CO-406

THE EVER IMPROVING GIS MAP: EVOLUTION OF CARTOGRAPHIC REPRESENTATIONS AND MAP PRODUCTION PROCESSES - LESSONS LEARNT FROM THE IGN TOPOGRAPHIC MAP PRODUCTION PROJECT

GUISLAIN P.(1), LECORDIX F.(2)

(1) GeoConcept, BAGNEUX, FRANCE ; (2) IGN - Institut Géographique National, SAINT-MANDÉ, FRANCE

CARTOGRAPHIC DATA AND SYMBOL MANAGEMENT IN THE GIS

The Demise of Cartographic Representation in Digital Maps

We can define the map as made of two different elements: one being the geographic data and the other the cartographic representation of this data. An ocean for instance, is defined in our data by the geometry of its coastline but the color shown on the map or the labels associated with it are not rooted in physical reality but merely the result of centuries old cartographic conventions. It is not so long ago that we got rid of mermaids and sea monsters in the marginalia.

Historically, when maps were drawn by hand, the cartographer was applying a cartographic representation model at the same time that he was surveying the landscape. We are still struggling to reenact that process with digital tools.

Buckley, Frye, & Buttenfield (2005) made clear the distinction between what they call the "geographic reality" and its cartographic representation. They also demonstrated how GIS initially focused on the first at the detriment of the second:

With the advent of GIS, data started to be compiled very differently. Over time, it became more important to capture features as more "exact" geographic representations [..] and GIS users were indisposed to displace, generalize or otherwise manipulate the data for cartographic purposes because any abstraction would compromise the utility of the data for geographic inventory and analysis. For many GIS users, the data were not far removed from "geographic reality"; for cartographers; however, they were representations of reality that required further abstraction in order for them to be appropriate for map making.

Dealing on the one hand with the needs of the many GIS users and on the other, with the expectations of much fewer cartographers, it does not come as a surprise that GISs were not initially conceived for managing cartographic data and rendering geographic features with the wealth of details and graphic options that manual processes or illustration software provide.

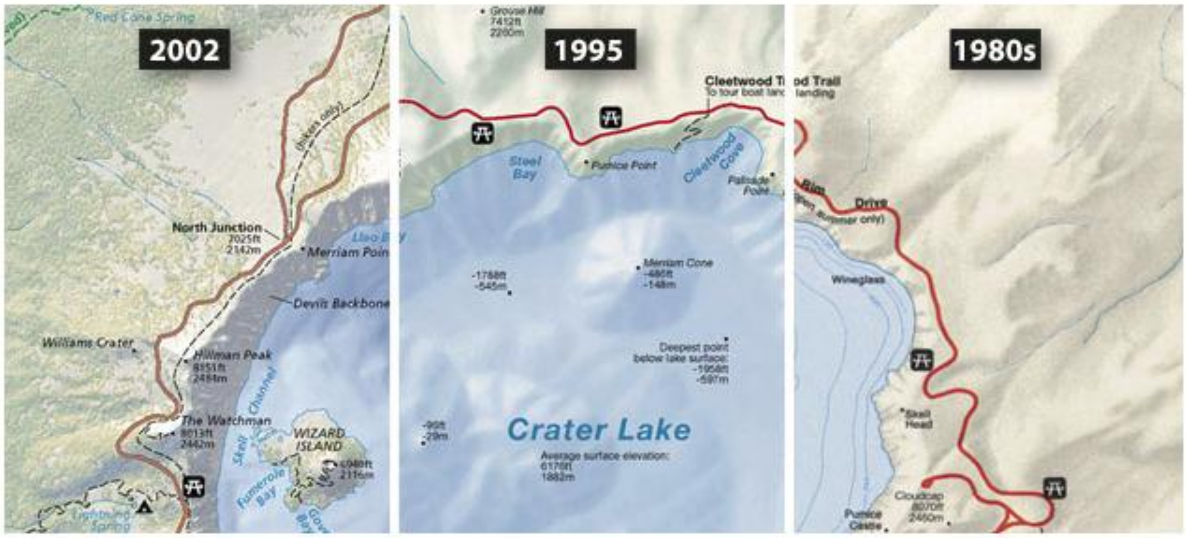
Closing the Illustrator Gap and Other Challenges

GIS publishers are however getting more aware of the need to enhance their products in order to output high quality maps. In our jargon it is referred to as "closing the Illustrator gap". In other words, provide sufficiently advanced graphic options and pre-press functions to spare the user the trouble of exporting the map in Adobe Illustrator or equivalent software for finishing her work and adding extra graphic effects.

GIS software is thus constantly evolving to take into account printing and online publishing requirements. It is only recently that it has been able to deal with CMYK color separation and directly produce EPS or PDF output. Spot colors for instance, are still the exception; we only introduced them in Publisher two years ago for IGN.

But closing the gap is a necessity since Illustrator does not have the capacity to easily handle the volume of data that we are dealing with. Another significant drawback is certainly that the work done outside of the GIS cannot be stored back in the database and will be lost when the data are updated. So, using Illustrator, or Photoshop or any external software in which you duplicate your data can be a practical solution if you are dealing with a small territory and a limited number of maps. But it is totally unmanageable if you are dealing with, for example, a large scale topographical map of France.

The next step is possibly "the Photoshop gap" which is about photo-realism, lighting effects and, complex textures. One notorious example of such photo-realism applied to cartography is the work of Tom Paterson of the National Parks Service. The illustration below (Paterson 2002) shows the evolution of the NPS map over two decades and Paterson's publications offer a wealth of information on the subject.



And if you wonder how, with your traditional mapping software, you will be able to mass-produce and print high-quality maps on paper, the traditional medium; think about the challenge that cartographers are now facing when it comes to publishing their work for what Cartwright names the "new media":

New Media includes a range of new delivery and display platforms; among them are the World Wide Web, interactive digital television, WAP technologies, interactive hyperlinked mapping services, and enhanced mapping packages that are "linked" to large databases – national of global (Cartwright 2002).

So whereas GIS software is still struggling to "close the Illustrator gap", it seems that new expectations are formed: we need no less than the ability to process automatically a large volume of data, render it using advanced cartographic representations and produce an output for a constantly evolving and ever richer media.

Defining and Managing Advanced Symbols with Cartographic Styles

Publisher is a cartographic rendering extension for the GIS GeoConcept. To better manage cartographic representations, it introduced the notion of cartographic styles. The name style was not chosen at random. Publisher styles are functionally equivalent to Microsoft Word styles. Just like the styles in Word are used to store all the properties defining the appearance of text paragraphs, cartographic styles are a simple and efficient way to bundle up the definition of a cartographic symbol.

The Publisher symbol paradigm is rather simple as only four different kinds of styles: text, (marker) symbol, line and, surface are defined but their rendition is based on a large number of parameters which are very much inspired by the options available in Illustrator. The options available are also very similar to those described in the OGC "Symbology Encoding Implementation Specification" and able to satisfy most of the drawing needs of the advanced GIS user.

The Illustrator inspiration has proven of great value since Illustrator users and cartographers - who are usually well versed in illustration software - do not face a very steep learning curve when trying to apply their knowledge to defining cartographic representations in the GIS. In fact, the acknowledged target was not only the "discerning" cartographer but also the geomarketing users who needed a simple way to visualize their data (Briat, 1999).

The versatile Publisher approach to managing the symbology is one of its most significant features: a coherent and simple set of styles organized in views, on which to easily build and manage the cartographic model. It clearly was a decisive factor in IGN's choice of Publisher for the topographic map project.

Cartographic Attributes and their Hierarchy

With software devoted to advanced cartographic representation, we are moving away from a traditional "black box" symbol concept in which the symbol parameters: color, line width, text size, etc; are defined at a generic level only – for a whole feature class for instance - and stored by the system in a proprietary format. One of the main drawbacks of the "black box" logic being that it has limited or no capabilities for customization at the feature level.

The Publisher approach to managing the Digital Cartographic Model (DCM) makes it straightforward to link cartographic styles and symbol parameters – which we refer to as cartographic attributes – to a class, a subtype or a subset of features identified by discrete values read from the database. Consequently it offers a flexible and hierarchical way of defining cartographic attributes that is well adapted to the management of the map display. At the lowest level with the overrides mechanism it lets the user set the values of cartographic attributes directly at the feature level.

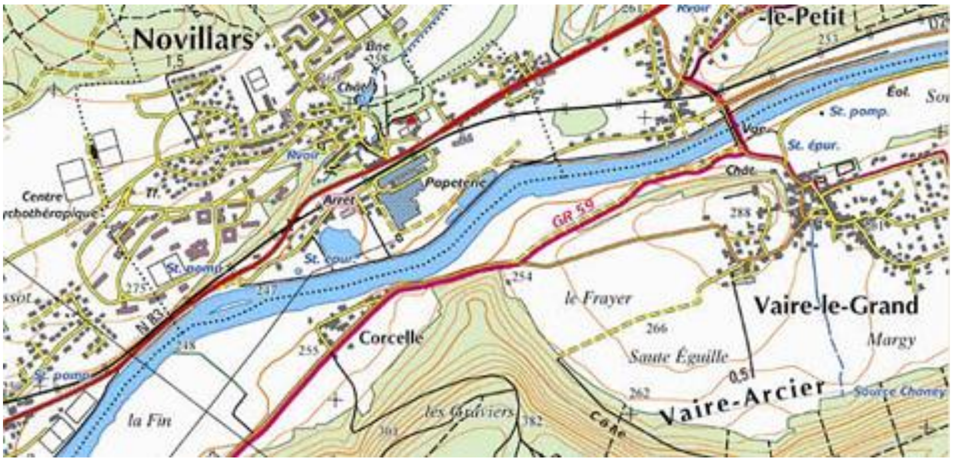
Data Driven Parameters - Overrides and Exceptions

An override is the ability to redefine the value of a cartographic parameter at a different hierarchical level. Typically a “style” will define the default value of a cartographic attribute at a generic level. This default aspect can then be redefined for a given subset of features: the representation class. Additionally, an override can then be set to link this attribute to a column in the database from which a specific value can be read for each feature.

By using the database to store feature specific values, the overrides mechanism provides the ability to manage exceptions i.e. to customize the representation of any feature simply by altering the data on which this representation is based. An exception is typically created by editing the data or managed with dedicated edition tools targeting specific kind of cartographic attributes.

The override logic and exception mechanism is extensively used in Publisher which heavily relies on symbols attributes read from the database to create rich cartographic artifacts and a seamless cartographic editing experience. A similar logic has been recently introduced in ArcGis (Eicher & Briat, 2005).

THE IGN TOPOGRAPHIC MAP PRODUCTION LINE



Context: Previous Production Methods

In 2007, IGN (Institut Géographique National) issued and RFP for the upgrade of its production lines for the 1:25,000 and 1:50,000 topographic maps.

There were at the time two different production workflows. One based on manual modifications of raster files, the other based on vector data in the proprietary GeoConcept format and stored in independent map files. DataDraw, an early version of Publisher, was used to manage the map DCM and provide cartographic editing and publishing functions.

Additional software developed by IGN or other software vendors were - and still are – used for generalization, text placement, hillshade generation and, the final output of the map for printing.

The “New Base Map” Project

The goal of the New Base Map (*Nouvelle carte de base*) project was to create a country-wide topographical map based essentially on the existing IGN database BDTopo® from which to extract and finalize the individual maps at different scales. The outcome of this project will be presented by IGN at this conference.

One of the requirements was also that the system would easily leverage the work that had already been done using the previous software with digital maps already covering 20 percent of the French territory.

The “Continuous Update” Project

The Continuous Update (*Mise à jour en continu* or *MAJEC*) project aimed at providing an update and information gathering mechanism for the constitution and maintenance of the base map.

The GIS was to be used to input and edit the features. The display was to be organized in a way that made it possible to visualize the map using different style sets or “views”. This way, the map shown on the screen would be as close to the final product as possible, but other views could easily be summoned for specific analysis and editing tasks.

The data collated in the GIS in different production sites would then be stored in a central database. A process aimed at integrating seamlessly the updates in the existing maps would ultimately be created based on this input.

Managing the IGN Topographic Map DCM: Requirements

A strong requirement for the new system was that it could manage the DCM using “cartographic styles” that would define all the parameters for a symbol and its association with the features it was to represent.

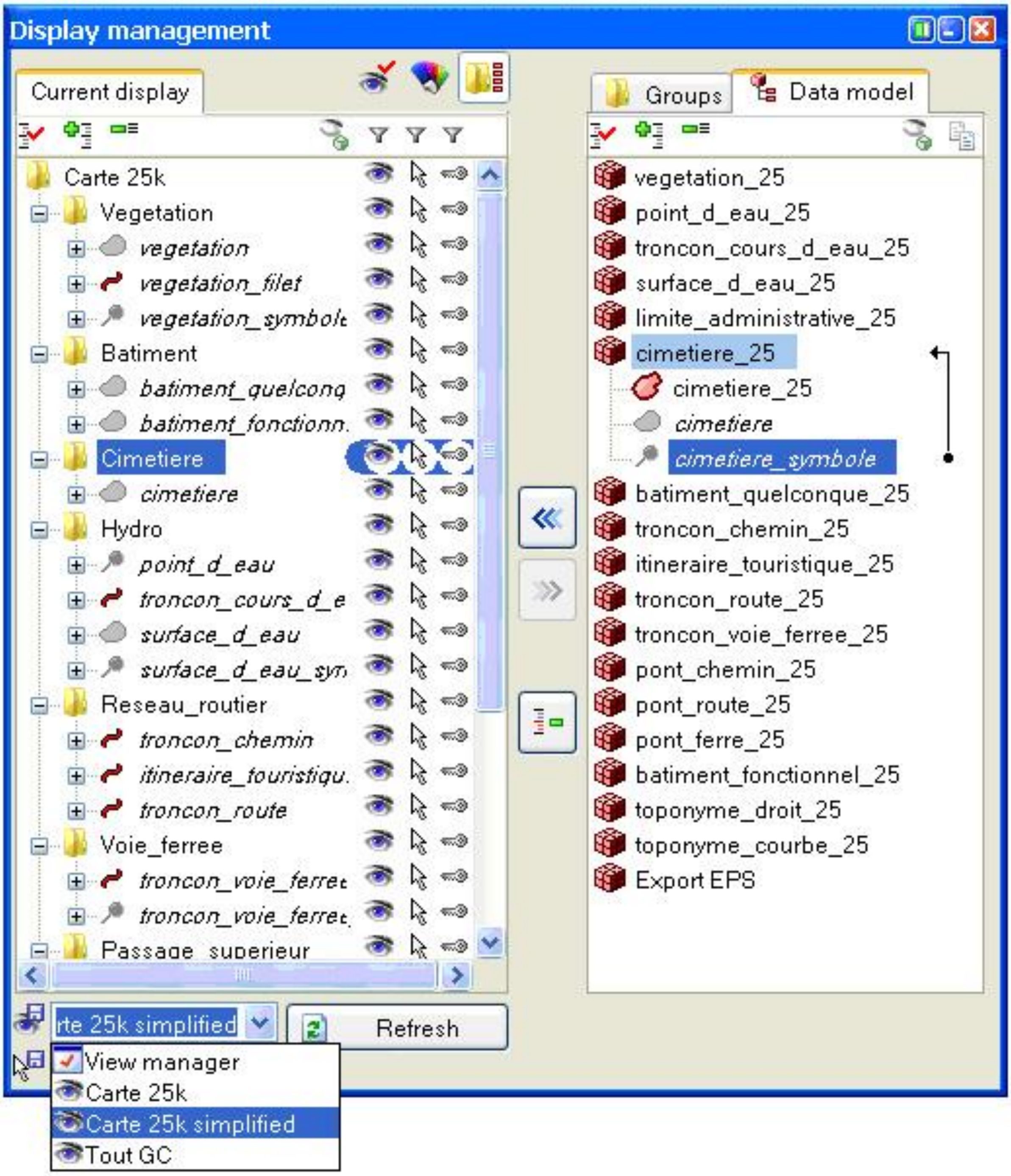
Other significant requirements were as follows:

- The colors would be defined as CMYK or spot colors.
- The styles would be editable through a GUI and described in xml files with the ability to export or import them from or into the map.
- Variations of a style could be defined based on a list of discrete values read from the database with the option of applying a default representation to empty or null values.
- The display would be based on a reference scale and the size of the symbols and texts automatically and precisely adjusted for any given scale.
- All the styles parameters would be cartographic attributes defined by either a global value at the style level or overridden by a value read from the database.
- The styles would be referenced in a “style library” from which they could be retrieved to be added to and organized in “views”.
- Multiple sets of styles – a.k.a. “views” - could be created, each defining a different display of the map. The styles in a view could be grouped hierarchically using folders since the topographical map DCM would typically comprise of a hundred or more styles.

Managing the IGN Topographic Map DCM: Functional Description

Display Management

In order to manage the topographic map DCM, GeoConcept had to come up with a complete redesign of the application Display Management user interface (UI). The new UI pictured below allows the cartographer to create the DCM by selecting the styles from a style repository where they are displayed in a hierarchical tree mirroring the data model (right pane) and organized freely in the left pane, with the possibility to group them so that related styles can be managed together in one click. The content of the current display can then be saved in what is referred to as views so that one can switch from one organization of the DCM to another in one click.

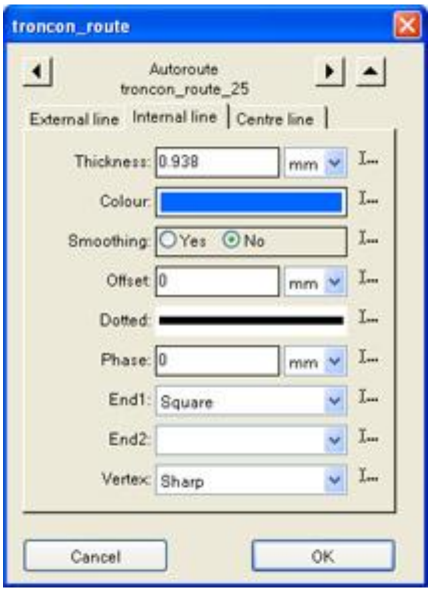


Our next task was to demonstrate that Publisher could be efficiently used to convert the requirements of the IGN into a functional DCM on the test data supplied. Reproduced below are a few extracts of the requirements and corresponding style parameters set with Publisher:

Road Network

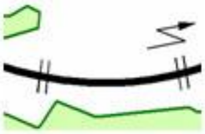
Representation Class	Casing	Filling	Centerline
Under construction	*Width* : 0.6 mm *Color* : BLACK *Dashes* : full 3 mm blank 1.2 mm	*Width* : 0.4 mm *Color* : WHITE *Dashes* : full 3 mm blank 1.2 mm	none
NC	*Width* : 0.512 mm *Color* : BLACK	*Width* : 0.312 mm *Color* : WHITE	none
Local 1	*Width* : 0.512 mm *Color* : BLACK	*Width* : 0.312 mm *Color* : YELLOW	none
Local 2	*Width* : 0.638 mm *Color* : BLACK	*Width* : 0. 438 mm *Color* : YELLOW	none
Regional 1	*Width* : 0.512 mm *Color* : BLACK	*Width* : 0.312 mm *Color* : ORANGE	none
Regional 2	*Width* : 0.638 mm *Color* : BLACK	*Width* : 0. 438 mm *Color* : ORANGE	none
Main 1	*Width* : 0.512 mm *Color* : BLACK	*Width* : 0.312 mm *Color* : RED	none
Main 2	*Width* : 0.638 mm *Color* : BLACK	*Width* : 0. 438 mm *Color* : RED	none
Ramp 2	*Width* : 0.638 mm *Color* : BLACK	*Width* : 0.438 mm *Color* : BLUE	none
Freeway	*Width* : 1.138 mm *Color* : BLACK	*Width* : 0. 938 mm *Color* : BLUE	*Width* : 0.187 mm *Color* : MAGENTA77

With the Line style, one can define at the same time several stacked line symbols in order to manage together the different elements of a road representation :



Railroad Network

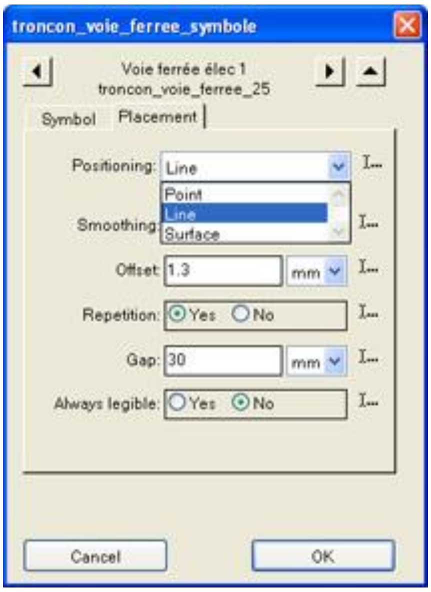
Line Style :

Representation Class	Casing	Filling	Centerline
Railroad elec 1 	*Width* : 0.312 mm *Color* : BLACK *Line ends* : Round *Vertices* : Round	*Width* : 1.4 mm *Color* : BLACK *Dashes* : full 0.08 mm blank 6.72 mm *Line ends* : Square *Vertices* : Round	None
Railroad elec 2 	*Width* : 0.312 mm *Color* : BLACK *Line ends* : Round *Vertices* : Round	*Width* : 1.4 mm *Color* : BLACK *Dashes* : full 0.08 mm blank 0.32 mm full 0.08 mm blank 6.72 mm *Line ends* : Square *Vertices* : Sharp	None
Sidetrack	*Width* : 0.1 mm *Color* : BLACK *Line ends* : Round *Vertices* : Round	None	None

Marker Symbol Style :

Representation Class	Marker	Linear Position
Railroad elec 1	*Name* : electric.eps *Color* : BLACK	*Offset from line* : 1.3 mm *Repetead along line* : oui *Repeatead every* : 30 mm
Railroad elec 2	*Name* : electric.eps *Color* : BLACK	*Offset from line* : 1.3 mm *Repetead along line* : oui *Repeatead every* : 30 mm
Sidetrack	none	none

The Marker Symbol style is used to define the positioning of markers along lines. The same UI will show different options when electing to display markers on a point or use them to fill a surface:

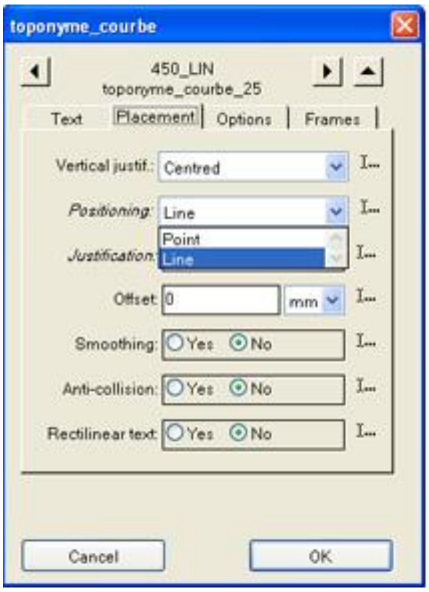


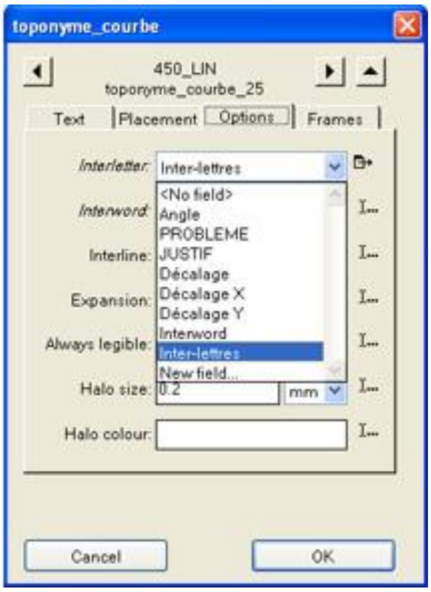
Curved Text

Curved text representation based on a class of line features.
Font : Optima Bold (TrueType).
Text value is read from database column *Ecriture*
Linear justification is based on column *Justif*, real number in the range [0 , 1].
Inter-letter is 0.4 mm for all representation classes.
Halo : 1 mm white.

Representation Class	Text Style	Text Size (mm)	Color
454_LIN	Bold Italic	1.88	BLUE
453_LIN	Bold Italic	2.227	BLUE
450_LIN	Bold Italic	3.55	BLUE
NOM_GR	Bold Italic	2.227	MAGENTA

Similarly, text can be positioned automatically along a line feature's geometry. The option to select a column from the database to override the default specification directly from the UI is shown on the right for the Inter-letter option:





COMPLEX SYMBOLS

This paragraph explores the use of complex cartographic representation based on multiple geometries and/or geometric transformations, complex exceptions or, the interaction of multiple features. The examples are based on the IGN projects (e.g. the football field) or on similarly intricate maps.

Geometric Transformations and Drawing Geometries

When the spatial dimension was added to the information system, the requirements were primarily to store the shape of landscape features, perform geometric computations and projections and, respond adequately to spatial requests. The successive models used for geometry storage in CAD then in coverage and, in the current systems are all about "real world geometry" and they are very good at it. However, additional geometries are needed for the purpose of cartographic representations.

In order to store these cartographic or drawing geometries, users tend to bend the rules. Within the CAD model, hash lines or texts for instance, were sometimes stored respectively as lines and polygons although they are clearly not geographic features.

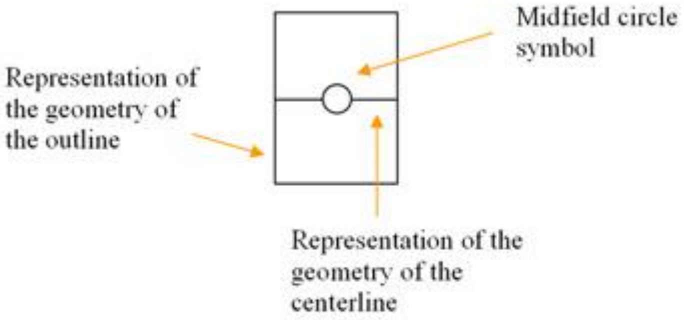
For the topographic map, Publisher implemented a "multi-geometry" model: the ability to create multiple geometry columns in a feature class.

Geometric transformations are used to derive a drawing geometry from a "real-word" geometry. This is done implicitly by most software, for instance for drawing hash lines or hachure in a polygon. Within a multi-geometry model these transformations can become explicit as it is possible to use dedicated geometric functions to transform the original geometry and draw the features using the resulting geometry.

In order to create new symbols in systems that used a more traditional approach, one would have at drawing time, read and convert the original feature geometry and create the graphic instructions needed to actually render the symbol. On the other hand, the multi-geometry can be pre-processed and stored in the database so that it does not need to be computed over and over again at drawing time and since it can be drawn using the existing “styles”, the programmer does not need to “reinvent the wheel” i.e. to also implement all the rendering instructions.

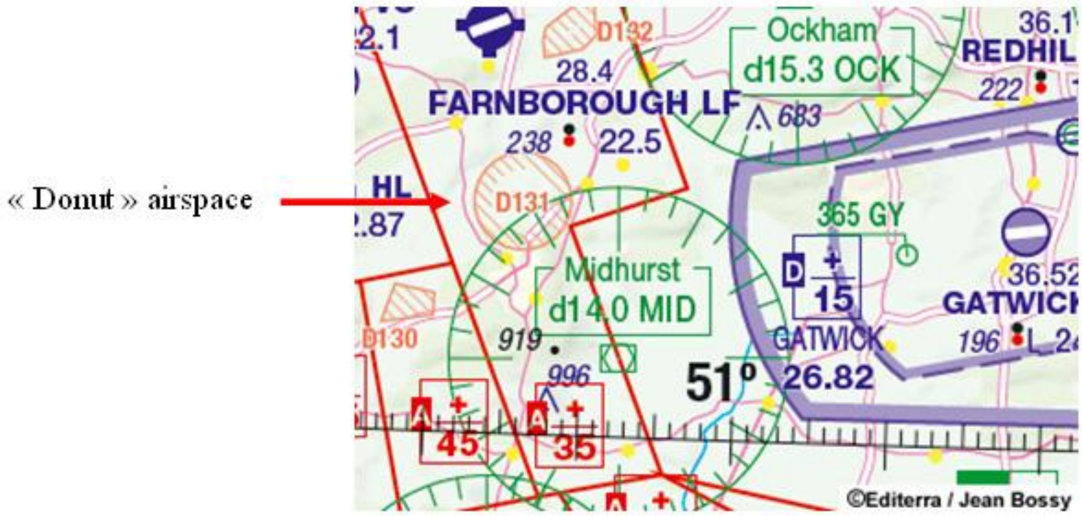
Examples of Cartographic Symbols Based on Multiple Geometries

IGN wanted to be able to draw football fields (as a polygon with a centerline a midfield circle and an outline). This implied managing three different geometries and of course editing them all in one click such as when rotating the whole thing.



Examples of Geometric Transformations

A typical example is the airspace on the aeronautical chart such as the restricted airspace D131 on the VFR map below which we will refer to henceforth as a “donut”. A common way to represent this feature is to duplicate the original geometry, and use a geometrical transformation to create the hole in the donut:



Complex Exceptions

A complex exception is one that cannot be managed by modifying the values of simple cartographic attributes. The need arises for instance when the user wishes to alter a symbol that the rendering mechanism manages as a whole and cannot normally dissociate in smaller components.

Examples of such instances include adjusting individual dashes in a dash line; editing individual markers in a marker line or, a marker fill.

The problem is of course to store the duplicate and to do so in a format that can be freely modified.

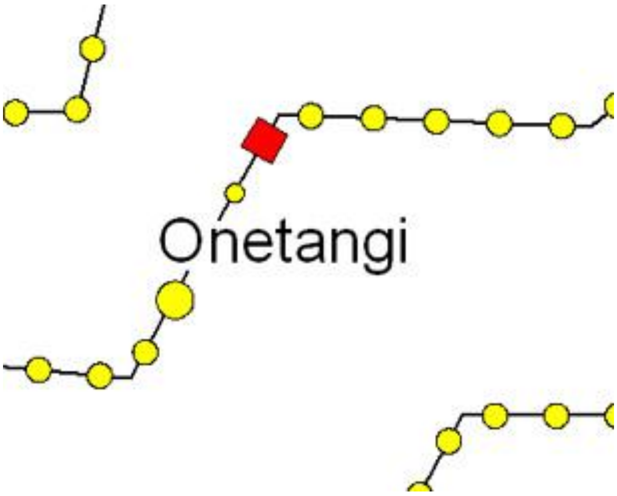
Many techniques can be used to achieve this result, for instance:

- Graphic elements (ArcGIS).
- Duplication of the feature in another feature class for geometric modifications.
- Storage in binary format of drawing primitives (ArcGIS Representation extension, DataDraw, Publisher).
- Conversion and storage of drawing primitives in dedicated feature classes (Publisher “graphic layers”).
- Multi-geometry (Publisher).

There are still many challenges in these approaches: keeping the link between the exception and the original feature, providing for a simple user experience by using standard editing tools for the edition of the exceptions, providing conversion functions and minimizing the manual steps.

Examples of Complex Symbolic Exceptions

In order to locally interrupt a dash line and modify individual markers in marker lines we tested converting the display into new objects in dedicated classes which we refer to as graphic layers:



Compound Symbols and Representations of Multiple Interacting Features

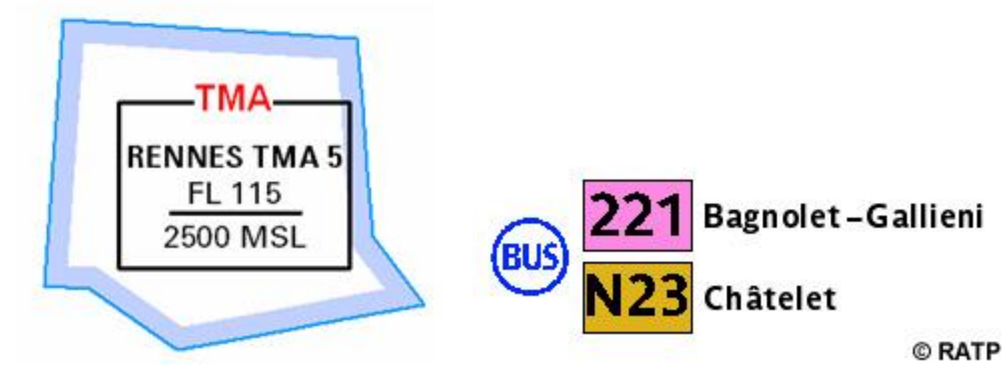
Compound symbols are based on the data stored in one single feature but the result is a symbol made of multiple parts which need to be arranged with a specific layout. It may for instance contain several text items and icons which will imply a symbol of adjustable size and relative placement of its constituents.

An even more challenging problem is to create cartographic symbols based on several features and their interaction.

As far as we know there is currently no generic solution for these kinds of representations.

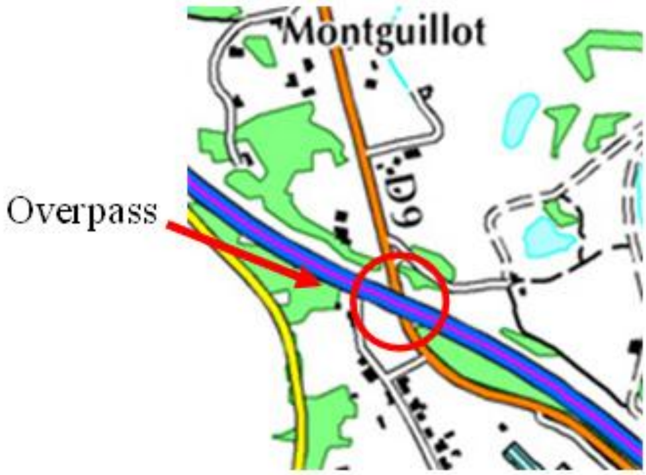
Examples of Compound Symbols

Aeronautical chart Navaid "boxes" and public transportation maps bus labels:



Symbols Based on the Interaction of Multiple Features

The overpass/underpass representations are a prominent example of such symbols. The complexity lies as much in creating the appropriate graphic output as in identifying the interactions between the features constituting the network:



AGENT BASED SYSTEMS FOR CARTOGRAPHIC RENDERING

Disclaimer

The present is only a research document and it does not constitute a description of existing functional abilities or a commitment to introduce new functions in future software releases.

References

Buckley, Frye, & Buttenfield (2005). “An Information Model for Maps: Towards Cartographic Production from GIS Databases”. Proceedings, ICA 2005.

Patterson, 2002, “Getting Real: Reflecting on the New Look of National Park Service Maps”. Proceedings, ICA 2002.

Open Geospatial Consortium Inc., 2006, “Symbology Encoding Implementation Specification” - OGC 05-077r4.

Cartwright, W. E., 2005, “Linking geographical facts with cartographic artifacts”, Cybercartography, Taylor, D. R. F. (ed.), Elsevier Science Ltd., pp. 331 – 338.

Briat, 1999, “DataDraw : a modeling platform for cartographic problems on top of a database”, in “Computational Cartography – Cartography meets Computational Geometry”, pp. 10-44.

E. Maugeais, F. Lecordix, X. Halbecq, A. Braun : Dérivation cartographique multi échelles de la BDTOPO de l’IGN France : Mise en oeuvre du processus de production de la nouvelle Carte de Base, 25th ICC, Paris, 2011.

Ferber J. “Introduction aux concepts et méthodologies de conception multi-agents” in :Amblard F. Phan D. eds.(2006) Modélisation et simulation multi-agents pour les Sciences de l'Homme et de la Société : une introduction, Londres, Hermes-Sciences, 414 p. ISBN : 2-7462-1310-9. chapitre 1, p.11-36, 2006.

Lamy, Ruas, Demazeau, Jackson, Mackaness, Weibel, 1999, “The Application of Agents in Automated Map Generalisation.” Proceedings ICA 1999.

Monnot, Hardy & Lee, 2006, “An optimization approach to constraint-based generalization in a commodity GIS framework”. Proceedings ICA 2006.

Monnot, Hardy & Lee, 2007, “Topological constraints, actions and reflexes for generalization by optimization”. Proceedings ICA 2007.

Eicher, Briat, “Supporting Interactive Editing of Cartographic Representations in GIS Software” . Proceedings ICC 2005.

Wooldridge, M. J. and Jennings, N. R. (1998) “Pitfalls of Agent-Oriented Development.” In: 2nd Int. Conf. on Autonomous Agents (Agents 98), Minneapolis, USA. pp. 385-391.

FIPA - Foundation for Intelligent Physical Agents, “FIPA Abstract Architecture Specification”, SC00001L, FIPA 2002, www.fipa.org.

For complex maps such as the topographic map, the public transportation map or the aeronautical chart from which a number of examples are taken; mapping agencies typically rely on an array of different products, custom software and scripts which require significant investments. The resulting production line may take years to implement and its evolution and maintenance is a constant concern.

Today a typical map production line would be based on the following tools and techniques:

Data and geometry processing using scripts. The cartographer today needs to be well versed in python, SQL, VB and other scripting languages to fully exploit her favorite RDBMS and associated GIS software.

The definition of symbols and the rules which dictates their association with the features in the map in other words, the DCM.

Processes and functions designed for the production of complex or industry-specific symbols.

Specialized workflows for generalization, labeling, text and symbol placement.

Architectural Limitations

Complex symbols, labeling, placement and, generalization functions seem to be quite an afterthought in the typical GIS suite. For that reason, the architecture of the GIS can be at times ill-suited for integrating these processes in its workflow.

The theoretical map production line described above is articulated around several independent "loops". On our "assembly line", different "loops" or processes will in turn take as their input the output of the previous "loop", churn out a result and pass it on down the line.

The GIS drawing loop is always the final stage in the process.

This final loop is crucial because it is the one that is executed most often. So, it would be a good thing to do as little computation as possible at drawing time. Unfortunately, in our GISs, it is only at this stage when the system is drawing the features, that some information relating to symbol footprint or overlapping can be obtained. Before the drawing takes place, the system is "blind". Consequently the programmers are trying to pack a lot of operations at the moment the features are being drawn. Of course, this adversely affects the overall performance and responsiveness of the system.

The following scenarios would be a major headache for the system:

The source data of an annotation feature class is updated.

The user moves a feature and expects all related annotations to be dynamically repositioned.

The user adds a feature and that should affect the geometry of the "donut". This purely hypothetical representation is shown below with the altered geometry of the hole shown as a gray dotted line.

Organizing Complex Cartographic Processes

Complex processes can be managed more easily as a series of discrete and simple steps and that those steps should be made far upstream of the GIS drawing loop.

Complex representations can be conceived as the result of a series of data modifications, geometric transformations and, drawing operations organized in a systemic way and taking into account features interactions. This may not only significantly expand the drawing capabilities of the GIS but also turn it into a dynamic rendering system in which data updates and user edits would be easier to handle.

Faced with the task of displaying a large number of symbols competing for space in the map according to complex representation rules, the GIS is relying more and more on the abilities of agent based systems to manage the interactions and constraints of multiple entities. In other words, to turn chaos into order.

The agent approach has been well described and successfully applied to generalization. For instance: in the AGENT project (Lamy, Ruas, Demazeau, Jackson, Mackaness, Weibel 1999) or more recently with the "Optimizer" constraint-based generalization project (Monnot, Hardy & Lee, 2006). Various implementations of text placement programs can also be considered as agent-based systems in their own right.

Multi-Agent System for Cartographic Processing and Rendering

Multi-Agent System (MAS) seem a logical way to manage complex cartographic representations and especially the drawing of symbols based on the interaction of multiple features. An hypothetical agent based architecture for that purpose would typically be founded on the following elements:

A system in charge of managing collections of agent groups, processes to execute and actions defining the functions that the processes can use to perform their tasks.

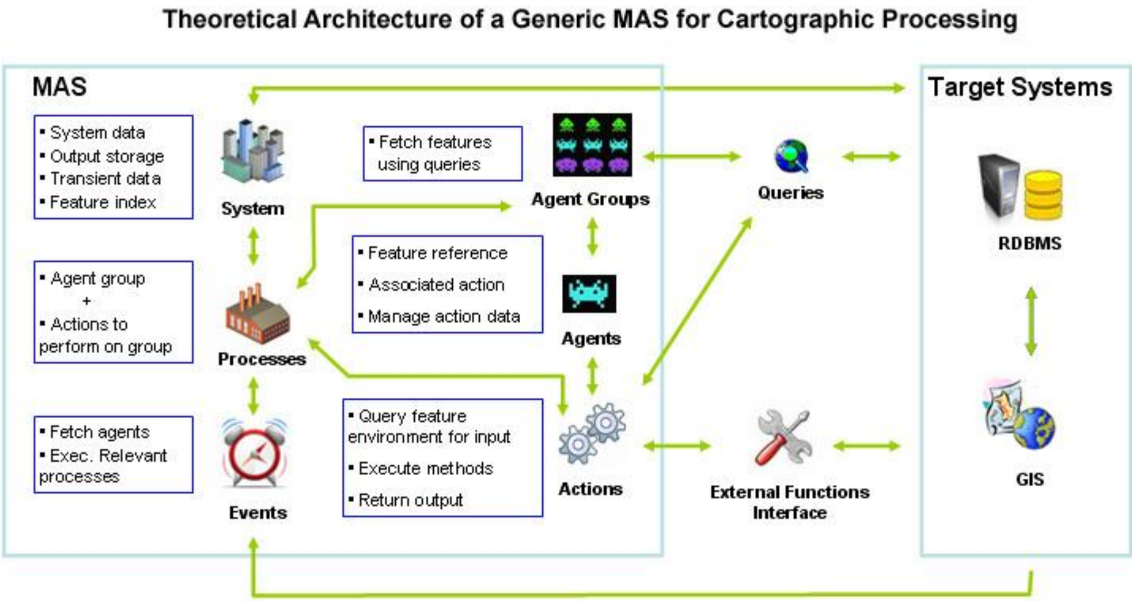
An agent need only be an extension of the actual GIS features. Its purpose is to give the feature the ability to query its environment to get some input, execute some function based on that input and eventually produce and store a result.

This ability is defined by an action. An action holds together the definition of a query used by the feature to get some environmental input and the definition of how the source feature or the features fetched by the query will be affected. The action provides access to an external function and defines how to communicate with it.

Agent groups are introduced with the aim of managing groups of features on which to apply a process. Agents groups can be ordered and organized hierarchically so as to precisely define which actions will be applied to different groups of features and in which order.

The process holds together a set of actions and relates them to a set of organized agent groups. In so doing, it defines the ordered execution of multiple actions on the agents contained in the agent groups.

Indexation and event handling to instantly retrieve and reactively execute the processes that are relevant for the features being edited or added by the user.



Such a system being only loosely coupled with the GIS – it could theoretically be connected only to a spatially enabled RDBMS and exploit functions from different GIS software or from remote locations through web services – it lets us envision a future in which instead of instilling small doses of agent based processing in existing GIS we could instead manage the GIS from a fully autonomous MAS dedicated to the processing and representation of cartographic data and related map productions tasks .