System for dynamic visualisation of choropleth maps

Lucie FRIEDMANNOVA
Laboratory on Geoinformatics and Cartography, Department of Geography, Faculty of Science, Masaryk University in Brno, Kotlarska 2, 611 37 BRNO, Czech Republic
E-mail: lucie@porthos.geogr.muni.cz

Karel STANEK
Laboratory on Geoinformatics and Cartography, Department of Geography, Faculty of Science, Masaryk University in Brno, Kotlarska 2, 611 37 BRNO, Czech Republic
E-mail: karst@porthos.geogr.muni.cz

1. Introduction

Choropleth maps belongs to most used type of thematic cartographic visualization in practice. Is very natural to visualize thematic data in relation with some predefined division of an observed area. From graphical variables point of view are choropleth maps very simple, key role play fill colors or hachure of defined areal units. Graphical representation of the unit borders is here only auxiliary feature for improving legibility. Units also play key role for localization and another localization features are often omitted. Creation and usage of choropleth maps looks relative easy, without surprise, this part of thematic mapping is most developed cartographic method in desktop GIS environment. But if we look closer on these implementations we can remark, that support for real automated choropleth map design is rather weak. We can to use some of implemented classification methods, to define unlimited number of classes, to use predefined color scales, to make scale between two limit colors, make color scale or hachure scale manually. Generalization is supported only indirectly, through scale limits for visualization of some group of features, so we can use different data at different zoom level. Of course purpose of desktop GIS is different then map design, but very often just thematic mapping is its main exploitation. In our lab processed project CASTOR is focused on automated thematic mapping from geodatabase. Objectives of project are to make certain equivalent of desktop GIS for cartographic purposes. In this paper presented work is part of above mentioned project.

2.System principles

System presented in this paper one of applications based on above mentioned project, just focused on choropleth maps. We try to make an “intelligent” designer, for electronic form of this type of map. We establish this application on following assumptions :

- geometry and attributes are stored in database
- map display and interface is based on common WWW browser
- color handling is automated
- maps are scalable in cartographic sense

We chosen WWW environment according to topical reality in computer usage and according to previous experiences with cartographic web publication. Above all, the SVG format and availability of its renderers for WWW browsers, make such type of carographic publication easier. Methods for application we adopt from CASTOR project.

These methods are especially related to color handling and generalization approaches, which we consider as key issues for this application. However, one of main inputs from CASTOR project is MapDL.

Map definition language (MapDL) is our proposal of an parametric description of map visualization process. This description we call metamap, because of it is something behind visible map. In addition we don't describe map compilation in its static form, but in dynamics, with changing scale, extent, content and purpose. Parametric description consequently cover map features, their graphic representations and their behaviour under above mentioned conditions. Parametric means, that description not include detailed process description, but only necessary keywords and parameters. Process self, i.e. map compilation, is provided by appropriate engine, which is driven MapDL In metamap definition we make allowances for compile map in both forms, analogue and electronic. This condition mean enrichment of classic set of map features by features, which ensure user communication with map in electronic form – controls. Top level of metamap object hierarchy is composed from following elements :

- map layout : tree of boxes in which are map features displayed, root box is map page
- list of map features : include base and complementary features such as map face, map legend, map scale, north arrow, title or various types of documents
- map space : consist from definition of cartoelements. Cartoelements are all possible visualized phenomena. Are organized in groups, which share identical behavior and symbols. Their geometry is an event over geometry stored in geodatabase. Their description include semantics, significancy, topology, symbolization parameters, generalization parameters and hyperlinks
- list of controls : controls are bind to boxes, map features elements, map space elements and cursor
- list of graphic symbols and fonts : their definitions are stored in database

For creation of formal metamap definition, i.e. MapDL, we use an XML. The XML is now very often used for such type of tasks, include digital cartography and GIS area. In our experimental engine all data exchange run over the XML and final output is also in XML based SVG format. Above all, we use flexibility of the XML, which allow easy extension, for evolutionary development of MapDL, according to practice. In the application we use only subset of MapDL, which is suitable for definition of a choropleth metamap.

We notice, that key parts of the application are color handling and generalization. In color handling we use following hierarchy :

- analogy : if it is possible, to classify visualized theme into group of themes with common characteristics and to use usual visualization method of the group representative. Based on analysis of map design in history we create metadata system, which support this way of color handling.
- positive/neutral/negative theme values sorting
- number of clases is related to number of used color tones
- color scales are described by color metrics, which ensure appropriate definition of distinguishable color steps within tone range

Color metric is based on the PANTONE sampler and colors are described trough sampler position and CMYK code. Color scale skeletons are predefined and stored in database.

Generalization of the choropleth maps consist from aggregation of units and enhancement of their border shape. Key issue is interaction with localization layer. During generalization we keep following rules :

- links between unit borders and localization features are established and fixed on both sides
- scales of conceptual changes are indicated (i.e. scales of change of an aggregation level)
- condition for structural generalization (i.e. change of border shape and symbolic) between conceptual steps are provided
- appointing of methods for border shape enhancement (i.e. segmentation, caricature and simplification)

Conditions of structural generalization are driven by cartometric measurements, which are related to map face content change. Cartoelements have basic support for different types of iterators over their structure and record of changes. Structural generalization methods are managed separate from cartoelements.

3.Implementation

In our previous attempts to construction electronic maps in WWW environment we use different technologies. From static image maps generated in GIS, static CGM drawings made in the same way, combination of PHP and GD library over MySql and SVG generation of SVG code from MySQL. In this project we use different approach, which we consider more flexible the previous one. We keep usage of relation database engine MySQL, but despite former methods, when we storing closed strings for polygon representation, we store arc segments constituted planar graph of cell boundaries inside of processed area. In relation database we store only arcs, cross points and points. Cross points contain arc topology and quad tree index is made over arcs MBR. Higher level features are stored as objects in simple object repository outside of MySQL. These complex objects are defined and handled by interpret of object oriented language Ruby. This OO language have lot of advantages for development of such type of application – rich string and database classes, an XML handling class, easy syntax and, what is important, interpret could be an Apache module. Ruby allows, relatively easy, to implement efficient object oriented model of map. Unlike PHP is in Ruby needed to generate whole HTML output, but with combination JavaScript, CSS and SVG attributes is possible to minimize this disadvantage (frankly, complete different story is reliability of unique browsers). Above mentioned sentences imply technological description of system. On the client side is web browser which processing html page defined by CSS, SVG and JavaScript (JavaScript ensure user interaction). On the server side is Apache web server with CGI application in Ruby language with dual persistent data management based on MySQL and file object repository.

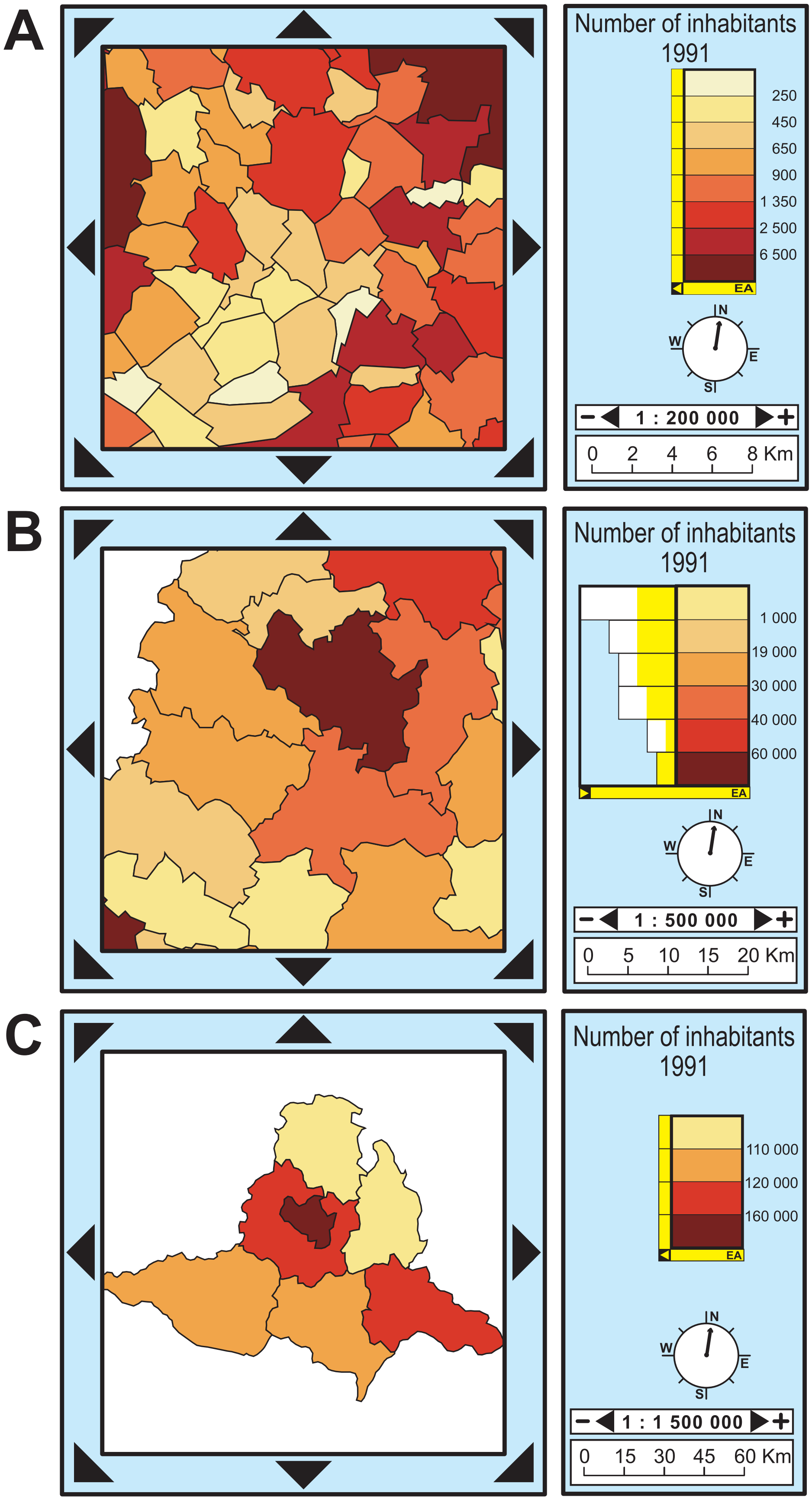
A
Number of inhabitants
1991
250
450
650
900
1 350
2 500
6 500
EA
N
W
E
S
1 : 200 000
0 2 4 6 8 Km
B
Number of inhabitants
1991
1 000
19 000
30 000
40 000
60 000
EA
N
W
E
S
1 : 500 000
0 5 10 15 20 Km
C
Number of inhabitants
1991
110 000
120 000
160 000
EA
N
W
E
S
1 : 1 500 000
0 15 30 45 60 Km

Fig.1: Examples of map layouts generated by the application in different scales

In presented example (Fig.1) we use very simple map layout consist from map face box and legend box. Map face box contains beside map face self, also margin with pan arrows. Legend Box consist from map name, value scale with histogram and classification method acronym, map bearing, map scale handler and graphical map scale. Graphic is defined in SVG format and generated trough CGI engine which process MapDL definition of map. Units are defined by administrative division, which also define conceptual generalizations level. Themes are different demographic characteristics of observed area (region South Moravia in Czech Republic). Aggregation of units is followed by reclassification of values. For caricature is used only keeping. Segments are defined by above mentioned quasi rectangular points, localization layer link points and arc cluster border points. Simplification of segments is provided by edgebuffer algorithm. As localization layer are used either centroids of settlements or river streams. Text are realized only temporary in relationship with cursor position.

From practice point of view, reasonable results of rendering are provided by Adobe SVGViewer and Batik. In any case, SVG looks like technology, which can bring maps in analogue like quality in WWW environment.

Conclusions

XML based technologies offer suitable frame for creation of dynamic maps in electronic environment. We try to extend our system for generating of choropleth map to managing all possible cases in this area. We are focused on amalgamation issues, improving of caricature, annotations and coexistence of several themes. Also more complex layouts management is one of extension directions. Practice with this application is useful for development of whole CASTOR project. Work on the application are related to grants GACR no. 205/00/D019 and 205/01/P133.

References

Bertin J.(1974): Graphische Semiologie, Walter de Gruyter, Berlin, 430s.

Friedmannová, L., Staněk, K.: Desktop GIS and Cartographic Visualisation. In GIS: Information Infrastructures and Interoperability for the 21th Century Information Society - Proceedings. Brno (Czech Republik) : ed.: M.Konečný, 1998., s. B3-29-5. sv.II.

Friedmannová, L., Staněk, K.: Publikace kartografických výstupů na WWW. In GIS Ostrava 99 - sborník referátů. Ostrava (Czech Republic) : ed.: P. Rapant, 1999., s. 212-215.

Kraak, M.J., Ormeling, F.J.: Cartography, Longman, 1996, 222s.

Lammens, J.M.G.: A Computational Model of Color Perception and Color Naming, Disertační práce, State University of New York at Buffalo, 1994, 253s

Lauermann L.(1978): Technická kartografie, Brno , 665p (In Czech)

Moellering H.(1983): Designing interactive cartographic systems using the concepts of real and virtual maps, Proceedings AUTO-CARTO VI, sv.2, str.53-64.

Staněk K.(2000): Zjednodušování a zhlazování liniových prvků v automatizované kartografické generalizaci, disertační práce, MU Brno,129s (In Czech)