes. Where the shapes of the flagged features are relatively simple, the conflict segments can be easily located visually; otherwise, the selected features can be checked against topology rules and the conflict areas highlighted.

4.2 Categorizing the generalized features and recording status

4.2.1 Status values in Simplify Building output

Buildings are analyzed into three categories by the Simplify Building tool: isolated single buildings, group of connected buildings with simple configuration, and group of connected buildings with complex configuration. Each category is processed differently.

Isolated buildings are simplified individually through a number of pattern recognition algorithms (Wang and Lee, 2000). Buildings of the second category are simplified by group (simplifying the outer boundary of a group and then restoring the interior boundaries). Buildings of the third category are identified by group; they are not simplified but flagged for human inspection.

A unique case of the isolated buildings is that when the tolerance is relatively big for a small building but it is to be kept according to a given minimum_area, the small building will be transformed to a rectangle (oriented to an approximate main axis, keeping the same area and ratio of its “envelope” dimension) instead of going through the pattern recognition process.

The output of Simplify Building contains an attribute field storing the above status values for each simplified building (Figure 5); decisions on further optimization can be made based on these feedback.

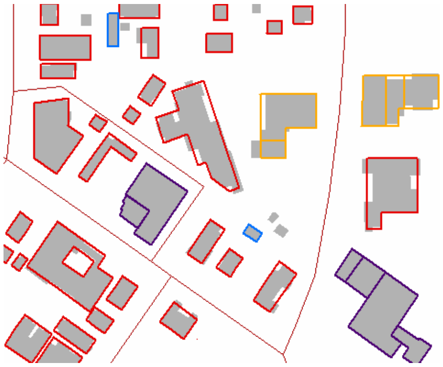


Figure 5. Status of simplified buildings (red – isolated building, orange – simplified group, dark blue – not simplified group, light blue – small but kept in rectangle shape)

4.2.2 Line types of Centerline output

Similarly, the output of the Collapse Dual Lines To Centerline tool contains a field which currently stores three line type values: 1 for derived centerlines, 2 for outlines around unresolved junctions or roads beyond the specified range of width, and 3 for connecting lines between junctions without clear left and right casing IDs (Figure 6).

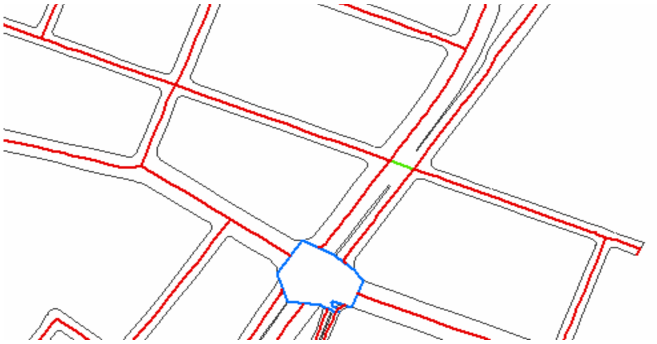


Figure 6. Line types of centerline output (red – centerline, blue – unresolved junction, green – connecting line with unclear left and right casing IDs)

4.3 Linking output features with input features

Another design aspect is to maintain the link between input and out features for necessary attribute transfer and possible reprocess of certain features. The simplification tools involve one-to-one feature relationships, therefore the input feature ID is carried over in a field in the output wherever applies. The derived centerlines carry the left and right casing IDs. The aggregated output includes a relation table to store the one-to-many relationships.

4.4 Process metadata

One further important form of status information is maintained as database metadata, logging all processing steps. When any geoprocess is applied to a geodatabase feature class, a record is added to the database metadata, giving information as to date, time, tool used, and tool success or failure.

5 GENERALIZATION SCENARIOS

The scenarios used in this paper represent typical generalization processes including data selection, classification, partitioning, reducing the level of detail using the generalization tools, analyzing the output, and suggesting further automatic or possibly interactive processes. The design of the generalization tools to include feedback information and the experience of assessing the results and investigating the follow-up processes will provide helpful input to the ongoing design and implementation of a comprehensive, optimized generalization solution.

5.1 Maintaining connectivity in generalization of hydrographic features

It is quite common that small streams are collected as linear features, wider rivers as double-lines, and lakes as polygons. Since their connections to each other are not stored explicitly in the geodatabase, simplifying each feature class separately would result in the loss of connectivity. The following process solves this problem.

First the lake polygons are converted into lines as the Simplify Line and Simplify Polygon tools don’t take mixed geometry types as input; and polygon label points are computed for later use in restoring polygons. Then the streams, rivers, and lake shorelines are merged into one feature class with a common attribute that carries their classifications. The merged feature class is then simplified with an initial tolerance and the PRESERVE_SHARED option; some line crossings are flagged. Provided that the number of flagged features is not large, indicating the tolerance used is appropriate, then the final simplification is done with RESOLVE_ERRORS option. The simplified lake shorelines are then selected along with the label points as input to Feature To

Polygon tool to restore the lake polygons. The end results are a set of simplified streams, rivers, and lake polygons with the correct connectivity (Figure 7).

Although the process is aimed at feature simplification, line crossings occurred in the dual-line river segments and were flagged. This may suggest that the dual-line rivers could better be collapsed and be represented by centerlines.

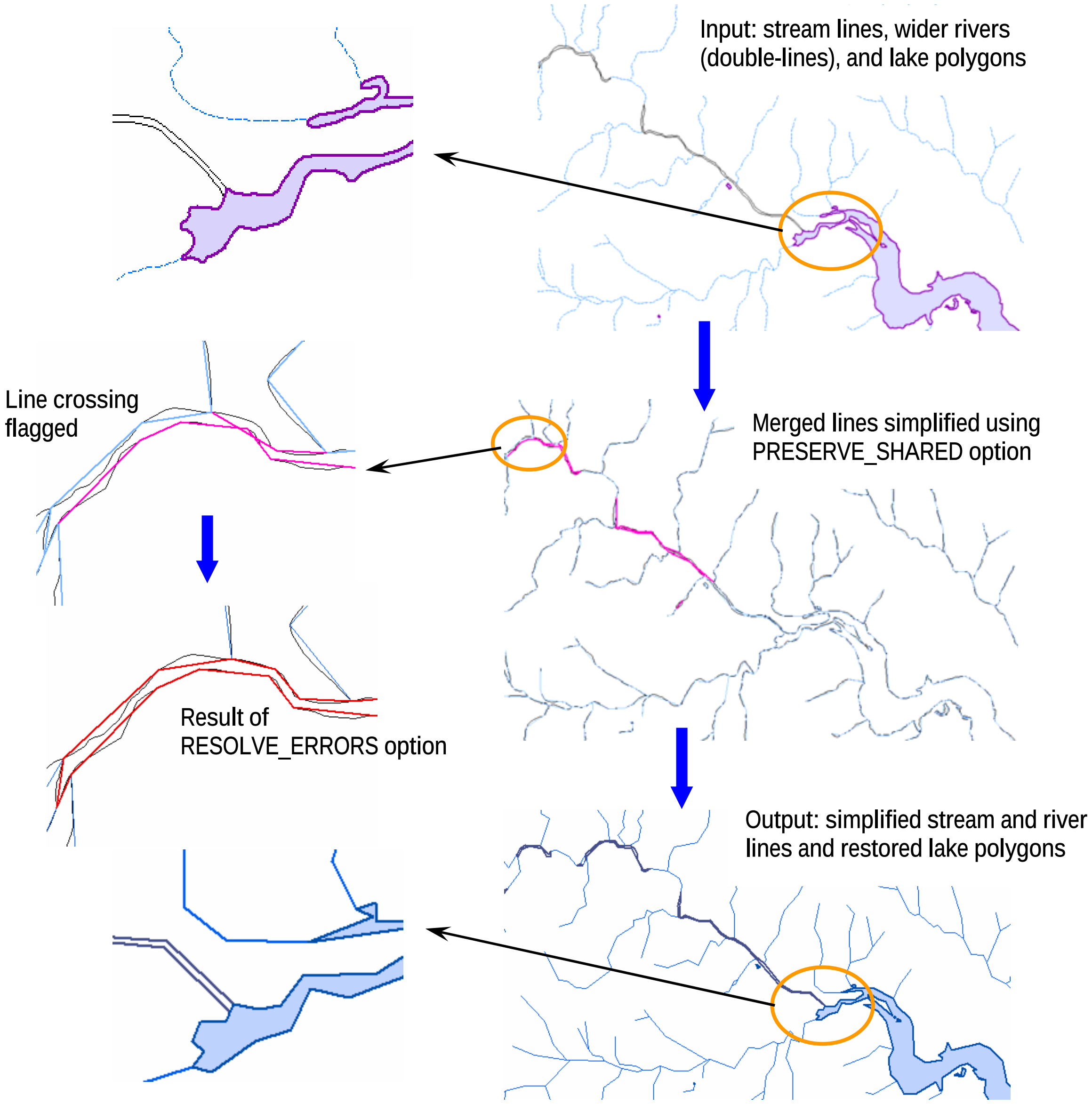


Figure 7. Simplification of multiple feature classes

5.2 Partition and generalization

One of the well-known problems in generalization is to identify the constraining spaces or partitions within which features share similar characteristics and are to be processed and balanced as a whole. In the following example, the street blocks are used as partitions to guide building generalization. The input buildings and road centerlines are collected at the scale of 1:5000. For a target scale of 1:25000, the buildings need to be generalized by these rules:

- Very small buildings will be eliminated
- Buildings close to each other will be aggregated
- The aggregated buildings and remaining buildings will be simplified
- The aggregation and any conflict resolution must be done within street blocks

Here are the steps to achieve the above requirements:

- Street centerlines are made into partition polygons by Feature To Polygon tool
- Buildings larger than the required minimum size are selected and then overlaid with the partition polygons using the Intersect tool so that buildings in a particular street block receive that polygon ID (Figure 8).
- Buildings within a street block are selected by its partition polygon ID; they are then
    - aggregated where within a specified distance to each other;
    - and simplified to satisfy a tolerance.

The conflicts in the end result shown in Figure 8 are flagged, which hints the need for displacement or adjustment to the parameters and steps in the above processes.

1:5k data displayed at 1:25k. Buildings are grouped by street block partitions.

Selected by a partition ID (small buildings already eliminated); then aggregated.

Simplified with conflicts flagged.

Figure 8. Partition and building generalization

6 FUTURE DIRECTIONS

While much useful generalization can be achieved by sequences of self-contained processes such as those described above, we realize that much more algorithmic understanding of spatial context and balancing of conflicting requirements is necessary if we are to reach the quality and scope of generalization that a skilled human cartographer can achieve. To that end, a research and development project is under way to provide a mechanism, tools and models for contextual optimization, aimed at the harder aspects of generalization, such as displacement or typification. The following section overviews the proposed mechanism, which will be described more fully in a future paper.

6.1 Optimizer, Constraints, Actions, and Exclusions

At the heart of the system is the ‘Optimizer’, which iterates through cycles, assessing the overall satisfaction of the chosen data as determined by the sum of the individual satisfaction ratings of a set of user-supplied ‘Constraints’. Each constraint has an associated ‘Action’ which should improve satisfaction. The optimizer also has the concept of ‘Prohibitions’ (‘strict constraints’), which have no associated action, but act to prohibit invalid states of the data.

The optimizer chooses features with low satisfaction and applies actions to try and improve the overall system satisfaction. The optimizer and its set of constraints, actions and exclusions are all implemented in the standard geoprocessing framework of ArcGIS, so that geoprocess models can easily be built to combine contextual optimization stages with traditional bulk generalization processes.

6.2 Optimization techniques

Within the optimizer, a simulated annealing technique is applied, with an exponentially declining notional ‘temperature’, allowing temporary falls in overall satisfaction early in the optimization process in order to prevent the system getting stuck in local minima. An important design factor is to apply spatial intelligence in the ‘action’ routines to quickly iterate to a better solution, rather than relying on inefficient random movements.

7 CONCLUSIONS

The design and implementation of a set of generalization tools has been described, operating in the geoprocessing framework of a commodity GIS. Building more generalization tools and enhancing existing tools will continue to be challenging development tasks at ESRI.

The design of production ready generalization tools has to focus on data integrity, automation quality, and productivity through the use of effective techniques and technologies, including the topology and TIN engines. Increased feedback information from the tools can help steer the development of processing strategies.

Critical further research is continuing in the area of deriving appropriate partitions based on analysis of cultural, natural, and computational factors for generalization processes. More logical sequences, parameters, control options, and evaluation methods will be investigated to increase the versatility of the generalization tools and to support map production needs.

Contextual generalization is the next step towards turning human cartographer’s experience into a digital solution. It requires a powerful system that takes into account multiple factors and prioritized rules, makes decisions based on analytical measurements and comparisons, and takes actions accordingly. The Optimizer prototype has been a starting point to explore the mechanism of such system and has produced encouraging results. As new generalization tools and functions become available and are integrated into the optimization process, the way is open for a comprehensive generalization solution.

References

Christensen, Albert H.J., 1996, “Street centerlines by a fully automated medial-axis transformation”, Proceedings of GIS/LIS96 Conference, Denver, CO, p.107-115.

Hardy, Paul and Lee, Dan, 2005, “GIS-Based Generalization and Multiple Representation of Spatial Data” Proceedings of CODATA International Symposium on Generalization of Information, Berlin.

Hoel, Erik; Menon, Sudhakar; and Morehouse, Scott, 2003, “Building a Robust Relational Implementation of Topology”, Proceedings of the 8th International Symposium on Spatial and Temporal Databases (SSTD 2003), Santorini Island, Greece, p.508-524.

Hojholt, Peter, 2000, “Solving Space Conflicts in Map Generalization: Using a Finite Element Method”, Cartography and Geographic Information Science, Vol.27, No.1, p.65-73.

Jones, Bundy, and Ware, 1995, “Map Generalization with a Triangulated Data Structure”, Cartography and GIS, Vol.22, No.4, p.317-331.

Lee, Dan, 2003, “Generalization within a Geoprocessing Framework”, GEOPRO conference proceedings, Mexico City, p.82-91.

Lee, Dan, 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

Peng, Wanning, 1997, “Automated Generalization in GIS”, ITC Publication Series, No.50.

Thom, Stuart, 2005, “A Strategy for Collapsing OS Integrated Transport Network Dual Carriageways”, The 8th ICA Workshop on Generalization and Multiple Representation, A Corunia, http://ica.ign.fr – Activities 2005 – Workshop Program.

Wang, Zeshen, and Lee, Dan, 2000, “Building Simplification Based on Pattern Recognition and Shape Analysis”, Proceedings of the 9th International Symposium on Spatial Data Handling, Beijing, China, p.58-72.

Ware and Jones, 1998, “Conflict Reduction in Map Generalization Using Iterative Improvement”, GeoInformatics, 2:4, p.383-407.