SVG BASED "SMART" THEMATIC MAPS DESIGN

Friedmannova, L., Konecny, M. and Stanek, K.

Laboratory on Geoinformatics and Cartography, Dept. of Geography, Masaryk University
Kotlarska 2, 611 37 Brno, Czech Republic.
E-mail: lucie@geogr.muni.cz , konecny@ics.muni.cz and karst@geogr.muni.cz

ABSTRACT

This article is devoted to description of ongoing development of basic framework for design so called "smart" maps at our laboratory. Basically is focused on implementation of cartographic models in one particular interactive digital graphic environment - SVG. Our contemporary approaches and experiences in this are are discussed.

1. INTRODUCTION

Our laboratory is, besides another activities, focused on thematic cartography in digital environment. One of contemporary task of the digital cartography is design so called "smart maps". Meaning of this term is usually electronic map with some degree of the user interactivity.

In previous several years we try to develop environment for creation electronic maps which complete following list of conditions:

- be able to handling scales (cartographic (or adaptive) zoom)
- be based on spatial database of geometric features
- to have flexible and cartographically correct symbol management
- to enable spatial and attribute querying
- to have a simple exploratory and spatial analysis support
- to be OS independent

We tried to find some simple environment which enables easy development and multiplatform design. Finally we choose like an environment for creation of "smart" maps graphic language SVG.

2. TECHNOLOGY

2.1 The SVG language

Besides another standards of W3 consortium appeared in 2001 first committed proposal for vector graphic. The SVG is based on XML metalanguage, a common background for all new W3C standards, and in its definition is contained lot of useful capabilities which make it suitable for creation of the smart maps. How was mentioned before the SVG is vector graphic language, in addition exist possibility to embed and manipulate raster elements. Besides traditional abilities, which are contained in another vector graphic languages, SVG offer filtering abilities which partially erasing visual enhancement differences between raster and vector graphic. This esthetic improvement is not main purpose for usage of this technology. More important role play fact that the SVG is open language based on W3C standards. Open means that is extensible by additional features. That feature could be standardized - example of such extensions are usage of SMILE for animations of the graphical objects, or proprietary such like our repository of geometric objects. Like in many another W3C technologies is supported DOM concept - every element in SVG document is addressable and could be dynamically changed, deleted or is possible to add new elements. Every SVG renderer is usually equipped with an ECMAScript (proper name for JavaScript) engine which can manipulates DOM and graphical elements could start scripts through various listeners. These abilities offer possibility to design interactive electronic maps which have unified definition of the user interface and data visualization.

2.2 Rendering engines for the SVG

Adobe SVGViewer is probably most used renderer. Is poly-platform but less trouble is in MS Windows environment. According to Adobe role in definition of SVG standard and high compatibility of this viewer with graphic made in another Adobe software gained some dominance and Adobe proprietary extension of the DOM also influencing another renderer creators. During our development we used mainly this engine, but in last year we are more focused on the Batik engine. The Batik engine is an open source project entirely written in the JAVA language environment in close cooperation with W3C SVG working group.

3. BASIC CONCEPTS

3.1 User interface

According use interface issue we finally decided to create pure SVG based smart map. It means that not only map face but all visual controls are realized in the SVG. This is allowed by possibility to attach program structures to graphical elements which are, through DOM, control another elements included in document. Because of controls homogeneity we lock all default controls which are attached to engines.

We create set of ECMAScript routines which cover following control tasks :

- Adaptive zoom
- Pan
- Named views display
- Rotation of the map face
- Enhancement of the legend feature
- Adaptive legend update
- Map face features statistics
- Link among map face and legend features
- Query on features with location or selection results
- Single feature or user defined geometry based query
- Feature info window
- On demand features labeling
- length and area measurement (use defined geometry, distance of two selected features, single feature)

Map document from visual point of view is composition of following units:

- Map face
- Map frame
- Legend
- Map name
- Orientation symbol
- Graphical scale
- Control elements

Above mentioned list keep usual cartographic conventions, from this point of view our design of the smart map is quite close to classic map.

Besides control graphic elements are used some simple routines attached to map face and legend features. From above mentioned functions could be mentioned labeling and map face legend interactivity. User interface is still very basic and need to be more elaborated.

3.2 Geometry database

Geometry is stored in hierarchical structure of data records. At the base level are line segments. On higher level are areal and line features represented by list of segments and point features. Third level are point features defined by percentage of line features. Highest level represent feature groups which represent pattern features for generalization purposes. That groups are created by combination of geometric and semantic attributes and are object of filtering, amalgamation and typification. Simplification of the line segments is made by preprocessing which create to each vertex index of minimal validity. Various simplification algorithms are tested. Smoothing is not built in data model, fake smoothing is provided in visualization stage by round captions. For support of spatial queries, which are used for definition of the map face, map face features statistics, adaptive legend and selections, are all geometric features equipped by quad-tree index number. This index makes easier DOM manipulation with elements through getElemntById method, nevertheless is necessary to say that all geometric database elements are from beginning of map manipulation in the main memory. Database is implemented like embedded XML namespace in SVG document. In our initial intentions was to define whole smart map like one electronic document. External database can dramatically improved performance of smart map operations, but we prefer simplicity of solution. This simple one document solution brings some limits in size of map which is possible to cross only with external database engine solution.

3.3 Adaptive zoom

As was mentioned before adaptive or cartographic zoom is one of main features our smart map. Basically all map face elements are generated by query scripts and added to the DOM structure on demand and of course deleted after another action which changes map face parameters. Nowadays is map face completely generated after any pan or zoom action. Cache techniques are still in development.

All generalization procedures are precomputed and stored in data model. In this time following generalization operators are supported :

- simplification of line segments
- scale index of vertices
- aggregation (attached elements, which belongs to one new defined class, are transformed in one element)
- redundant polygon (i.e. list of line segments)
- amalgamation
- redundant polygon and auxiliary line segments
- filtering (geometric or attribute based)
- scale attributes of feature
- typification
- redundant definition of whole substitution pattern
- resymbolization
- collapse
- redundant point element and redundant surrounding polygon definition

Generalization process is based on defined scale sequence which infers application of generalization operators on various features. Generalization is related to whole scope of features on map face, is not adapted to various combination of feature sets.

3.4 Symbols and colors

All symbols are defined like a SVG vector elements. Are stored in <defs> container of the SVG document. Symbols are derived from various thematic atlases from Czech (or Czechoslovakian) Republic which was published during last century. Symbols was selected according frequency of usage or according best perception parameters. Vector scale ability is used in very limited form, many symbols have scale dependent variants. For color manipulations was created special color metric based on munsell system. Basic idea is to generalize color scale where new class keep color of dominant included class from previous classification or new class color is in centroid, according proposed color metrics, of initial colors of incorporated elements.

4. PRACTICAL ASPECTS

Smart map was applied on few examples of thematic maps with dominance of areal features. Biggest tested sample was group of the hydrology themes in the Svratka river-basin. Source data was usually derived from maps in scales between 10 000 to 50 000. Scale manipulation was from 10 000 to 1 000 000. Necessary ECMAScript code was about 180 KB, coding is provided in pure procedural way. Empty map shell (i.e. basic layout with controls, without database and symbol library) is from 250KB to 600KB depends on layout composition. All application is sensitive on RAM memory, especially frequent manipulation with DOM structure. From this point of view Batik 1.5beta react better then Adobe SVGViewer 3.0, we do not use another engines, because of incompleteness in DOM method implementations. Is necessary to mention that our smart map is a cartographic application not GIS, interaction with user is quite rich but all visualization procedures are hardcoded in application. Massive preprocessing is result of used way, main reasons are efficiency of ECMAScript and XML parsers used in engines. This paper is without illustrations, example of map and pdf version of the appropriate poster will be placed at http://www.geogr.muni/lgc/smartmap.htm .

5. CONCLUSIONS

Nowadays status of SVG engines enable possibility to use this language for smart map development. The pure SVG approach is sufficient for cases with limited amount of geometric features, but of course it depends on hardware configuration. In future development i this are we try to database manipulation, include generalization procedures, and visualization with necessary number of controls. Development of smart map SVG framework is nowadays supported by projects "Open regional elecronic atlas" (205/03/1102) and "GISA2E" (F/01/B/P/PP - 118054 Leonardo da Vinci).

6. REFERENCES

[1] Friedmannová L., Staněk S: Systém for Dynamic Visualisation of Choropleth Map. Proceedings of The 20th ICC – Mapping the 21st Century, Bei.-Jing, China, st.2627-2632. (2001)

[2] Friedmannová L.: Klimatické mapy v atlasové tvorbě–vizualizační charakteristiky. Kartografické listy 9/2001, st.67-72. in czech (2001)

[3] Friedmannová L., Staněk K: Electronic multimedia choropleth maps design through SVG. In: SVG Open/ Carto.net Developers Konference (http://www.svgopen.org), Curych, Swiss. (2002)

[4] DeweeseT., HardyV., HillionS., Thierry K.: Introduction to the Apache Batik SVG Toolkit. In: SVG Open /Carto.net Developers Konference (http://www.svgopen.org), Curych, Swiss. (2002)

[5] Friedmannová L., Staněk K.: Using SVG for example design in WWW courses. In: Eugises 2002, Third European GIS Education Seminar, Girona, Spain. (2002)
[6] Lindsey K.: Tutorials. In: http://www.kevlindev.com/tutorials/index.htm (2000-2003)
[7] Scalable Vector Graphics (SVG) 1.1 Specification. In: http://www.w3.org/TR/SVG11/ (2003)
[8] Munsell A.H.: A Color Notation, Boston (1905)