GENERALIZATION WITHIN A GEOPROCESSING FRAMEWORK

Dan LEE
Software Development Department, ESRI, Inc.,
380 New York Street, Redlands, CA 92373
USA

ABSTRACT

Map generalization, the process of data transformation, reduction, and integration, requires a powerful and flexible environment in modern GIS. With a new architecture and user experience, ArcGIS, the object-oriented generation of ESRI's GIS product, provides a spatial framework to support GIS and mapping needs. Geoprocessing, combining its earlier command operation with a modern user interface, has become an integral part of the upcoming releases. Developing generalization tools within a geoprocessing framework has given us opportunities to explore new technology and data models and to make enhancements using better techniques. This paper briefly reviews the research and development in the past few years, introduces the geoprocessing concepts and environment, and discusses how map generalization tools have been enhanced and implemented in the geoprocessing framework, and outlines what remains ahead.

KEY WORDS
geoprocessing, generalization, data integrity

1. INTRODUCTION

Many data providers, map producers, especially national mapping agencies, and other GIS organizations have built high resolution, high accuracy databases to represent the geographic world. The potential of using these detailed master databases to serve multiple-purpose and multiple-scale applications can be greatly extended if the automated generalization becomes available in modern GIS systems. Tremendous efforts have been made within the research and development community from deriving numerical methods (as summarized in [1] and [2]) to implementing generalization functions into commercial GIS products and receiving evaluations from major national mapping agencies [3]. Better qualification of generalization solutions and full integration of generalization capability for deriving new datasets and compiling cartographic produds has become inevitable.

1.1 Previous development

In pursuing GIS-based map generalization, the concepts and classifications of generalization operations were defined [4] and a set of the most requested generalization tools was created for the coverage data model in ESRI's Workstation ArcInfo in late 1990s [5]. These tools were implemented following generalization principles and produce less complexity and reduced detail in the output, as shown in Figure 1, while preserving close representations of the geographic objects and meeting data integrity requirements. Our main tasks included defining generalization rules, creating algorithms, setting up logical procedures, facilitating post-processes, and supporting user's requests and benchmarks. The above practice prepared us to meet new challenges.

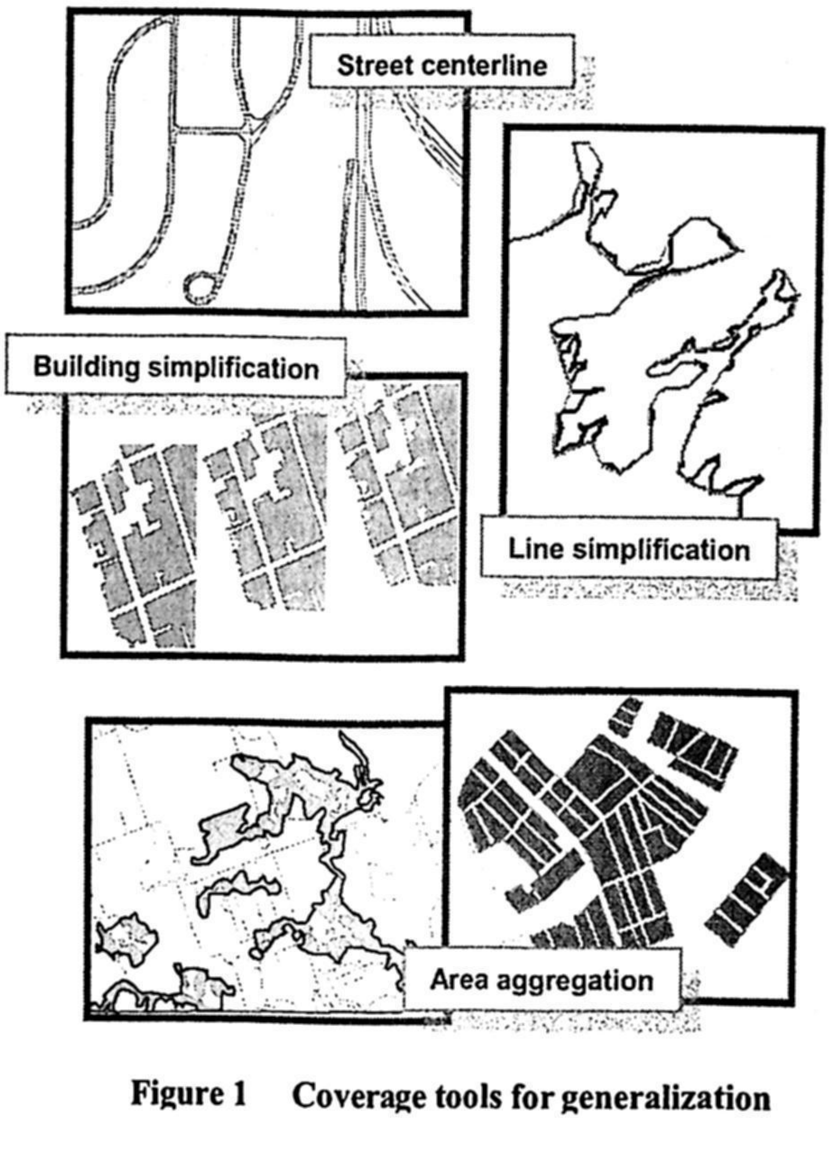


Figure 1 Coverage tools for generalization

1.2 Critical experiences

Successful automated generalization translates human knowledge in manual generalization into explicit rules and logics so that they can be coded in computer language. Very few manual generalization guidelines

 S. Levachkine, J. Serra & M. Ege nhofer (Eds.) – ISBN: 970-36-0103-0

exist in textbooks, and they are usually too general and incomplete. For example, the instructions for area building simplification state: "The measured area of the simplified outline should remain roughly the same as the area of the original", "General form should be maintained", and "If possible, draw rectangles" [6]. When we created the Arc command BUILDINGSIMPLIFY, we had to analyze existing maps, use reverse engineering and common sense to extract explicit rules for the implementation. Here are some examples:

- A building must be simplified if it contains one or more sides shorter than a specified length.
- Building simplification should preserve and enhance orthogonality, that is, making near-90-degree corners exactly 90 degrees.
- A building can be simplified by filling up corners, cutting off or widening isolated small spaces (intrusions or extrusions), or by straightening or averaging a number of consecutive sides, while keeping the measured area roughly the same.
- A building that does not retain a minimum area will be excluded, if the user chooses to.
- Under relatively large reduction, a building can be turned into a rectangle taking the shape of the bounding box oriented along the longest side and an area close to the original.
- For connected buildings, only the outer boundaries are simplified and the buildings should remain connected after simplification.
- Five simplification statuses are to be recorded to support post editing; they are: properly simplified, partially simplified due to potential conflict, simplified but too small, partially simplified group, non-simplified group.

A previous paper presented more details on building simplification [7]. For every generalization tool we develop, a set of such explicit rules must be defined and enhanced over time.

Geographic databases usually store features in various themes, for example as political boundaries, transportation, vegetation, and so on. Each can contain point, linear, and polygonal feature types and their attributes. Generalization may involve designing a new classification for a target output, selecting features for inclusion, reducing details through simplification, aggregation, typification, and so on, and resolving spatial and cartographic conflicts for the final products. These operations must be carried out in a logical order; sometimes decisions may depend on the status of the intermediate results; interactive inspection and editing may be needed. In response to some users' requests and major benchmark specifications, we delivered procedures that transform large-scale data into a small-scale space with fairly encouraging results [8]. However, without a complete integration of generalization capability, from enriched databases that support generalization analysis and decisions to a powerful, flexible framework that facilitates automated processes and interactive compilation, the limitations of the early solutions were obvious.

2. GEOPROCESSING IN ARCGIS

With object-oriented technology and the new geodatabase for modeling the world, ArcGIS marks a new generation of ESRI software. The upcoming release of ArcGIS 9.0 will present a geoprocessing framework for carrying out GIS operations. The integration of generalization tools into ArcGIS has been in progress with the ultimate goals to support database generalization and cartographic generalization from geodatabases.

Taking a relatively broad definition, geoprocessing in ArcGIS 9.0 [1] refers to the application of core GIS operations that create new spatial data from existing or derived data. The basic GIS capabilities found under this umbrella include data conversion, spatial analysis, and data management. A typical geoprocessing operation takes input geodatabase features, performs an operation on them, and returns an output geodatabase feature class. For example, the Buffer tool takes point, line, or polygon features and creates a buffer polygon feature class based on user-specified parameters.

The two core components in ArcGIS Desktop are ArcCatalog (database creation and management software) and ArcMap (start-to-end mapping software). The geoprocessing tools have been made accessible from the dockable ArcToolbox window and the Geoprocessing window in both software environments. Figure 2 shows the access to the ArcToolbox window and the Geoprocessing window in ArcCatalog and the partially expanded view of the toolboxes and toolsets.

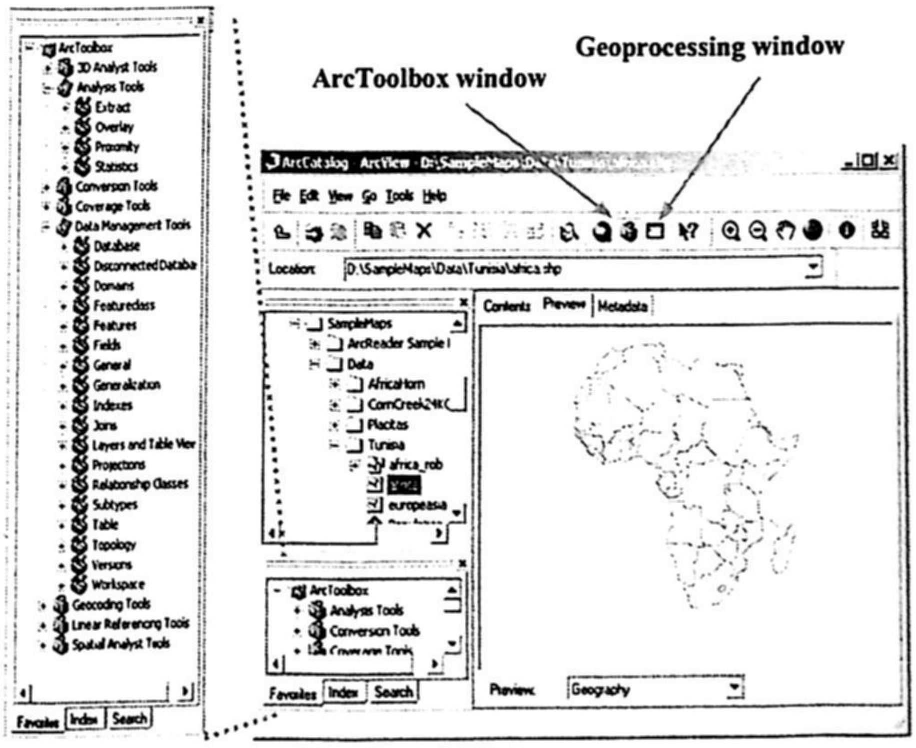


Figure 2. Access to geoprocessing in ArcCatalog

To perform geoprocessing tasks, you can choose one of the following four methods: tool dialogs, command lines, model tools, and scripts (Figure 3). A tool dialog can be invoked from the ArcToolbox window. The dialog gives an easy user interface for you to specify data and parameters to perform a single operation. A command line, similar to Workstation ArcInfo command line, allows you to specify and execute a cammand in the Geoprocessing window. Once you type in a tool name, the Geoprocessing window will prompt you with the

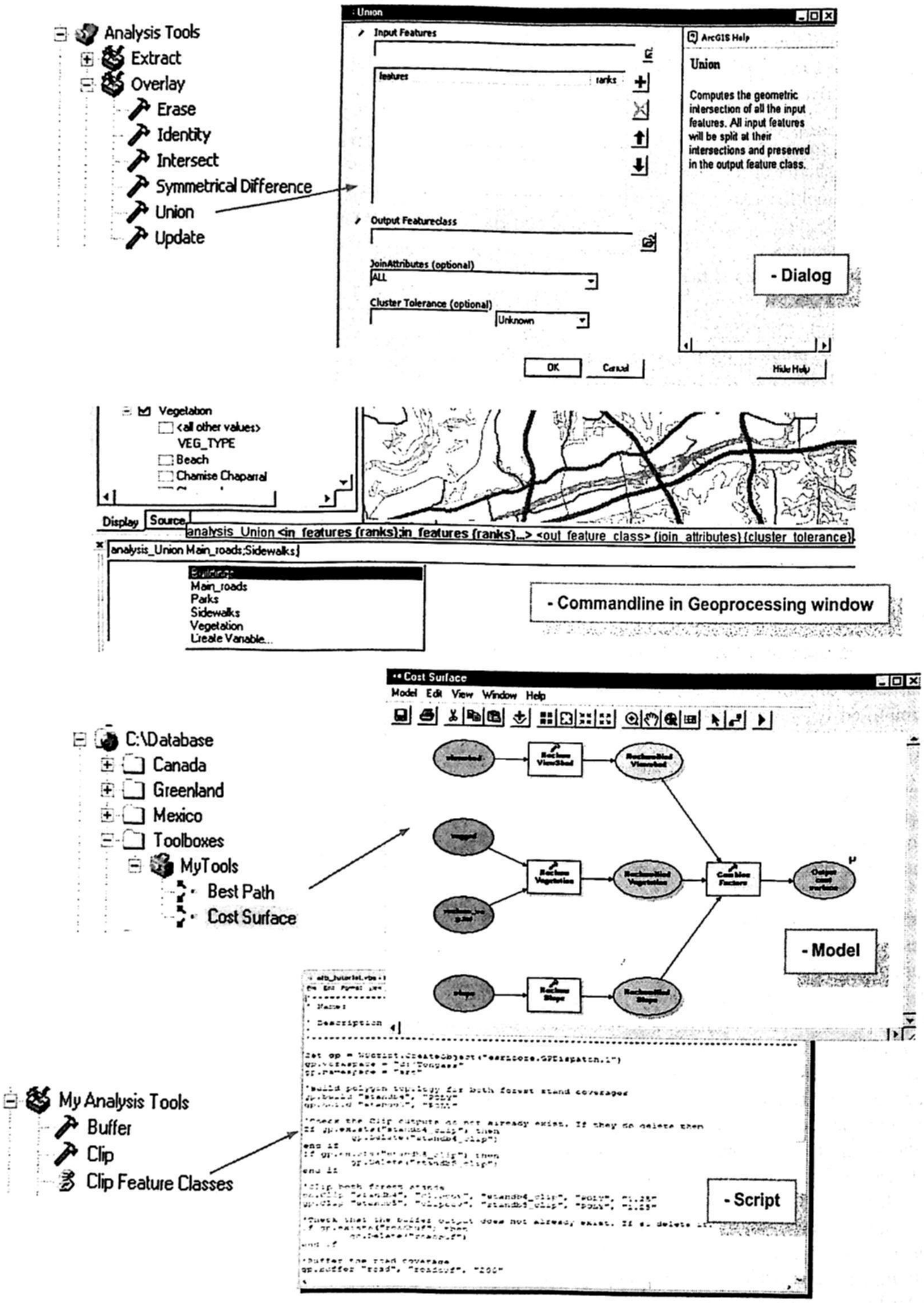


Figure 3. The four methods for performing geoprocessing tasks

usage of the command so that you can enter parameters and options accordingly, then execute the command. Both dialog and command line allow one tool execution at a time. A model can be created in Modelbuilder, which provides you with a graphical environment to construct a diagram of the steps–representing a model–to complete a geoprocessing task. A model executes processes in chained sequence. A script offers an efficient and effective way of managing geoprocessing tasks, especially those involving a large volume of data, repetitive work, and more complex decision-making. The development platform for ArcGIS Desktop applications is known as ArcObjects, a collection of the Component Object Model (COM)-based software components with GIS functionality and programmable interfaces [10]. The Geoprocessor is an ArcObject that supports the COM interface IDispatch, which enables interpretive and macro languages, such as VBScript, JScript, and Python, to access COM objects. This IDispatch object is called IGPDispatch and it exposes all geoprocessing tools to scripting clients.

A variety of environmental settings, such as default workspace location, output extents, cluster tolerance, and so on, can be set and applied at the application level, the model level, or a specific tool level. The geoprocessing framework sets the fundamental, flexible environment for users to manage geographic data operations.

3. DEVELOPING GENERALIZATION TOOLS IN GEOPROCESSING

The integration of generalization into ArcGIS will tremendously extend the power of using master databases for multiple purposes and multiple scales applications. Our development is underway with the ultimate goals to support both database (or model) generalization and cartographic generalization, as distinguished by researchers [11] from geodatabases.

For database generalization, a new dataset or database is derived from a master database with a reduced level of detail, usually for a smaller scale analysis or representation. Such a process can be a single operation on particular features, for example simplification of rivers or aggregation of buildings, or a logical sequence of generalization operations in conjunction with other necessary steps to reach the desired result among multiple feature classes. The geoprocessing framework in ArcGIS described above provides an ideal environment for managing database generalization and preparing data for cartographic finishing. Each generalization tool will be made in compliance with all other geoprocessing tools and can be executed via one of the four geoprocessing methods mentioned above. Certain parameters and options, for example minimum spacing, symbol sizes, and so on, can be specified as environmental settings (not in place yet) and used to guide generalization processes.

2.1 Coverage generalization tools

To continue supporting coverage model applications, the Coverage Tools toolbox in geoprocessing contains tools created from Workstation ArcInfo commands, each of which takes coverage input and produces coverage output. All of the generalization commands implemented in Workstation ArcInfo, as mentioned above, are included in the Generalization toolset in this toolbox. The conversion tools going between coverage and geodatabase features (see more about geodatabase features below) make it possible to process data interchangeably.

2.2 Geodatabase feature generalization tools

The geodatabase model is an object-oriented data model created with ArcGIS and stores geographic data in a commercial off-the-shelf DBMS [12]. The geodatabase extends the traditional coverage model with support for intelligent features and complex networks. Geographic features can now be defined and stored as objects with rules, behaviors, and relationships to other objects.

Based on our earlier research and understanding about generalization [13], a set of generalization functions have been defined to be built for geodatabase features; some are relatively simpler and more straightforward; others are more complex and involving features in context. We have begun to implement the simpler tools that each performs a unique generalization operation.

One of the geoprocessing toolboxes is named Data Management Tools, under which a Generalization toolset is created (see Figure 2) to contain geodatabase feature generalization tools. Like all the other geoprocessing tools, generalization tools will take geodatabase features as input and produce geodatabase feature classes as output. Currently, for line generalization, an enhanced Line Simplification tool and a new Line Smoothing tool have been added into this Generalization toolset (Figure 4). More generalization tools have been planned for future releases.

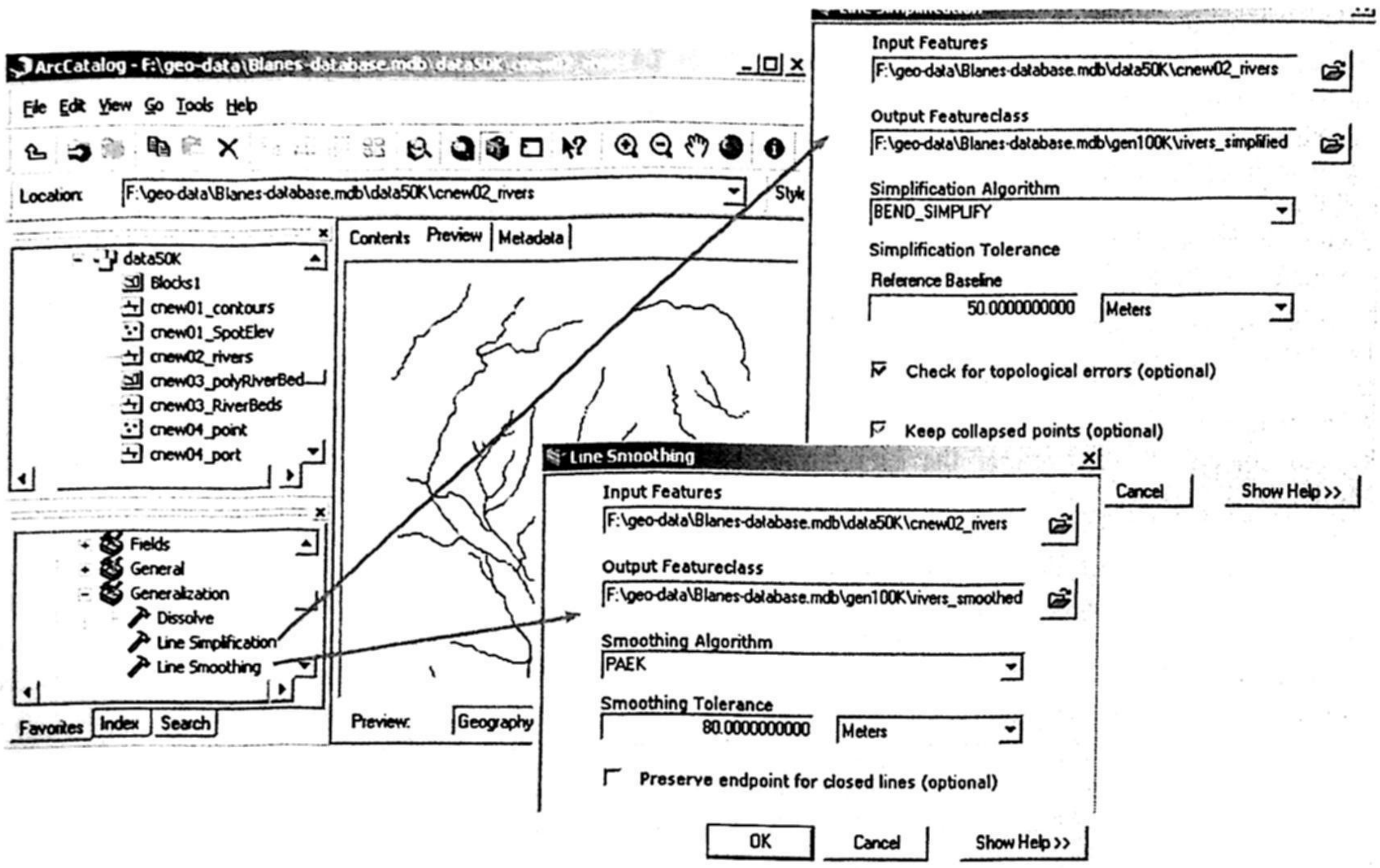


Figure 4 **Line Simplification and Line Smoothing tool dialogs**
(Thanks to the Institut Cartografic de Catalunya for providing the testing data.)

2.3 General principles and requirements for developing generalization Tools

To ensure the quality of generalization and data integrity, some general principles and requirements are closely followed. The Line Simplification tool will be used as an example to explain and illustrate our development considerations through out this section.

Using efficient and effective techniques

Without going into long discussion, an efficient and effective generalization technique simply means it carries out the unavoidable reduction of feature complexity "in such a way that typical characteristics are least affected" [14] and the computation is elegant, easy to understand and control, and inexpensive.

To simplify digital lines, the two primary tasks are to compress over-digitized vertices and to remove small undulations so that "the course of the feature is to be perceived without ambiguity" [15] at the intended scale. There have been a wide range of published algorithms for line simplification, some are listed in [1] and [2]. We chose two distinctive algorithms: POINT_REMOVE, which is an enhanced version of the well-known Douglas-Peucker algorithm [16] and compresses lines or serves minor line simplification quite effectively, while creating more angularity; and BEND_SIMPLIFY, which is an in-house designed algorithm that reduces extraneous bends along lines and preserves the essential shapes of the lines and aesthetic quality quite well [17]. Figure 5 shows the nature and the difference between the two line simplification algorithms.

POINT_REMOVE **BEND_SIMPLIFY**

Figure 5 **The two distinctive line simplification algorithms**

Determining parameters

Usually generalization parameters determine the degree and quality of generalization. For research and learning, it might be nice to expose all involved parameters to the users so that they can analyze and understand the impact of each parameter by fixing other parameters. However, for easy usability and production work, too many parameters can be confusing and hard to control; simplicity in parameter design would be preferred.

For the Line Simplification tool, currently only one parameter needs to be set for either algorithm; other additional necessary parameters are internally derived, taking either empirical values or values that are logically related to the specified parameter.

The question always arises: how to come up with a reasonable value for a parameter? For geoprocessing tools, wherever possible a default value will be provided for a numerical parameter. However, for the line simplification tool or any generalization tools, unless the scale related specifications or some measurements about the data are available, no reasonable values can be suggested for the parameters. Based on our experience, setting the line simplification tolerance for both algorithms can begin with a value in a ground unit close to or a little greater than one converted from the minimum allowable spacing between lines on the map at the target scale; trials and errors may be expected in order to reach a suitable tolerance value.

Resolving topological errors

Many generalization operations more or less alter the geometric representation of the features. As soon as that happens, the spatial relationship among features might be destroyed or become incorrect. For example, when the shape of a building outline is simplified, it may overlap a neighboring building outline, which in reality is impossible. A generalization tool should avoid creating these types of topological errors, or if they are created during the process, try to resolve them.

Topological errors that might be created in line simplification are: line-crossing, coincident lines, and collapsed zero-length lines. The user has the option to have these errors detected and resolved. There could be many different ways to do so. Since these errors usually occur in congested areas, they indicate that the simplification tolerance may not be suitable for those areas, usually that it's too large. So here is how we resolve the problems:

The data will be simplified using the specified tolerance. In order to find out if errors are created, the Delaunay Triangulation will be constructed from which information about line-crossing and coincident lines can be extracted. Each pair of involved line segments will be located, a reduced tolerance (half of the original) will be applied to re-simplify these segments, and the resulting line segments will be used to update the triangulation. If errors still occur, a reduced tolerance (half of the last used) will be applied to the involved line segment. This iteration will repeat until no more errors are found. Figure 6 shows a comparison between an input line and its simplified form. The small bend the arrow points to is much smaller than those in the left circle, but could not be removed as were those in the right circle without causing line-crossing; so it was under-simplified and kept in the result.

The collapsed zero-length lines could only happen to closed (circular) lines when the tolerance is relatively too large. This type of error can be detected easily without using triangulation. When an error is found, a reduced tolerance will be used to re-simplify the original closed line. Again, the iteration will repeat until the line won't collapse anymore.

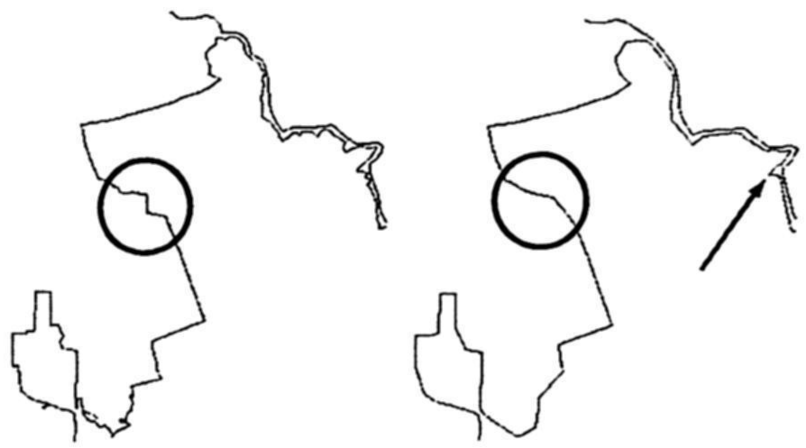


Figure 6 **Before (left) and after (right) simplification: where the arrow points at is obviously less-simplified compared to the shape change in the circles, as the result of resolving line-crossing errors.**

Although the result of line simplification won't contain the above three types of topological errors, other types of topological errors may still occur, for example, a severely simplified line might end up being on the opposite side of a point feature. The resolution to this problem has not been implemented.

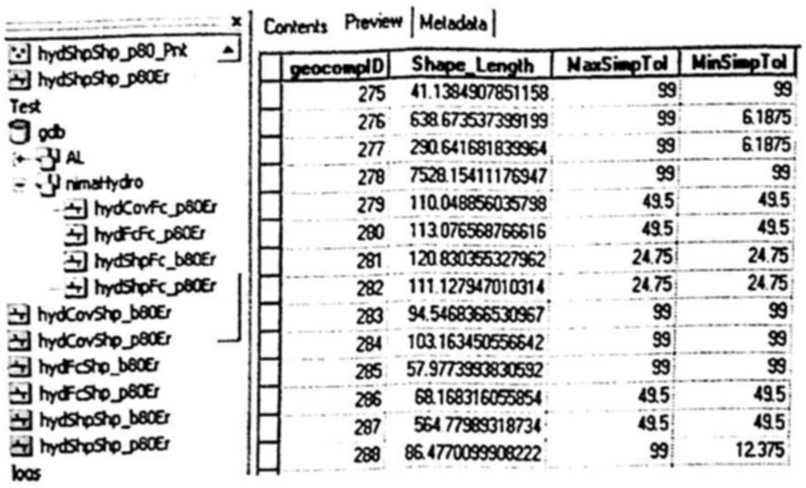
Flagging status

It is very likely that the automated generalization process cannot produce a complete and satisfactory result. Therefore, the generalization process status and problems must be identified and flagged to support the evaluation of the result, the analysis of the parameters and option used, the post processing (interactive or semi-automatic), and the research for enhancements in the incomplete and problem areas. Quite often, the flagged areas may require a different type of generalization operation.

Again in the line simplification case, the iterative approach of resolving the topological errors described above may result in a line simplified by possibly the specified and reduced tolerances in different parts. To make the user aware of the situation and be able to review the under-generalized lines easily, two new attributes, MaxSimpTol and MinSimpTol (the maximum and minimum simplification tolerances used to simplify a line), are written for each line in the output feature class. The user knows immediately what range of tolerance is used for a particular line and whether the specified tolerance is suitable for the majority of the data. The partial attribute table of a simplified line feature class in Figure 7 illustrates the MaxSimpTol and MinSimpTol

© Lee GEOPRO 2003 – ISBN: 970-36-0103-0

values. In this case, the specified tolerance was 99 map units. Where 49.5, 24,75, 12.375, and 6.1875 are listed, topological errors were found and resolved by reducing tolerances for the problem line segments in one through four iterations.



geocompID	Shape_Length	MaxSimpTol	MinSimpTol
275	41.1384907851158	99	99
276	638.673537399199	99	6.1875
277	290.641681839964	99	6.1875
278	7528.15411176947	99	99
279	110.048856035798	49.5	49.5
280	113.076568766616	49.5	49.5
281	120.830355327962	24.75	24.75
282	111.127947010314	24.75	24.75
283	94.5468366530967	99	99
284	103.163450556642	99	99
285	57.9773993830592	99	99
286	68.168316055854	49.5	49.5
287	564.77989318734	49.5	49.5
288	86.4770099908222	99	12.375

Figure 7 Partial attribute table showing the MaxSimpTol and MinSimpTol values.

The under-simplified areas may imply the need for a different generalization operation, for example two closely located lines representing a narrow river may not be simplified without crossing each other. Perhaps they should be collapsed into a single line representation. A collapsed zero-length line may need to be excluded.

<u>Insuring data integrity</u>

There are many aspects that define data integrity. To generalization, data integrity means, at least, that the data is complete for the target output, consistently created and processed, properly linked, and timely updated. Some generalization operations reduce the level of details at individual feature level, for example, collapsing an area building to a point; therefore, the generalized features and their source features have one-to-one relationships. Others may reduce the number of features in a group or combine features in a group, for example, aggregating trees in close proximity into forest areas; therefore, the generalized features and their source features may have one-to-many relationships. It is important that the generalized features are properly linked to their source features so that the feature attributes can be calculated and transferred to the generalized features.

One of the data issues in line simplification is about how to properly handle shared geometry. Shared geometry is very common in route networks; for example, one part of a road may be shared by different routes. Theoretically the shared part of the features should be simplified the same for each route. With the availability of the topology engine in ArcGIS, we are able to identify such coincident line segments. For each group of coincident lines, the geometry of the lines will be simplified only once and the resulting geometry will be used to replace all coincident segments in the group. In other words, shared geometries in route network are consistently processed with the coincident segments simplified and still coincident.

Also in line simplification, if the user did not choose to detect and resolve the potential topological errors, then there is a good chance that some closed lines will be collapsed into zero-length lines, as mentioned above. These features become invalid in the geodatabase and cannot be stored in a line feature class. To inform the users about these "lost" data and allow them to keep track of where the lines are, a point feature class will be generated to carry the endpoints of the collapsed or lost lines with their source line object IDs; the user can then decide to delete them, if they are indeed unimportant, retrieve the original lines through the linked object IDs, or do something else.

Although lines are simplified in shape, they still represent the same individual geographic objects, that is, they have one-to-one relationships. Therefore, normally all the source line attributes including the object IDs are copied over to the corresponding simplified lines.

2.4 Importance of generalization models

The generalization process may not be straightforward; to model the process is always a challenge. The Modelbuilder mentioned earlier helps us to experiment with different procedures, adjust the workflow according to different themes and target maps, and make the generalization processes easy to manage. You can create and edit a model diagram in Modelbuilder to put the generalization steps in a desired sequence. The diagram can be saved as a model in a user-specified toolbox and modified easily to repeat the same or similar processes for different datasets or for the same data with different parameters and options.

The model diagram in Figure 8 illustrates an experimental building generalization sequence. The goal is simple:

- Larger buildings (larger than 6000 sq ft) will be kept as areas and only their footprints will be simplified
- Medium-size buildings (4000 – 6000 sq ft), since they are too small to remain as areas but large enough to be included, will be represented by points
- Small buildings (smaller than 4000 sq ft) will simply be excluded.

The input clip_bldgA is a large-scale area building feature class. It is first converted to a coverage so that the coverage Building Simplification tool can be used to simplify the building footprints, keeping only the three large buildings. Then, in a parallel process, the Select tool selects the three medium-size buildings from clip_bldgA based on a query, that is, size greater than 4000 and smaller than 6000 sqft, followed by the Feature To Point tool that collapses the buildings into their corresponding centroid points. All other buildings are left out. It is possible that some of the small buildings are aggregated if they are in close proximity; constraints can be added so that the aggregation won't cross major roads. But for the simplicity of illustration, aggregation was not considered. Three other simple models were also used to generalize the rivers (collapse, simplify, and smooth), roads (select, collapse to centerlines, and extend), and contours (select, simplify, and smooth). More details will be given at the presentation due to the limitation of the paper length. Figure 9 shows the result of the experimental generalization.

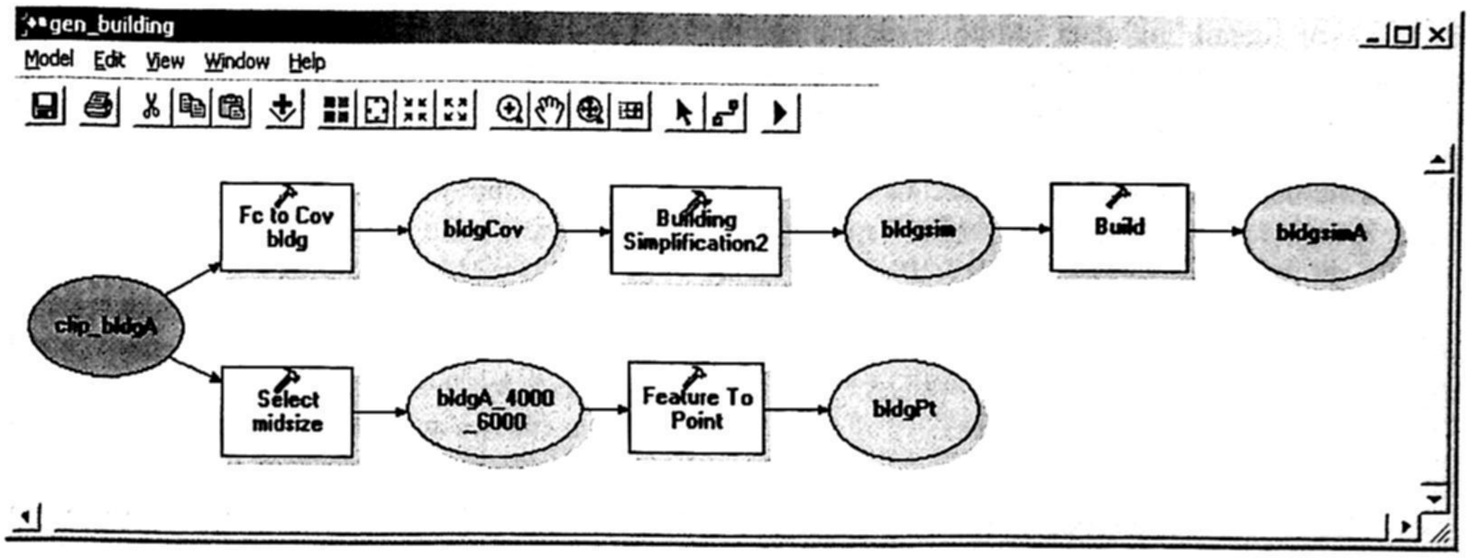


Figure 8 **Example model for building generalization**

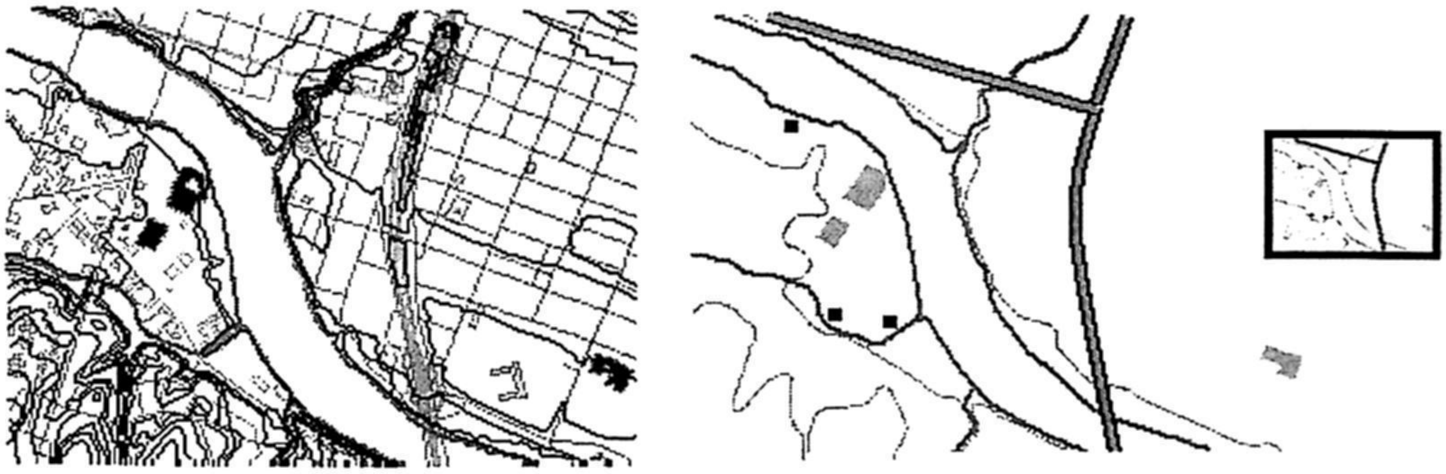


Figure 9 **Before (left) and after (right) generalization; result at a smaller scale (in the black box)**
Source data: the USGS 1:24000 mapsheet, Austin East, TX, USA

In reality a much more complex model or even a number of models might need to be set up for generalization. Features to be included for the target database or map must be selected from the master database; different generalization operations, parameters, and sequences are to be decided for various themes and features; some intermediate and final results need to be inspected and possibly edited. As shown in the generalized map at the reduced scale in Figure 9 (about four times reduction), some buildings look too close to rivers and may need to be displaced that the space between a building and a river is perceivable and no new conflicts are created. A complete generalization workflow could include automated processes done by models and any necessary interactive or semi-automatic processes. Being able to record the whole workflow and easily alter it for repeated use and updating would be very helpful in production. Our research and development continue.

4. ONGOING AND FUTURE WORK

For line simplification in particular, there are still areas that need enhancement: bottle-neck areas may need to be widened, small consecutive bends, usually representing the switchbacks in mountain roads or rivers, may be combined into fewer and larger bends, multiple features need to be taken into account when detecting spatial conflicts, line symbol width will need to be considered for cartographic quality. The recursive or iterative approach of resolving topological errors may not be the only or best way. Other techniques and choices are to be explored.

The Line Smoothing tool has been implemented following similar general principles as for line simplification. Although the topological errors occurring in smoothed results can be detected and flagged, the resolution of these errors has not been fully defined.

We will continue to derive techniques that satisfy the essential requirement of generalization, that is, reducing the level of detail in data while maintaining the characteristics of the presented geography. It is not the purpose of the this paper to discuss the details of the rules, algorithms, and steps for all the planned generalization tools, but tremendous research has been made to analyze feature properties and preserve their representation faithfully in reduced forms. Figure 10 shows some preliminary results of developing area aggregation tools and collapsing dual-lines to centerlines tool using triangulation.

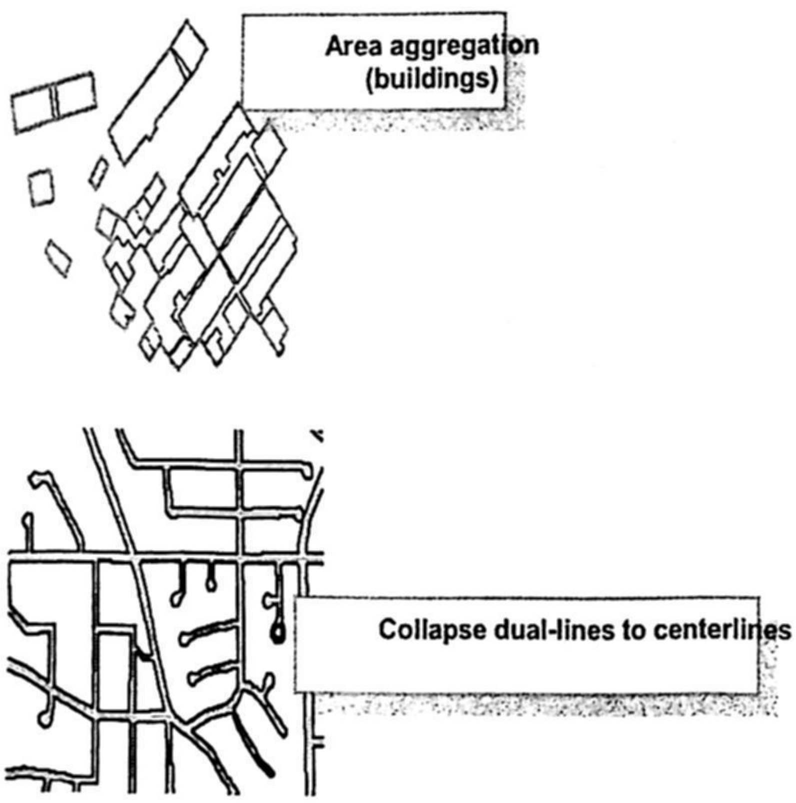


Figure 10. Development of area aggregation and collapse tools in progress

The full integration of generalization in ArcGIS must consider interactive operations in addition to automated batch processes. Interactive generalization means allowing interactive selecting of features, applying generalization to the selected features, dynamically viewing the changes while altering a parameter, being able to undo and redo the changes, and saving the results as needed. When a mapped area contains mixed levels of complexity, a uniform generalization process may not be the desired solution. The interactive generalization could help to apply different parameters and choices to different areas based on visual judgment, and do it right.

Until automated generalization can produce perfectly satisfactory results, if that is ever possible, the completion of generalization tasks will still need to count on the necessary interactive editing and refinement of the automated results. Some of the unresolved areas or imperfect solutions can be flagged, as already discussed above. An efficient post-editing management environment should be part of the geoprocessing framework. Meanwhile, specialized post-editing tools will need to be developed; they can be semi-automatic, meaning that a human being makes the decisions and choices that the computation cannot make, and the computer does the work. For example, the coverage tool Create Centerlines flags unresolved intersections (too complicated to connect properly in the automated process). An interactive specialized post-editing tool could let the user pick which lines should be connected and which intersection style to use (assuming a set of pre-defined intersection styles, such as "T"-intersection, "Y"-intersection, and so on, is available) and let the program finish making the desired intersection. The current geoprocessing framework has not yet supported interactive generalization and post-editing, but the design and planning are underway.

One very important and often neglected side of the integration of generalization in our GIS systems is database design. Building effective classification of features and enriching databases with necessary geometric, relational, and attributive information to support generalization is very critical. No matter how sophisticated the techniques used to create generalization tools are, their ability to understand and achieve what a human being can see and do is always limited [18]. The fact that computational methods simply cannot interpret geographic differences from purely feature geometry and may result in the loss or distortion of spatial integrity after generalization, such as disconnected road or stream networks, broken boundaries, or misplaced features, has led to the increasing interests and demands on multiple-scale representations and database-driven cartography. Our research and prototype efforts have begun.

5. CONCLUSIONS

There is obviously a long journey ahead in pursuing generalization solutions. The development of generalization tools in the geoprocessing framework is just the first major step towards our goal of supporting database and cartographic generalization.

As geoprocessing development advances, more generalization tools will be added and more features in context will be taken into account in generalization operations. Our main focus will be on providing flexible and practical ways of managing generalization work (interactive or automatic), formalizing generalization guidelines and models for further automation and enhancement, following up with post-editing management and tools, and exploring database potentials in balancing the computational limitation and expenses.

REFERENCES

[1] R.B. McMaster and K.S. Shea, Vector-based Generalization in *Generalization in Digital Cartography* (Washington D.C.: Association of American Geographers, 1992) 71-98

[2] Elsa Maria Joao, The algorithmic approach in *Causes and Consequences of Map Generalization* (London: Taylor & Francis Ltd, 1998) 10-13.

[3] B. Baelia, M. Pla, and D. Lee, Digital Map Generalization in Practice, the working paper for *ICA Workshop on Progress in Automated Map Generalization*, Barcelona, 1995.

[4] ESRI, Automation of Map Generalization in ArcInfo – the cutting edge technology, search for map generalization at http://arconline.esri.com/arconline, 1996.

[5] ESRI, Map Generalization in GIS: Practical Solutions with Workstation ArcInfo Software, search for map generalization at http://arconline.esri.com/arconline, 2000.

[6] Swiss Society of Cartography, *Cartographic Generalization – Topographic Maps*, Cartographic Publication Series, No.2 (Zurich: Swiss Society of Cartography, 1987) 22

[7] Z. Wang and D. Lee, Building Simplification Based on Pattern Recognition and Shape Analysis, *Proceedings of the 9th International Symposium on Spatial Data Handling*, Beijing, China, 2000, 58-72.

[8] D. Lee, Data Conversion and Generalization from 1:1000 Scale Database to 1:5000 Scale Database, benchmark report to Hong Kong Land Information Center, 1997.

[9] ESRI, ArcGIS Overview, http://www.esri.com/software/arcgis/overview.html, 2003.

[10] ESRI, ArcObjects Technology, http://www.esri.com/software/devsolutions/arcobjects/index.html, 2003.

[11] R. Weibel, and C.B. Jones, Computational Perspectives on Map Generalization, *GeoInformatica*, 2(4), 1998, 307-314

[12] ESRI, ArcGIS Geodatabase, http://www.esri.com/software/arcgis/geodatabase.html, 2003.

[13] ESRI, Automation of Map Generalization in ArcInfo, search for map generalization at http://arconline.esri.com/arconline, 1996.

[14] E. Imhof, *Cartographic Relief Presentation* (Berlin: Walter de Gruyter, 1982) 55.

[15] R.W. Anson, *Basic Cartography for Students and Technicians*, Vol.2 (London and New York: On behalf of International Cartographic Association, Elsevier Applied Science Publishers, 1988) 37.

[16] D. H. Douglas and T. K. Peucker, Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Character, *The Canadian Cartographer*, 10(2), 1973, 112-123.

[17] Z. Wang, Manual versus Automated Line Generalization, *Proceedings of the GIS/LIS '96 Annual Conference and Exposition*, 1996, 94-106.

[18] D. Lee, Making Databases Support Map Generalization, *Proceedings of the GIS/LIS '96 Annual Conference and Exposition*, 1996, 467-478.