EXTENDING A GENERAL-PURPOSE GIS FOR COMPUTER-ASSISTED GENERALIZATION

Andreas Schlegel and Robert Weibel

Department of Geography
University of Zurich
Winterthurerstrasse 190
8057 Zurich (Switzerland)
Fax: +41-1-362 52 27
{schlegel, weibel}@gis.geogr.unizh.ch

Abstract

This paper reports on a project whose major objective was to develop a prototype system allowing to assess the potential of extending the basic functionality of a general-purpose GIS in the direction of more specialized generalization tools. The GIS used as a carrier system was ARC/INFO in this case (although it should be noted that many other systems would have been equally appropriate!). A variety of generalization algorithms as well as several supporting facilities were programmed in the Arc Macro Language (AML) and C, and imbedded in a graphical user interface. The extension of a general-purpose system, in principle, offers a feasible way of enhancing the usability of GIS for cartographic production. The major problems encountered with the current prototype system were the slow computational performance and restrictions of the underlying data model of the GIS. Suggestions for further enhancements and future research are presented.

1 Introduction

The overwhelming majority of existing geographic information systems (GIS) offers some basic capabilities which may be exploited for generalization, such as selection/elimination of specific features, line simplification (usually the algorithm by Douglas and Peucker [3]), smoothing by splines, and amalgamation of adjacent polygons or line segments. Generally, however, these tools are developed rather poorly, and most often restricted to rather primitive operations. Generalization operators are usually implemented in an inflexible way; users are not supported in their decisions which tool to use and which parameters to apply. The consequence is that mapping from GIS frequently involves interactive editing or even laborious and expensive re-digitizing whenever projects need to be carried out at different scales. The alternative is to ignore generalization, which can lead to illegible maps, examples of which are easy to find throughout the world of GIS.

This study therefore focused on improving tools for computer-assisted generalization in GIS software, with a two-fold objective [10]. First of all, the question was how complex generalization functions could be built into a general-purpose GIS by re-combination of basic tools, extensions to the user-interface, as well as integration of additional generalization methods. Secondly, the potential of this approach should be evaluated with respect to its usability for map production. A prototype system called *GenTools* was developed on the basis of a the commercial

GIS ARC/INFO. (Please note, however, that most other GIS packages allowing macro programming and binding of software modules written in high-level programming languages could be used just as well!). In section 2, the conceptual and technical foundations of this project are discussed. Sections 3 and 4 are devoted to a description of the design and implementation of GenTools. Subsequently, section 5 presents a brief evaluation of the current prototype. Finally, the concluding section discusses suggestions for future research.

2 Conceptual and Technical Basis

The development of GenTools was based on several elements. The conceptual foundation was formed by an architectural framework for interactive cartographic systems termed "amplified intelligence", as well as a model for the modularization of the overall generalization process. Elements of GIS, in particular ARC/INFO, formed the technical basis of this project.

2.1 Interactive generalization and amplified intelligence

Cartographic generalization represents a complex and ill-structured process. Therefore, its automation cannot be comprehensively solved by purely algorithmic methods nor by expert system (ES) techniques. As a consequence of this observation, a conceptual framework for generalization systems, called amplified intelligence (termed so in deliberate contrast to artificial intelligence), was outlined in [14] and subsequently defined in more detail in [15]. This concept attempts to combine the strengths of both algorithms and ES. The human is kept in the design loop: key decisions default explicitly to an expert user, who initiates and controls a range of algorithms and tools which automatically carry out generalization tasks. Algorithms are imbedded in an interactive environment and complemented by various supporting facilities (e.g., tools to determine feature clustering, spatial conflicts and overlaps), thus providing decision support to the user. The approach is thus obviously related to decision support systems (DSS), and also shares similarities with so-called cooperative systems in the domain of artificial intelligence [11]. Knowledge is contributed directly by the expert user or is incorporated within the support facilities, generalization algorithms or a rule base. In the design specifications made in [15], five research areas were deemed relevant for the development of interactive generalization systems:

- data models;
- generalization operators;
- support facilities;
- human-computer interaction (HCI);
- methods for knowledge acquisition and representation.

The objectives of this project were certainly much more pragmatic and mundane. Still, it was decided to use the conceptual basis of amplified intelligence as a long-term perspective for developing the demonstration prototype. Of the five domains listed above, however, the last one, which is most associated with "intelligence" was obviously not relevant to this project.

2.2 *Modularization of the generalization process*

In order to be amenable to interactive operation on the basis of individual tools, the overall generalization process must be modularized into meaningful components. Various authors have proposed models in an attempt to explain the elements of generalization; an overview of some models is given in [8]. The most useful model for the purposes of this project was that by McMaster and Shea [8], because its modularization of the generalization process is rather detailed. A preliminary design of generalization operators and support facilities (e.g., for cartometric analysis) can be directly deduced from that framework.

Generalization operators (how to generalize) are subdivided into spatial and attribute transformations. Ten operators achieving spatial transformation — simplification, smoothing, aggregation, amalgamation, merging, collapse, refinement, exaggeration, enhancement, and displacement — are defined. For attribute transformations, two operators — classification and symbolization — are proposed.

Support facilities can be based on the functions for cartometric evaluation outlined in [8]. Cartometric evaluation governs the need for generalization (when to generalize). A first class of functions is devoted to testing *geometric conditions* such as congestion, coalescence, conflict, inconsistency, or imperceptibility. A next group involves *spatial and holistic measures* quantifying inherent characteristics of cartographic features such as their density, distribution, length, sinuosity, and shape. Finally, *transformation controls* aim at supporting the selection of generalization operators, algorithms, and associated control parameters.

While the validity of this model still needs to be assessed in experimental studies, it did provide a very practical framework for our research.

2.3 *Geographic Information Systems*

Graphics and maps are the predominant means of communication used in the exploration, assessment, and presentation of information in GIS. Therefore, all GIS offer functions to generate maps. Usually, the larger general-purpose GIS products contain quite extensive cartographic functions. Additionally, systems of this kind include a great variety of tools for interactive data editing, and data manipulation (e.g., selection, filtering). Many of these functions could form useful building blocks of computer-assisted generalization. Unfortunately, however, these functions are made available to the user as separate and atomic commands performing rather narrow tasks. It is thus very hard for most users to achieve good generalization at reasonable effort. On the positive side, most systems offer the possibility of using a macro language to build more comprehensive tasks. The above observations form the technical basis of the work reported here.

The commercial GIS ARC/INFO (Release 7.0.2) by ESRI was used in this study. The system implements both vector and raster data models; in this project, only the vector part was used. It has a highly modular structure building on a great number of commands which can be bound into macros using the Arc Macro Language (AML) which also allows to build graphical user interfaces (GUI). This GIS, given its functionality, architecture, and associated software tools, can be considered as typical of a large class of commercial general-purpose GIS. As a first step of this project, the functions of ARC/INFO were evaluated for their potential to act as elements of an interactive generalization system [10]. Functions from the following software modules were then used: ARC (basic commands for data manipulation); Arcedit (interactive geometric and attribute editing); Arcplot (graphics subsystem); Arc Macro Language (AML; command macros and GUI); ArcTools (building blocks for GUI development); Inter Application Communication (IAC; communication between multiple processes/modules).

3 System Design of GenTools

The design of GenTools concentrated on the four domains data model, human-computer interaction (HCI), generalization operators, and supporting facilities. No requirements study was carried out in order to relate the system design to the needs of specific production applications. The specifications were purely based on the objective of this project to develop a demonstration prototype offering sufficient functional breadth and depth to assess the feasibility of the approach.

3.1 Data model

Since GenTools was developed on the basis of ARC/INFO, the basic data model was already given. It was attempted to apply the layer concept defined by the "coverages" of the carrier system in a consistent way. A coverage in GenTools is a layer representing a single feature class (e.g., spot heights, roads, buildings) of a single topological type (point, line, polygon), along with its attributes. Through the generalization process, an input (source) coverage is transformed into an output coverage (result), corresponding to the representation of a feature class in the source map and the resulting target map, respectively.

3.2 HCI

The user interface of GenTools was designed according to the basic principles of graphical user interfaces outlined in [1], including elements such as direct manipulation, visual feedback, and a high level of consistency of the workflow. Since ArcTools was used as a basis for GUI development, many building blocks (e.g., functions to open/close coverages, standard dialogs, GUI widgets) could be taken from that toolbox, which resulted in a generally coherent and consistent appearance.

The layout and the major user interface elements of GenTools can be seen in Figure 1. Their are three graphics windows: the central working window used for interactive generalization of coverage data, and two optional windows showing the situation in the source map (upper left) and target map (upper right) with symbol size/width according to the associated scale. In the working window, symbolization of point and line features can either be set to a centerline display, to the symbol size/width of the source scale (as is the case in Figure 1), or to the size/width of the target scale (for detection of overlaps and coalescence). Optionally, a raster map can be displayed as a background. All graphics windows have zoom menus associated with them.

The functions and tools of the system are accessible via the main menu bar (to the top of the working window) or via tool palettes. These palettes combine frequently used functions. Various palettes are available for graphical feature selection, generalization operators, basic editing functions (Arcedit), command level entry, and further aids. Not shown in Figure 1 is a further type of window which is presented to display descriptive statistics such as histograms in support of feature selection and generalization.

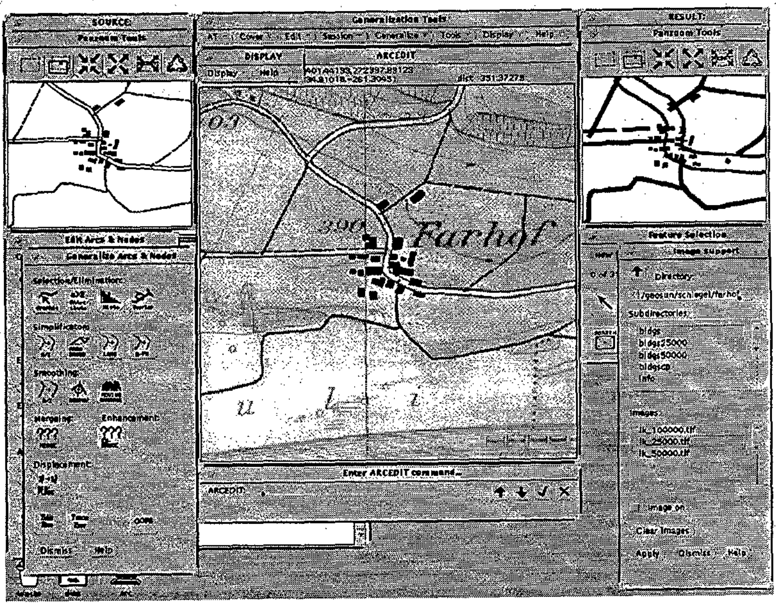

Figure 1: Screen snapshot of a GenTools session. The elements of the screen layout are explained in section 3.2. Given the restrictions of Arcedit, only one editcoverage (layer) can be generalized at a time. In this case, it is the `roads` coverage; the `buildings` coverage is displayed as background. Symbolization of `roads` in the source and target map (result) corresponds to 1:25,000 and 1:50,000, respectively. Symbolization in the working window corresponds to 1:25,000. The scanned basic map respresents part of the national map of Switzerland 1:25,000, © of the Swiss Federal Office of Topography (D-2289).

3.3 Generalization operators

The primary objective of designing generalization operators for GenTools was to offer selected functions representative of different topological types (point, line, polygons) and of varying complexity as a basis for further experimentation. Thus, the available operators do not comprehensively cover the list proposed by [8]. For any *generalization operator*, one or more *generalization algorithms* are available which implement the task defined for the operator. On the user interface, operators represent buttons on the generalization tool palettes; appropriate algorithms and associated parameter settings are subsequently selected in a form menu.

The operators and algorithms available in GenTools are listed in Table 1. Two things become apparent: 1) particular emphasis was put on offering a large variety of selection algorithms in order to improve interactive generalization support; and 2) some (but only few) algorithms could directly be based on ARC/INFO commands, thus only requiring better imbedding in the interactive context.

feature selection / elimination	– selection by attribute value (e.g., on the basis of precomputed shape or distribution measures) – selection by histograms (drag relevant range) – selection by Radical Law [12] – selection by classification (e.g., by road classes) – selection by geometrical overlap
simplification (lines)	– Douglas algorithm [3] (`generalize`) – Lang algorithm [6] – area based line generalization [13]
smoothing (lines)	– `spline` command of ARC/INFO – moving average (simple and weighted) – Brophy algorithm [2]
aggregation (points, small areas)	– *interactive:* 1) objects to be aggregated must be selected (e.g., via minimal distance), 2) initial aggregation by means of the convex hull (`thiessen`), 3) interactive editing of the outline
amalgamation (lines, areas)	– `dissolve` command to remove common boundaries between adjacent polygons or edges of networks carrying same attribute values
exaggeration (points)	– highlighting of problem cases (size below minimal dimensions), subsequent interactive editing – object enlargement by means of scaling factor derived from scale reduction factor
displacement	– highlighting of problem cases (overlaps, coalescence, below minimal dimensions), interactive editing – propagated displacement with a distance decay function (interactively invoked)

Table 1: Generalization operators and algorithms available in GenTools.

3.4 Supporting facilities

Supporting facilities are of great importance in an interactive cartographic system because they enable the user to explore the structure and properties of the source data. Tools for exploratory analysis of input data are very rare in current GIS or cartographic systems. Therefore, the major objective here was to demonstrate the usefulness of having a variety of such functions in the cartographic toolbox. Table 2 shows the shape and distribution measures currently offered by GenTools.

points	– arithmetic mean center – center of minimum travel (median center)
lines	– measures proposed by [4]: average segment length, coefficient of variation of average segment length, error variance, coefficient of variation of error variance, average angularity, coefficient of variation of average angularity, total angularity, curvilinearity
areas (polygons)	– area (calculated automatically by ARC/INFO) – perimeter (calculated automatically by ARC/INFO) – ratio area/perimeter

Table 2: Shape and distribution measures in GenTools.

While this is a rather narrow selection of quite simple measures, they gain more power because they are made available to the user in a variety of ways:

1) select individual object, compute and show desired measures;
2) select multiple objects, compute desired measures and store in database attribute;
3) select single or multiple objects, query shape attributes from database and display as list;
4) same as 3, but display shape attributes as histogram;
5) same as 4, but use histogram for selection and highlighting (cf. Table 1).

Besides shape measures used to characterize geometric object properties, further interactive aids are made available (Table 3).

"yard stick"	– used to interactively check minimal dimensions; available as line or polygon object
visualization of object properties	– show classed properties/attributes by color or line width – show unclassed properties by continuous variation of color or line width
highlighting of problems	– automatically find overlaps and intersections and zoom in on each problem area (for editing) in turn – zoom in on each object of a selected set of features in turn – find objects which are too close (uses command `buffer`)

Table 3: Further interactive supporting facilities in GenTools.

4 Some Implementation Details

GenTools was implemented using the following components: AML; ArcTools; commands from ARC, Arcedit and Arcplot; IAC; and the C programming language. Most functions were implemented as AML macros exploiting commands of the GIS were possible, and adding missing algorithms. For performance estimation, a limited set of algorithms was implemented in C. For lack of space, we restrict the description to a few selected functions (for more details, see [10]).

Graphics and HCI: Graphics and visualization functions were completely based on Arcedit and Arcplot commands. The user interface was written in AML using components of ArcTools; some ArcTools functions (e.g., the image support dialog) were streamlined for better performance. Arcedit was used to drive the main working window, while the "source" and "result" windows (cf. Figure 1) were controlled by an Arcplot process communicating with Arcedit via IAC.

Area based line generalization: In this algorithm by [13], vertices on a line are eliminated based on the area of the triangle which they form with the vertices immediately before and after in the chain. The algorithm starts with the vertex having the smallest area and repeatedly eliminates points and recalculates areas of adjoining points until a user-specified threshold is reached. Because this algorithm is iterative, it was chosen for a performance comparison of AML vs. C. In the AML version, all vertices are read from the database in one go. The order of elimination from the line is then computed for all vertices and kept in a list. Thus, according to the tolerance value interactively specified via a slider bar, vertices can be eliminated or added. The result is only written to the database when confirmed by the user. As an alternative, a mixed version was written, with the actual algorithm in C, while reading and writing the line geometry is carried out in AML.

It turned out that the AML version of the algorithm was too slow for "real-time" slider operation. The C version was sufficiently fast, but since database access could only be achieved through AML, that part was slow; additionally, the transfer of the line geometry data to the C program (which is a separate process) needs to be carried out via a file. This situation could be vastly improved by using the application programming interface library ArcSDL (Arc Software Development Library), which allows direct database access from C programs (cf. section 5).

Propagated displacement: This algorithm is based on an idea communicated by [5]. Given one or more points causing a displacement (e.g., points on a line), and a vector indicating the major direction and magnitude of the displacement, surrounding objects are shifted while the degree of displacement decays with increasing distance. In GenTools, propagated displacement was implemented using three parameters (entered interactively): 1) initial displacement vector indicating the direction and distance within which full displacement takes place; 2) distance at which displacement decays to zero; and 3) distance decay function (several available). A grid of displacement vectors is then interpolated within the affected region (the grid arrangement was chosen in order not to have to read the coordinates of the affected features). These vectors are then passed to the Arcedit command `adjust` which performs a piecewise linear rubber-sheeting transformation of the affected features. While `adjust` simplifies the implementation of displacement, its performance is so slow that slider operation is not feasible. Instead, it seems preferable to additionally write a proper rubber-sheeting transformation.

5 Discussion

The discussion of GenTools must be restricted to the major points; for a detailed evaluation, see [10]. The major ***advantages*** of the approach chosen here relate to the ease of implementation, to increased user-friendliness and usability, and enhanced functionality. Using the tools available from the carrier GIS — commands for graphics, geometric manipulation and analysis; database functions; macro language; and GUI elements — an interactive generalization system with considerable functionality could be built in a very short period of time (approximately 6 months of graduate student time). Of course, GenTools must be considered as a demonstration prototype regarding functionality — it is far from being a complete system. Yet, the system allowed to compare the ease of use and usability of complex interactive generalization algorithms to the functions offered by the basic GIS. In informal experiments, it was found that the ease of use was drastically improved by the provision of GUI elements such as sliders. Users can more readily locate, invoke, and control relevant functions for generalization. Generalization operators and associated algorithms are provided as logical units, as compared to sequences of individual commands of the GIS, which the user first has to identify in a cumbersome trial-and-error process involving lengthy research in the documentation. The usability is further improved by the fact that specific generalization algorithms deliver better results than the unspecific commands for geometric manipulation available in the GIS. Thus, the need for interactive retouching is reduced.

The major *disadvantages* of GenTools are due to performance limitations and restrictions of the data model. Performance is mainly affected by two facts: 1) AML is an interpreted and thus slow language; and 2) access to the database through AML is not optimized (e.g., repeated and often unnecessary open/close of geometry files or the attribute database). This situation could be improved by using ArcSDL (cf. 4) which is installed at our site. With ArcSDL, the major computational parts of the software could be programmed in C, using ArcSDL functions for direct access of database tables and files; use of AML would then be restricted to GUI and graphics functions. Unfortunately, however, version 7.0.2 of ArcSDL was not yet available at the time of this project, and thus this alternative could not be explored for the current version of GenTools. The second major problem, restrictions of the GIS data model, relate to the following issues: 1) the layer structure of ARC/INFO's data model represents a serious impediment to ensuring a good interplay of the different feature classes (displacement, topological consistency); 2) the data model is not rich enough to support complex operators such as displacement and amalgamation, implying additional computational effort for the generalization software; and 3) topological inconsistencies cannot be easily checked. Since these restrictions are imposed by the underlying GIS, there is no easy way of overcoming them.

A further deficiency of the current version of GenTools is that generalization algorithms are not rich and specific enough (e.g., no algorithms to simplify and amalgamate building outlines). Finally, the implementation of GenTools has shown that sometimes, seemingly insignificant details have a rather significant effect. In ARC/INFO, the action which is controlled by a slider bar widget is only invoked when the mouse button is released, and not during the process of pushing the slider to the left or right. Thus, "dynamic" generalization operated by sliders is not really possible in the same way as, for instance, in Intergraph's Map Generalizer [7].

6 Conclusions

This paper has presented an overview of the conceptual basis, design, and implementation of GenTools, an interactive generalization system based on ARC/INFO. For us, the value of this project and the implemented system is three-fold. Firstly, it allowed to assess the feasibility of the use of GIS and the tools offered by them for the purpose of map generalization; deficiencies of current GIS in this respect could thus be identified. Secondly, it offers a platform to demonstrate concepts such as interactive generalization, amplified intelligence, and supporting facilities. We hope that demonstration prototypes such as GenTools can help convince vendors of the importance of generalization and show possible ways of coping with the problem. Thirdly, though incomplete, GenTools offers a convenient platform to drive some of our research projects. For instance, the system was already used as a basis of an experiment in knowledge acquisition by process tracing [9].

As a consequence of the third point, future plans focus mainly on streamlining and extending the system in light of other research projects. Relevant parts needed to support specific research will be rewritten in C, use of AML restricted to graphics and GUI functions, and the database accessed directly through an application programming interface. Generalization operators and algorithms need to be improved and tailored more specifically to particular problems. Also, support facilities need to be significantly extended, most notably by adding further shape measures as well as facilities for evaluation of generalization alternatives. The tight coupling with GIS, however, still seems logical as GIS will need to be improved in terms of their functionality for generalization sooner rather than later.

References

[1] Apple Computer, Inc. (1992): *Macintosh Human Interface Guidelines.* Reading, MA: Addison-Wesley, 384 pgs.

[2] Brophy, D.M. (1972): *Automated Linear Generalization in Thematic Cartography.* MA Thesis, Dept. of Geography, U. of Wisconsin.

[3] Douglas, D.H., and Peucker, Th.K. (1973): Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. *The Canadian Cartographer,* 10(2): 112-122.

[4] Jasinski, M.J. (1990): The Comparision of Complexity Measures for Cartographic Lines. *National Center for Geographic Information and Analysis, Technical Paper,* 90-1, 73 s.

[5] Lagrange, J.-P. (1994): *Personal communication.* Institut Géographique National, Paris.

[6] Lang, T. (1969): Rules for the Robot Draughtsmen. *The Geographical Magazine,* 42(1): 50-51.

[7] Lee, D. (1993): From a Master Database to Multiple Cartographic Representations. *16th Int. Cartographic Conference,* Cologne: 1075-1085.

[8] McMaster, R.B. and Shea, K.S. (1992): *Generalization in Digital Cartography.* Washington, D.C.: Ass. of American Geographers, 134 pgs.

[9] Reichenbacher, T. (1995): Knowledge Acquisition in Map Generalization Using Interactive Systems and Machine Learning. *17th International Cartographic Congress of the ICA,* Barcelona (E), 3.-9.9. 1995.

[10] Schlegel, A. (1995): *Erweiterung eines universellen GIS für die computergestützte kartographische Generalisierung.* M.Sc. Thesis, Dept. of Geography, University of Zurich, 108 pgs.

[11] Stolze, M. (1990): Task Level Frameworks for Cooperative Expert System Design. *AI Communications,* 4(2/3): 98-106.

[12] Töpfer, F. (1974): Kartographische Generalisierung. Gotha-Leipzig: VEB Hermann Haack, 335 pgs.

[13] Visvalingam, M., and Whyatt, J.D. (1993): Line Generalisation by Repeated Elimination of Points. Cartographic Journal, 30(1): 46-51.

[14] Weibel, R. (1991a): Amplified Intelligence and Rule-Based Systems. In: Buttenfield, B.P., and McMaster, R.B. (eds.): *Map Generalization — Making Rules for Knowledge Representation.* Longman, 172-186.

[15] Weibel, R. (1991b): Specifications for a Platform to Support Research in Map Generalization. *Special Session on Cartographic Generalization,* 15th ICA Conference, Bournemouth (GB), 23.9.-1.10. 1991.