GENERALIZATION WITHIN A GEOPROCESSING FRAMEWORK

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

ABSTRACT

Map generalization, the data transformation, reduction, and integration process, 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/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 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 what remains ahead.

In pursuing GIS-based map generalization, a set of most requested generalization tools had been added to ESRI's "classic" ArcInfo system, now called Workstation ArcInfo. These tools derive reduced data with less complexity and detail to satisfy the target scales and other requirements. With the coverage data model, features are processed through commands. Our main practices included defining generalization rules, creating algorithms and procedures, facilitating post-processes, and supporting user's requests and benchmarks, which prepared us to meet new challenges.

With the object-oriented technology and the new geodatabase to model the world, ArcGIS marks a new generation of ESRI software. The integration of generalization tools into ArcGIS has been in progress with the ultimate goals to support database generalization and cartographic generalization from geodatabases. The upcoming release ArcGIS 9.0 will present a geoprocessing framework to support the applications of core GIS and mapping operations. To perform geoprocessing tasks, you can choose one of the following four methods: <u>a tool dialog</u> (which gives an easy user interface for you to specify data and parameters to perform a single operation), <u>a command line</u> (which specifies and executes a process in the command window), <u>a model</u> (which is a diagram of the steps - representing a model - to complete a task, constructed in Modelbuilder), <u>a script</u> (which offers an efficient and effective way of managing user's geoprocessing needs, especially for tasks involving large volume of data, repetitive work, and more complex decision-making). The geoprocessing framework sets the fundamental environment for user to manage database generalization (deriving less detailed, simplified data) and to prepare data for cartographic finishing.

In developing generalization tools in the geoprocessing framework, a number of basic, but critical principles are followed closely, including the use of efficient techniques to

generalize features while preserving their natural representations, the maintenance of feature relationships, the insurance of data integrity (completeness, consistency, and so on), the support for the evaluation of the results and necessary post processing. The new line simplification tool will be used to illustrate the above principles. The line simplification tool uses one of the two algorithms: POINT_REMOVE, which is an enhanced version of the most popular Douglas-Peucker algorithm, and BEND_SIMPLIFY, which is an in-house algorithm that reduces extraneous bends along a line and preserves aesthetic quality. This tool provides an option for resolving certain types of topological errors (line-crossing, line-overlapping, and collapsed zero-length lines) potentially produced in the simplification process. The less simplified lines will be “flagged” for inspection or post-processing. The one-to-one feature relationship and all attributes are preserved.

A couple of simple scenarios will be used to illustrate how models are set up for generalization. Generalization sequences should follow common logics, but could vary according to data structure, target output specifications, and the quality of the generalization tools. In reality a quite complex workflow might need to be set up for generalization, from feature selection to the choice of generalization operations, parameters, to the inspection and possible edits of intermediate results, and to the final evaluation and cartographic refinements.

Our future integration of generalization will eventually not only support the automated operations, but also the interactive operations. Meanwhile, we have also realized the increasing interests and needs for multiple-scale representations and database cartography. Our research and prototype efforts are underway.