

DATABASE-DRIVEN CARTOGRAPHY FROM A DIGITAL LANDSCAPE MODEL, WITH MULTIPLE REPRESENTATIONS AND HUMAN OVERRIDES

Paul Hardy, Marc-Olivier Briat, Cory Eicher, Thierry Kressmann
ESRI Redlands, California
phardy@esri.com

KEYWORDS: Multiple Representation, Database Cartography, DLM/DCM/DKM, Overrides, Generalization

ABSTRACT

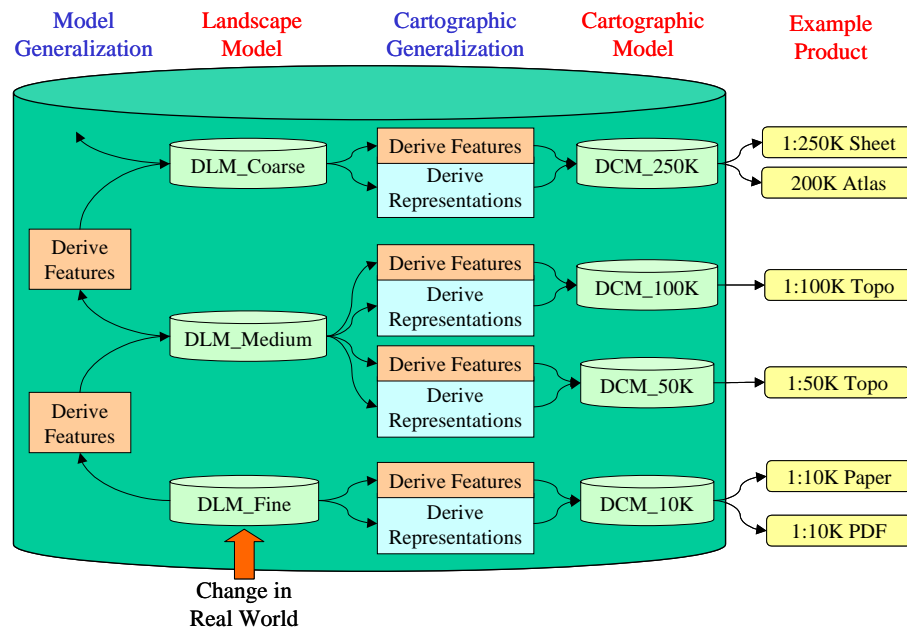
Many cartographic production organizations have a strategy of building a Digital Landscape Model (DLM) in a central database, and from that deriving a range of cartographic products. However, in the past, the cartographic presentation tools and symbolization mechanisms available in the GIS software that is used to build and maintain the DLM, have not provided the cartographic quality or the human freedom to produce directly the desired cartographic outputs. As a result, many organizations have had to fall back onto a divided workflow, where the cartographic finishing stages are done in publishing software dissociated from the central database.

This paper describes the rationale, scope, and technology of a project underway at present at ESRI to implement high-quality cartography with multiple representations in the database, including override mechanisms to empower the human cartographer to modify individual features and representation parts without affecting the master DLM data, and without unnecessary data duplication. This project is part of an ongoing strategic direction to deliver database-driven cartography capable of satisfying the demanding requirements of cartographic production organizations.

1 INTRODUCTION

1.1 Requirement for Multiple Representation

Many cartographic production organizations have a strategic goal of building a Digital Landscape Model (DLM) in a central database, and from that deriving a range of cartographic products. In practice, a single, scale-free DLM with 100% automated generalization is usually beyond current technology, so the preferred model has a small set of DLMs of increasing coarseness, and a set of associated Digital Cartographic Models (DCM or DKM) [AdV 1988]. Change data flows from the finest DLM through a set of derivation processes to the DCM and the ultimate product.



The subsystems that flow data between the models are mainly generalization processes - model generalization between the fine, medium and coarse landscape models, and cartographic generalization between the landscape model and its associated cartographic model. Many of the operations and algorithms needed are similar (selection, aggregation, simplification, etc), except that model generalization requires no displacement or exaggeration.

The principal rationale for this DLM/DCM approach is to avoid the expensive duplication of effort that currently occurs when manual updates have to be individually applied to each separately maintained and independent database for each commercial product (paper map or vector dataset). Another influence is the need for 'true-to-ground' (not cartographically generalized) data products for use in GIS and GPs-related systems. However, there is a trend towards hiding the mid-scale DLMs if there is not sufficient commercial benefit in shipping them as data products. An example of this is the shift in the German ATKIS scheme to provide only three resolutions of landscape model (DLM 25, 200, 1000) but five scales of cartographic model (DKM 25, 50, 250, 500, 1000).

This DLM/DCM model (and any simpler subset) often requires that one DLM feature be represented differently on different products, for reasons of scale or product specification, or of cartographic license by human judgment for maximum clarity. Clearly this can be achieved by copying the feature into one or more per-product and per-sheet DCM features in new and separate classes, but this unnecessarily multiplies entities and introduces divided workflows, which lead to problems in handling change through time.

1.2 Divided workflows

The traditional approach to higher-end cartography has been to use a GIS to generate and store the geography, as the DLM, but then transfer to a graphics package or a specific map finishing system, to carry out the representation and cartographic edits. This has strengths in using best-of-breed tools, but in most cases this is outweighed by the handicaps of having to export and import the data, the lack of WYSIWYG between the two halves, the lack of access to feature attributes during finishing, the need to carry out expensive remedial work when problems are found only at the finishing stage that then need to be reapplied to the GIS data, and the tremendous problems of handling incremental changes when a new edition is needed.

2 SOLUTION OVERVIEW

2.1 Database Cartography

At the heart of ESRI's cartographic solution is the ArcGIS software product family [ESRI 2004 & Ormsby 2001], with its geodatabase [ESRI 2004B & Zeiler 1999]. This provides a logical, structured framework to hold all the data for the DLM and for cartographic production, and the rules for its consistent presentation. The target is to provide a single, consistent, intuitive, and efficient environment that is used from the start to the end of the cartographic process, providing optimal tools and environment for the production cartographer, centered on the rigor of the geodatabase but allowing artistic freedom where needed.

2.2 Cartographic Layers

Although ESRI software is currently used by the majority of cartographic production organizations worldwide, particularly for creation and maintenance of the master GIS or DLM data, it has been recognized that better support is needed for production cartography. To that end, a research prototype was developed for 'cartographic layers' in ArcGIS, to store the representation rules and cartographic overrides in the database, to provide representation editing tools, and hence allow the complete cartographic production workflow to be carried out in a uniform environment.

The prototype was based on design concepts made available to ESRI when it acquired the French software company ALIDA and its DataDraw product [ESRI 2003]. This 'carto layers' prototype has been successfully demonstrated, and hence a formal project has been started to engineer similar capabilities into the ArcGIS core for a future release.

2.3 Database-driven representation rules

The first step is to store in the database the cartographic representation rules needed to create the required scale-specific product to the required cartographic specification. This rule storage mechanism needs to be backed by a high-quality representation drawing pipeline that can apply a set of geometry filters to derive the appropriate shape, and then symbolize that derived shape to achieve the cartographic visualization.

Examples of cartographic representation rules range from the simplistic:

- draw this stream line in blue at 1mm width,

to the complex:

- draw this cemetery area filled with a pattern of black tombstone cross symbols of 2mm size (unless religion attribute is Islam, in which case use crescent, or Jewish in which case star), with the grid aligned along the angle of the long axis of the graveyard, each symbol rotated to the same angle, suppressing symbols that would cross the boundary, and any that fall within any buildings in the area, superimposing the whole on a 50% grey background offset inwards from the boundary of the area by 2mm.

In many cases, these rules can be applied and re-applied automatically at draw-time, for maximum automation. However, there are other cases where the representation needs to be generated and stored. Examples are where the cartographic representation is derived not from a single DLM feature, but from a combination, such as embankments defined by upper and lower edge lines, or bridges with parapets and masking areas defined by the crossing of two road representations.

2.4 Overrides

Another reason for storing explicit representation information for a feature is that a human cartographer wishes to override the bulk rule for the feature class, to modify the representation of this particular feature. They may wish to displace it to avoid graphical conflict, to reshape it for elegance, or to change its graphic properties such as its color to stress its importance. All these actions are necessary to allow the map, atlas, or softcopy image to achieve clarity and quality, and hence to sell against other competitive cartographic products in the commercial marketplace.

2.5 Representation Editing

For those overrides that involve changing the shape or position of the representation, editing tools will be provided to allow the cartographer to manipulate the representation geometry rather than the GIS feature geometry. These allow the user to intercept either before or after the geometry filters, and to interact using intuitive tools modeled on those in desktop graphics packages. These representation edit tools support graphics concepts, such as Bezier curves with handles, and show the full representation while editing rather than a skeletal sketch.

2.6 Cartographic Features

The stored representation capabilities described above are in addition to the inherent ArcGIS and geodatabase facilities to derive, create, and store cartographic features as layers of points, lines, or polygons. Such cartographic feature classes are often vital to the implementation of digital cartographic models from digital landscape models and can act as a source for subsequent stored representations. Such cartographic features are often product-specific, and hence belong in the DCM rather than the DLM.

2.7 Generalization

Good cartography requires clean and appropriate geographic data as a starting point, and here the historic strengths of ESRI for geographic data manipulation come to the fore. ArcGIS 9 has a rich set of geoprocessing tools and a powerful geoprocessing environment complete with dialog and scripting based interfaces. It also has a visual model builder that simplifies the linking of geoprocessing tools into process flows that can select, filter, merge, and manipulate GIS features to create cartographic features in the DCM. The toolbox has a set of tools for generalization operations such as aggregation, collapsing and refinement. Future releases of ArcGIS will include additional generalization tools, particularly tools for contextual generalization. The associated ESRI paper [Lee 2004] submitted to this workshop describes the problems faced and some ongoing work in this area, while a new initiative is addressing contextual generalization including partitioning, displacement and adaptive constraint-based processing.

These geoprocessing generalization tools form the primary stage of an overall multistage framework for deriving and storing map products. The second stage is the geometry filters in the representation symbolization pipeline. These can carry out simple dynamic generalization on a feature-by-feature basis at draw time. If a particular series of filters are computationally expensive, then the result can be cached as a stored override geometry in the representation.

3 DATABASE STORAGE OF REPRESENTATIONS

Conceptually, representations can be stored in a representation table, linked to the GIS feature class by a relational join, or (more practically) can be added directly as extra columns to a GIS feature class. One row in the table can hold several related representations – useful where the cartographic visualization requires multiple components.

The representation information itself is held in two columns, one of which is a link to the symbolization rules to be applied. The other holds a structure, which can contain any overrides applied to this representation.

This structure is encapsulated and exposed to the user as a Representation Class, similar to the way that vector features are exposed as a Feature Class. They can be interrogated and edited much like features, and new cartographic editing tools are being developed, modeled on the intuitive style of desktop publishing.

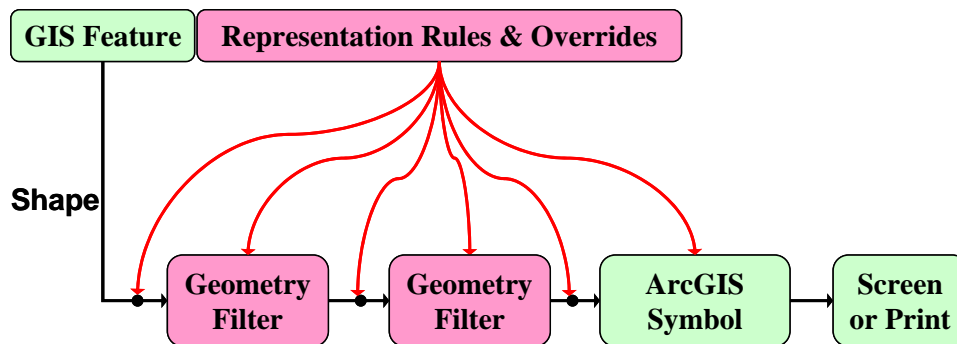
There is a central catalog table of representation classes, recording which representation table is involved for each

4 REPRESENTATION OVERRIDES

The starting point for display of a representation is to take the geometry from the GIS feature and feed it into the symbolization pipeline. This will apply any geometric filters, such as offsetting or smoothing a line, converting a polygon into a mesh of points, or generating specific cartographic patterns.

The resultant geometry is fed into the standard ArcGIS symbol routines, providing access to the vast range of markers, fills and line styles, before passing on to the graphics engine for rendering to screen or plot.

If the human cartographer wants to intervene, then they can intercept the geometry (either before or after the geometric filters), modify it, and store it in the representation field of the representation table as an override. They can similarly override the default attributes to any of the geometric filters, perhaps to smooth this feature more aggressively than usual. Finally, they can override any of the graphic attributes of the symbol stage, such as color, width, size, angle, dash pattern, marker symbol, etc.



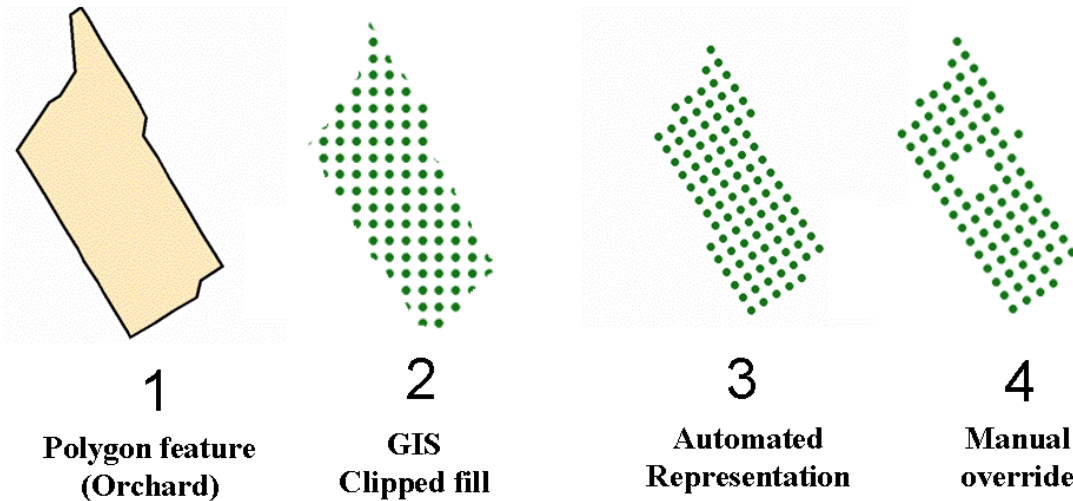
Usually such overrides are stored in a structure in the representation field, but the designer can opt to extract a particular override (e.g. angle) into a graphic attribute field, perhaps so that it can be used to override multiple aspects of the representation (angle of fill grid and angle of marker symbol).

5 EXAMPLES

The following examples show the stages in control of presentation as the representation facilities are deployed.

5.1 Orchard Example

The first example takes an orchard with its polygon extent (1), and shows it with a typical GIS symbol fill pattern (2). Note that parts of trees are shown where the fill is clipped to the polygon, and that the pattern is horizontal and vertical. The next stage (3) is where the cartographic symbolization rules have been applied, through geometry filters in the drawing pipeline. Note that we have whole trees, and that they are aligned in rows along the long axis of the orchard. The final stage (4) is where a human cartographer has decided to override the automated result, and has deleted or moved individual trees. They could equally at this time have overridden the color or symbol size.

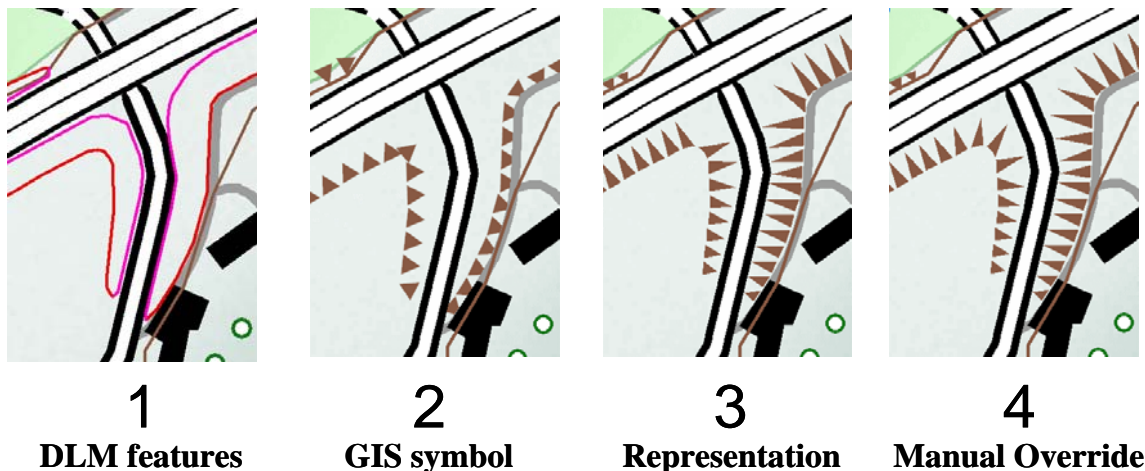


5.2 Embankment Example

The next example takes an embankment, modeled in the DLM as two line features, one at the top and one at the bottom of the slope, as shown in (1). In (2), it is shown with a standard GIS marker line symbolization. Note that the triangles are uniform. Note also that the orientation of the triangles is incorrect on the right-hand line, because the GIS features are not consistent in their order of digitizing (downhill is not always to the left).

In (3), the automated cartographic representation has been applied. Note that the triangles now indicate the size of that part of the embankment. Note however that at the sharp corners there is ambiguity as to whether the embankment is continuous or divided in two.

In (4), the human cartographer has intervened, and dragged the vertex of a few of the triangles, so as to give a clearer, more uniform and more attractive appearance at the sharp corners.



6 UPDATE

It is a fundamental design constraint of the representation mechanism that the system should handle the change through time of the DLM without losing or overwriting the efforts of the human cartographer on the derived products in the DCM. Much of this comes for free through the philosophy of not copying data unnecessarily.

If the shape of the orchard from the first example above changes in the DLM, then stage (3) of the process will trigger automatically and the fill angle and pattern will be recalculated. It is only if the operator had overridden the geometry as in (4) that there is a possible conflict, and mechanisms have been developed in the prototype to draw the attention of the operator to each such occurrence (probably only a few on any map), and ask whether the recalculated or the manual should win.

The update mechanisms have been designed to work in different organizational workflow cases. In a small organization, the person updating the DLM and the cartographer using the DCM may be the same person, and the consequences of DLM changes can be handled immediately or at end of session. In larger organizations, it may be a different department maintaining the DLM, whose operators do not want to know about DCM and cartographic products. In this case, it is possible to use delta information either from the version tree of the long transactions of the versioned enterprise geodatabase, or from one-way database replication, to ensure that the appropriate automatic processes are triggered in the DCM, and that remaining conflicts are traversed.

7 CONCLUSIONS

Advanced cartographic symbolization tools and mechanisms will soon become available in ArcGIS that will dramatically change the scope of database-driven cartography, and for the first time allow a map production agency to use the same uniform software from beginning to end of the cartographic production process, generating multiple products from a central database. The human freedom of the override system combined with the symbolization rule pipeline will permit high-quality, attractive cartography within a database-centered environment.

8 NOTES

1. The sample data used in the figures is swisstopo VECTOR25, copyright Swiss Federal Office of Topography.
2. This paper is a forward-looking document, and the capabilities it describes are still under development. As such, it is intended to give guidance as to likely future direction and should not be interpreted as a commitment by ESRI to provide precise capabilities in specific releases.

9 REFERENCES

- AdV 1988, Amtliches Topographisch - Kartographisches Informationssystem (ATKIS) - Das Vorhaben der Landesvermessungsverwaltungen zum Aufbau Digitaler Landschaftsmodelle und Digitaler Kartographischer Modelle , "Gelbes Heft", Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder (AdV), Bonn
- ESRI 2003, ESRI Press Release, "ESRI Acquires DataDraw Firm, ALIDA", February 5, 2003, http://www.esri.com/news/releases/03_1qtr/alida.html
- ESRI 2004, "What is ArcGIS", ESRI Press, ISBN 1-58948-090-2, 2004, http://store.esri.com/esri/showdetl.cfm?SID=2&Product_ID=109&Category_ID=28
- ESRI 2004B, "Building a Geodatabase", ESRI Press, 2004, http://store.esri.com/esri/showdetl.cfm?SID=2&Product_ID=90&Category_ID=28
- Lee 2004, "Geographic and Cartographic Contexts in Generalization", submitted to ICA Workshop on Generalisation and Multiple Representation, Leicester, UK, August 2004.
- Ormsby et al 2001, "Getting to know ArcGIS Desktop", ESRI Press, ISBN 1-87910-289-7, 2001, http://store.esri.com/esri/showdetl.cfm?SID=2&Product_ID=186&Category_ID=42
- Zeiler 1999, "Modeling Our World", ESRI Press, ISBN 1-879102-62-5, 1999, http://store.esri.com/esri/showdetl.cfm?SID=2&Product_ID=553&Category_ID=56

[Original 2004-05-14, Revised 2004-07-20]