OBJECT-ORIENTATION, CARTOGRAPHIC GENERALISATION AND MULTI-PRODUCT DATABASES.

P. A. Woodsford

Laser-Scan Ltd, Science Park, Milton Road, Cambridge, CB4 4FY, UK.

Fax: (44)-1223-420044; E-mail peterw@lsl.co.uk

Abstract

Considerable advances have been made in the use of common geographic base data to support a wide range of cartographic specifications over a limited range of scales. Table-driven techniques, using multi-part prioritised representation and automated masking, are extensively used to generate, for example, physical geography and road atlas specifications from a common database, with consequent economics and benefits in revision and bespoke mapping.

A portfolio of techniques for cartographic generalisation have come into use, with varying degrees of effectiveness. Selection, sampling, simplification and alternative representation methods are well established. Techniques for solving the combinatorial issues (conflict resolution, displacement, exaggeration, aggregation and fine-tuning of name placement) are known, but are essentially 'ad hoc' and consequently less effective. The strong positive impact of the hybrid raster/vector approach to digital cartography also poses interesting generalisation issues, in terms of sampling techniques to maximise the preservation of graphic content.

The case is made that the new generation of object-oriented GIS technology provides for the first time an appropriate theoretical framework for the development of a systematic and effective solution to cartographic generalisation, and hence to the realisation of multi-product cartographic/GIS databases. The essential properties of object-oriented technology are outlined, with special reference to the use of methods and behaviours, the handling of topology and other relationships, data modelling and integration and versioning.

The paper finally describes current research and the development of a new generation system based on the Laser-Scan Gothic object-oriented application development environment.

1. ADVANCES IN CARTOGRAPHIC PRODUCT GENERATION

In recent years, techniques for computerised production of maps and charts to a range of specifications from common base data have matured to the point where they are widely and successfully used by both commercial publishers and governmental agencies [6]. This has been achieved without prejudice to cartographic standards - computer generated maps are no longer poor relations of conventional products. Key elements in this progress have been:

the realisation of hybrid raster/vector approaches, as described in the paper by Hardy and Wright in these proceedings [2]

the consolidation of productive techniques for raster-to-vector conversion, for those components of the map content that require vector representation

advances in representation techniques, in particular the use of table-driven techniques to separate graphical parameters from the geographic base data. Multi-part prioritised representation, for example, readily handles complex road casing symbolisation and is the basis for automated masking for text and symbols the emergence of an effective *de facto* standard for the implementation of typography in a common form across computer displays and proofing and finishing plotters. Use of PostScript and Display PostScript gives access to industry standard fonts for many languages, and effective WYSIWYG graphics across the whole cartographic production process

the continuing dramatic increases in computer hardware power and falling digital storage costs underpin these advances. The ever increasing "horsepower" available to system developers both permits and necessitates a new generation of software architecture in order to deliver benefits to users.

2. CARTOGRAPHIC GENERALISATION

As the other elements of productive and economically successful automated cartography have come into place, practical attention has become increasingly focused on the cartographic generalisation problem - long a staple topic for research activities. For vector-based cartography, techniques of selection, sampling, simplification and alternative representation or symbolisation are well established. Flexible table-driven representation in conjunction with a well thought out coding schema provides readily usable access to such techniques. For example, the provision of a null representation is a simple means of omitting a particular feature class in a given cartographic product. Alternatively selection can be provided by menus. Algorithms for simplifying base data for display at a particular scale, or for improving cartographic appearance by, for example, building squaring, are often used. In practice they need to be supplemented by powerful general purpose capabilities like automatic masking. The utility of a system function or procedure to resample a contour line in a cartographically acceptable manner needs to be complemented by a system function to automatically position and mask out height labels, or much of the practical value is lost.

In summary, the aspects of generalisation that can be handled on a per feature class basis are more or less readily automated; those that are specific to individual features or combinations of features are still in practice handled interactively, albeit with assistance from relevant tools. Such combinatorial issues include conflict resolution, displacement and exaggeration, aggregation and the fine-tuning of name placement. No one seriously looks to a fully automatic process to resolve all these issues; the key is still to find an approach which provides a sufficiently high degree of automation, together with techniques for managing the residual level of interaction, so as to make production of cartography at a substantive range of scales from common base data an economic reality.

In the pure raster or hybrid raster-vector approach where one or more layers (or colours) are held and maintained as raster data, resampling and more sophisticated image processing techniques can be used to "generalise" raster layers in the sense of change in their scale or appearance. Particular care has to be taken in defining the rules for combination of raster layers, and to provide preferential treatment for layers of more significant information content. For example, a "black-wins" algorithm can be used to maintain as much as possible the legibility of raster text held in a black layer. The system issue is usually that of providing access to all the relevant raster and vector functionality in a coherent and user-friendly manner.

3. OBJECT-ORIENTATION

It has become increasingly apparent that the feature-based approach, even when complemented by a relational data model, provides limited scope for the implementation of a multi-product, automated cartography capability with powerful generalisation facilities. A better and more coherent theoretical framework is needed. For some time, it has been postulated in academic circles that this is provided by the object-orientation paradigm. Thus J-C Muller [5] in a survey article on "Generalisation of Spatial Databases" writes:

> "Finally, the database must be object oriented to afford an object-oriented programming approach to generalisation.. Much work remains to be done, however, in defining what the semantic of a truly object-oriented approach for automated generalisation should look like."

In an object-oriented database, real world entities are abstracted and held as objects. All objects belong to object classes. For each class there may be many objects, but each object belongs to only one class. The class defines what values can be held by an object. Values can be simple datatypes (integers, strings, dates, etc) together with more specialist types (geometrics, locations, rasters, tables). Furthermore, objects can hold structural information or references between objects.

A key, and defining, concept of object-orientation is that of methods defined on objects. These methods are bound to behaviours. When a method on an object is invoked, the behaviour bound to it is executed, possibly using values and references also held by the object. The ability to define behaviours as part of the database schema, rather than as part of the application, is a fundamental concept of object-orientation.

A further key concept is that of inheritance, which provides the means to define a new object class in terms of existing classes. The new class inherits the characteristics (values, references, behaviour methods) of its parent class or classes, unless superceded or redefined. Using inheritance, trees of classes can be created and maintained in a systematic manner.

Object-orientation has gained much popularity in software engineering and computer graphics. Of late, object-oriented GIS software has become available and has been successfully applied, particularly in facilities management applications, as described for example by Hartnell and MacAllister [3]. For GIS applications, object-orientation has to be harnessed in conjunction with powerful spatial indexing techniques, capable of performing efficiently with large amounts of spatial data. The use of versioning techniques to manage change across long transactions and within very large continuous datasets is also in practice a key element in the new wave of GIS technology loosely characterised as object-oriented or "O-O", although it is not strictly part of object-orientation. As is inevitably the case when there is a generation shift in technology, hype takes over from clarity and the reader will find it difficult to find a GIS that does not seek to fly under the Object flag. A recent article seeking to restore some clarity to the situation is to be found in [1]. In reality there are still relatively few systems that support all the key elements to a level that can successfully support large GIS applications. The automated cartography application, with acceptable generalisation capabilities sufficient to support viable multi-product databases across a range of scales, is particularly demanding and has much to gain from a properly object-oriented approach.

4. OBJECT-ORIENTATION AND CARTOGRAPHIC GENERALISATION

The first benefit of object-orientation in the cartographic application is that the data model becomes object-centred rather than geometry-centred. Attention is naturally shifted to the real world entities to be depicted cartographically and away from the ink marks on a particular map. Such a change of focus is necessary to support general purpose GIS data products. It is also beneficial to the cartographic process.

An immediate benefit of the shift to an object-centred model is that an object can hold multiple geometries. For example, different geometries can be used to support fundamentally different display representations within different scale bands, corresponding to the catastrophic change with scale described by Muller [5, pp466-8]. To take a very simple example, urban area outlines can give way to town symbols at smaller scales. Furthermore, an object can hold both vector geometries and rasters. At a very fundamental level, the object-based approach gives a natural integration of vector and raster data. Conceptually, at least, it is possible to hold with an object both vector data to support techniques such as thinning and raster data to support techniques such as area thickening/eroding/agglomeration.

The major benefit, however, comes from the use of methods and behaviours to systematise and organise generalisation procedures. It was observed above that the more difficult generalisations are those involving particular features and operations involving combinations of features. The techniques for performing these are procedural, and may well be rules-based. Behaviours on an object class basis provide a natural way to implement such procedures, and methods the way to invoke them on an individual object instance basis.

An object model provides a very powerful way of implementing topology. In addition to holding values, attributes and geometries, objects can hold connections with other objects. These can be used to implement 'dependency' methods, which can be used to identify dependent objects that are involved when an object is changed. Exploitation of topology in this way, together with efficient spatial indexing, provides a powerful set of tools for addressing the combinatorial aspects of the generalisation problem.

5. DEVELOPMENTS AT LASER-SCAN.

Since 1990, Laser-Scan has been developing an Open Systems object-oriented Application Development Environment (ADE) named Gothic [4]. Gothic applications are built using a versioned object-oriented database, with a full implementation of methods, behaviours and inheritance. Versioning provides each user with a stable view of the database without incurring the costs of making a separate copy. It does so by maintaining a record of changes relative to a parent version. Versioning provides an elegant solution to two problems that have to be solved in order to provide productive solutions to cartographic generalisation and multi-product databases - long transactions and very large continuous spatial datasets.

In Gothic, the topological model supports both simple level classes (points, lines and areas) and low-level, primitive classes (nodes, links and faces). The former are constructed from the latter. The primitive structure forms the topology, and can be spatially indexed. Gothic maintains topology dynamically. User-level object classes inherit their spatial data and topological relationships from simple level classes (points, lines and areas).

Gothic provides 'reflex' methods which are automatically invoked at defined stages in an object lifecycle. In particular, these can be used to maintain integrity whenever objects are modified. Alternatively they can be used to automatically change the cartographic display in consequence of a set of updates to the underlying geographic base data.

Since 1994, Laser-Scan has been developing a new generation Mapping and Charting application using the Gothic ADE. The new application, named Aladdin, uses a central database of map data to support a range of products. Operation is in two phases

Compilation - database creation and maintenance from a range of sources

Product Generation - extraction, symbolisation and generalisation, using WYSIWYG cartographic display

In the initial release (Summer 1995), generalisation functionality is limited mainly to the per object class and per object operations, with a view to exploiting the object-oriented framework to present these relatively well-known techniques in as user-friendly and efficiently automated manner as possible. Examples include the following scale-dependent operations and functions

Selection

Alternative cartographic representations

Thinning and sampling by point frequency

Spatial filtering (e.g. removal of small islands)

Exploitation of multiple geometries

Generated alternative geometries (e.g. points from areas)

Subsequent developments will seek to incorporate more powerful generalisation techniques, including combinatorial operations, in the Aladdin application. Confidence is already growing that the Gothic versioned object-oriented database provides a powerful environment for the realisation of economically useful multi-product, multi-scale cartographic databases.

REFERENCES.

[1] Batty, P. "Object-Orientation - some objectivity please!". GIS 93 Conference Proceedings, May 1993, Birmingham, UK.

[2] Hardy, P G and Wright, P. "Techniques for Update in Raster and Vector Cartography". ICA/ACI Barcelona 1995 (this conference).

[3] Hartnell, T J and MacAllister, B. "Meeting Utility Needs Using Object-Oriented Spatial Technology". GIS 95 Conference Proceedings, may 1995, Birmingham, UK.

[4] Laser-Scan Ltd. "The Gothic Versioned Object-Oriented Database: an Introduction". November 1994, Laser-Scan, Cambridge UK.

[5] Muller, J-C. "Generalisation of Spatial Databases", in "Geographical Information Systems Volume 1: Principles" edited by David J Maguire, Michael F Goodchild and David W Rhind, pp457-75, Longmans, 1991.

[6] Woodsford, P A. "Cartographic Production from a GIS Database". 5th Canadian Conference on GIS, March 1993, Ottawa, Canada.