MAPPING WITH LIVE FEATURES: OBJECT-ORIENTED REPRESENTATION

P.G. Hardy and P.A. Woodsford

Laser-Scan Ltd, Science Park, Milton Rd, CAMBRIDGE, CB4 4FY, UK
Fax: +44 1223 420044, Email: paul@lsl.co.uk and peterw@lsl.co.uk

Abstract

The recent availability of object-oriented database technology opens up new possibilities for map and chart representation, on paper as well as on computer screen. In a traditional map database, features are held as passive data: coordinates and attributes. It has been up to the application program to display or plot these in graphic form, usually by static lookup tables, based on a simple feature code. In the new object-oriented world, this constraint is broken as static features are replaced by dynamic objects.

Instead of the application drawing the features according to a fixed representation derived from a feature code and stored coordinates, the application sends a message to each feature object, asking it to draw itself. Each object can decide how best to draw itself, using any available information. As well as object class (equivalent of feature code) and coordinates as before, this information can include combinations of attributes, relationships with other objects, either explicit or inferred through spatial adjacency, and environmental information such as day of the week. This means that from a single geographic dataset, a wide set of products can be produced, to suit a range of task-oriented requirements.

This paper outlines the representation capabilities of a modern object-oriented cartographic system, and presents examples of dynamic representation for navigation and topographic mapping.

Introduction

Traditional computerised mapping has benefited over the last few years from a series of advances in representation techniques and capabilities, in particular:

- The use of table-driven representation lookup from feature coding, separates graphical parameters from the geographic base data. This allows common geographic data to be used to produce a variety of cartographic products, for example road atlas pages and topographic map pages of a particular area at a similar scale.
- Multi-part prioritised representation, where each feature may be drawn several times with different graphical styles at different levels during the building up of the cartographic picture can achieve for excellent quality map display or hardcopy. For example, this technique readily handles complex road casing symbolisation and is the basis for automated masking for text and symbols.
- The emergence of an effective and powerful de facto standard (PostScript) for graphical hardcopy across proofing and finishing plotters frees the application from the constraints of a particular plotting device.
- The implementation of typography in a common form across computer displays and hardcopy, by 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 software developments. 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.

These advances have taken us down the path towards the ‘holy grail’ of digital cartography - the concept of a single ‘scale-free’ geographic database from which a wide range of products can be derived. These could be visual products (maps and charts at various scales) or data products (for display or analysis).

However, to achieve the transition to this desirable situation, the constraint of static representation must be broken. The advent of Object-Oriented (O-O) mapping products provides the technology for this step into the new world of active objects.

Object-Orientation

In an O-O 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 (geometries, locations, rasters, tables). Furthermore, objects can hold structural information or references between objects.

A key, and defining, concept of O-O is that of methods defined on objects. These methods are bound to behaviours. When a method on an object is invoked by sending a message to the object, 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 the O-O paradigm. The realisation that representation and display can and should be achieved through object messaging is central to the premise of this paper.

A further key concept of O-O 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 superseded or redefined. Using inheritance, hierarchies of classes can be created and maintained in a systematic manner.

True O-O has gained much popularity in software engineering and computer graphics, and is making ground in GIS. 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. An article seeking to restore some clarity to the situation is to be found in [Ref. 1]. In reality there are still relatively few systems that support all the key elements to a level that can successfully support large multi-product mapping applications.

Active representation

In an O-O mapping system, the appearance of an object on the screen or on hardcopy is generated at draw time by execution of an arbitrary ‘display method’ defined on the object class and stored in the database under the direct control of the customer. This contrasts with the traditional approach as indicated in the following table.

O-O Active Representation	Feature Representation
Dynamic - objects can draw themselves differently each time	Static - defined by feature class
Defined in the database	Defined in the applications program
Can be defined and enhanced by the customer	Can only be enhanced by the software supplier
Can be influenced by combinations of attributes	Indexed by single feature code attribute
Can be influenced by attributes derived from other referenced objects	Each feature is represented in isolation
Can adapt to external influences (e.g. required map scale)	Not adaptive

Developments at Laser-Scan

Since 1990, Laser-Scan has been developing an Open Systems object-oriented geographic environment named Gothic [Ref. 2]. 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. See [Ref. 4] for the use of object databases in Cartography and Navigation.

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 is 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. They can also be used to provide active representation, and to automatically change the cartographic display in consequence of updates to the underlying geographic base data.

Since 1994, Laser-Scan has been developing a new generation Mapping and Charting application using Gothic. The application, named LAMPS2, uses a central database of map data to support a range of products. Operation is in two phases:

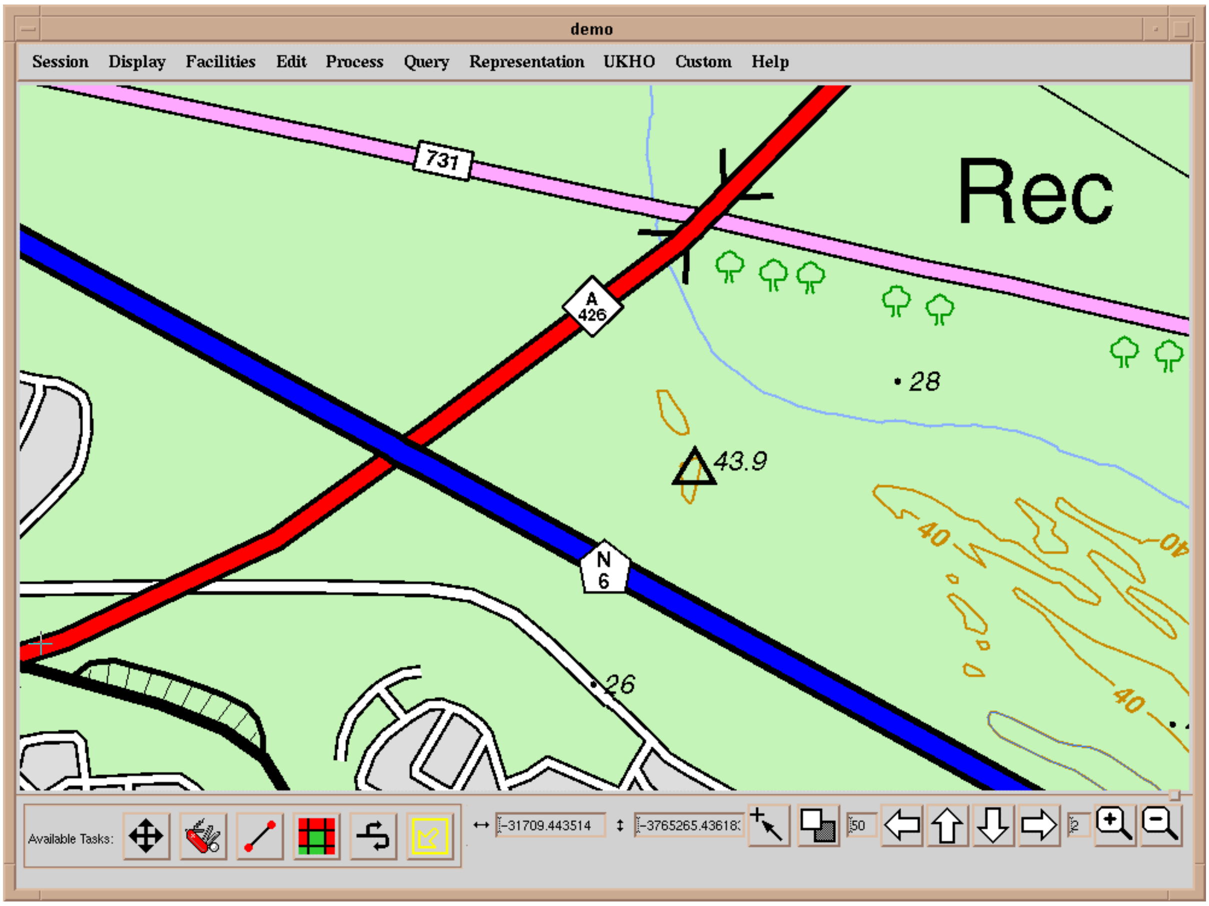
- Compilation - database creation and maintenance from a range of sources.

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

Articles such as that by P.A. Woodsford [Ref. 3] have covered the use of O-O database and specifically LAMPS2 for building multi-product databases and for cartographic generalisation and map production.

Topographic Map Example

The following map extract simulates 1:50000 scale topographic mapping, generated from an object database using display methods. Some of the methods just lookup a static representation, while others take full advantage of the power of the spatial toolkit to derive a complex representation from information extracted from combinations of attributes and references to other objects.



Base map data courtesy of DOSLI South Africa. Copyright RSA.

In particular, the following features should be noted:

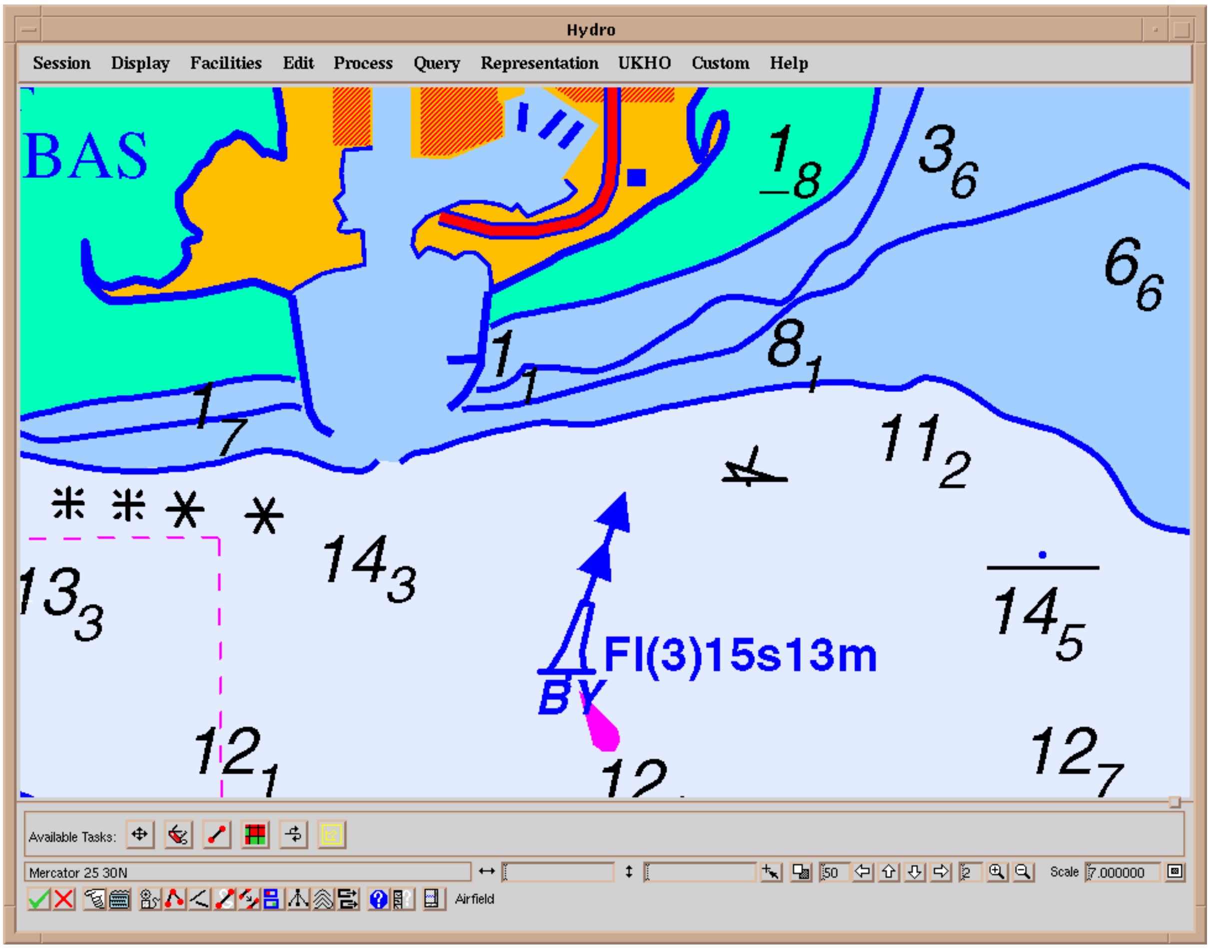
1. The road number symbols are objects of a single class which have no attributes and no coordinates. When digitised, a pre-commit reflex method defined in the database locates the nearest road object and sets up a direct object reference. Whenever a message is sent to the number object, it follows the reference to determine the road number, road type, position, and orientation. It then uses this information to draw itself with the appropriate text and box.
2. The contour label objects similarly extract height, position, orientation from the contour object, and use it to blank out a section of the contour and draw the label, oriented so that it reads correctly going up hill.
3. The bridge object is a short line object which has its ends snapped onto the road (A425), and therefore shares link geometry for the section. It has no line

representation of its own, but repeats the representation of the road to which it is attached, but at a higher drawing priority to ensure that it overdraws the crossing road (731). The display method calculates the position and orientation of the bridge abutments (> <) and draws those also.

4. Roads are drawn as two parts with differing priorities, firstly in black to get the casings, then more narrowly in the infill colour.
5. The trig point uses multiple representation derived from combinations of attributes to get both the triangular symbol and the text label for the height.

Hydrographic Chart Example

The following extract is from a hypothetical navigation chart, and shows several examples of active representation.



Base chart data courtesy of UK Hydrographic Office. Crown Copyright.

In particular, the following features should be noted:

1. The soundings are point objects with several attributes (depth, quality, technique, etc.) Their representation is generated by a display method which takes these attributes into consideration, as follows:

    - Figures after the decimal point are subscripted, unless the depth is greater than a limit, when they are suppressed.
    - Drying heights (negative depths) are shown with an underbar, as in 1.8 at top right.
    - extra symbology is added for soundings with particular attribute combinations, for example no bottom found at 14.5 metres.
2. The rocks and wrecks are single object classes, with symbol selected according to combinations of attributes.
3. The floating navigation mark is a composite object, in that there is a master object (NAMFLO) which has pointers to a set of other objects storing the light, the top mark, the base, etc. The overall representation is determined by a display method which combines multiple attributes to generate the primary text label (3 flashes every 15 seconds), the light colour (BY) and selects appropriate symbology for the topmark, base, light flare, etc.

On-demand mapping

There is a continuing movement away from static media such as printed maps toward dynamic media such as portable computer displays, multimedia CD-ROM software, and Internet web browsers. Users of such new media expect to get a map tailored to their immediate requirement, containing just those objects relevant to the task in hand.

However, users have also been conditioned to expect a cartographic representation of those geographical objects in the form of a conventional map, and that can only be achieved by active representation responding to the unique requirements of this specific map request.

The application of object-oriented methods to the task of cartographic generalisation is now recognised as a powerful new paradigm for this task, as typified by the Gothic LAMPS2 Generaliser option. Object-oriented techniques for real-time generalisation are the next stage in the evolution of on-demand mapping. In the interim, until real-time generalisation becomes available, active representation using object-oriented display methods are the best approach to providing tailored mapping in response to user queries.

Conclusions

- Active representation through display methods is now available in a new generation of object-oriented mapping and GIS software typified by Gothic LAMPS2.
- Active representation is needed to solve the problems of multiple product generation from a single database
- Active representation will become increasingly important as the rise of the Internet and multimedia software stimulates the need for on-demand mapping.

References

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

[2] Laser-Scan Ltd. "The Gothic Versioned Object-Oriented Database: an Introduction". November 1994, Laser-Scan, Cambridge UK. See also up-to-date information on the Internet web pages at http://www.laser-scan.co.uk/.

[3] Woodsford, P. A. "Object Orientation, Cartographic Generalisation and Multi-Product Databases", ICA/ACI Conference Proceedings, September 1995, Barcelona, Spain.

[4] Woodsford, P. A. and Hardy P.G. "Databases for Cartography and Navigation", ICC Conference Proceedings, June 1997, Stockholm, Sweden.