A Software Framework for the Automated Production of Schematic Maps

Joao Mourinho[1], Teresa Galvao[1], Joao Falcao e Cunha[1], Fernando Vieira[2], and Jose Pacheco[2]

[1] Faculdade de Engenharia da Universidade do Porto, 4200-465 Porto
[2] OPT - Optimizacao e Planeamento de Transportes, 4200 Porto
joao.mourinho@fe.up.pt

Abstract. Schematic Maps are mainly used for depicting transportation networks. They are generated through a schematization process where irrelevant details are eliminated and important details are emphasized. This process, being manually performed by teams of expert designers, is expensive and time consuming. Such manual execution is unsuitable for the production of schematic maps for location-based services or on-demand schematic maps, as near real-time and user-centered properties are needed. This work proposes GeneX, a framework that can support the automated generation of schematic maps. The framework and the new algorithms developed were able to completely eliminate erroneous map point placement, and to decrease by 33% the contention for map point placement, producing schematic maps without human intervention in soft real time.

Keywords: Schematic Maps, Software Framework, Public Transportation

1 Introduction

Schematic maps have been increasingly used in response to the need of better and simpler maps to describe complex transport networks. This apparent simplicity is achieved through a simplification process called “schematization process” where choices are made regarding the level of detail and simplification. A special type of schematic map, called spider map, has also appeared recently. It presents innovative features such as a spider structure improve visual presentation, user learning and spatial context communication. Schematic maps, by their inherent simplicity and symbolic meaning are good maps for being used in the transportation area as they are far more intuitive than conventional maps [1]. In fact as people travel more often, they need flexible and easy to understand maps which may take in account their context [2]. Automation in the production of maps is a key factor to achieve flexibility to tailor maps to user context, as it happens with Location-Based Services [3]. There is the need, then, to develop a software framework which could support efficiently and comprehensively the automated generation of schematic maps. In this paper we propose and describe a software framework which serves as an engine to the generation and test of schematic maps.

2 Schematic and Spider Maps

Some authors define schematic map as *"an easy-to-follow diagrammatic representation based on highly generalized lines which is in general used for showing routes of transportation systems, such as subways, trams and buses, or for any scenario in which streams of objects at nodes in a network play a role"* [4]. One remarkable schematic map applied to a transportation network was the Harry Beck's London Underground diagram. Beck's map was considered both bold and innovative, as for the first time lines were drawn either horizontally, vertically or diagonally at 45. This map also uses differential zoom scaling and although it gives the traveler some clues about the terrain features (ex: river) and his/her location, it does not mimic the geography of London. Spider maps are special schematic maps. Like schematic maps, the stops and lines of the transportation network correspond to vertices and edges. However, they have enhanced features such a spider architecture, thus having a specific set of characteristics which sets them apart from schematic maps. Spider maps pay special attention to context in order to enhance user learning and ease of use. A spider map such as the one depicted in figure 1, comprehends three components:

- **Hub:** Describes the area in which the user is, as well as the surrounding area with a higher degree of detail (buildings, roads, etc). The hub, as it is the central part of the spider map, is the first component the user will look at, as it makes uses of *"focus and context"* [5] and detail focusing techniques. The hub may not comply with the 0/45/90 degrees line orientation.
- **Lines:** The lines follow the orthogonal orientation of the traditional schematic maps, and describe the paths of the transport network where the user can go through while being at the zone depicted by the hub.
- **Stops:** The stops are the destinations accessible to the user from the hub.

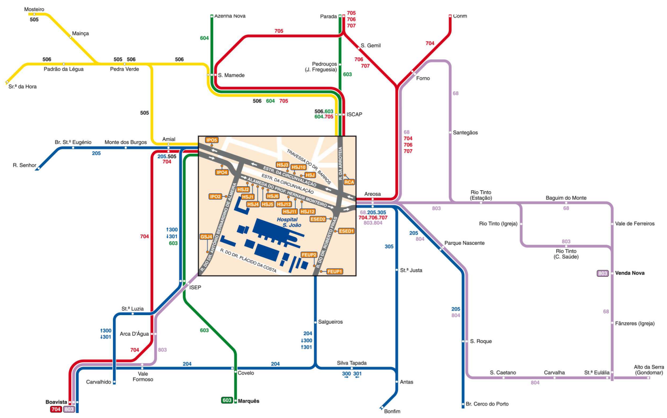

Fig. 1. Hospital de Sao Joao (Porto, Portugal) spider map. [6]

Visual simplicity of schematic maps is achieved through a sequential decision process regarding the level and nature of detail and schematization. In practice, this "schematization process" is still a manual process carried away by teams of expert designers. The automation of the schematization process requires effective and efficient algorithms to achieve, in one hand, high quality schematic maps which can be understood by people and, in the other hand, a time-efficient process. Through schematization, certain map details are emphasized while others are deemphasized. It is fundamental to present the smallest amount of information the user needs to learn the map to decrease user learning time. Therefore information shall be reduced to its basic components to achieve that goal. There are some studies regarding the automated drawing of schematic maps [7] [8] [9] [10], nevertheless these studies tend to focus only some areas of the problema and do not make a multidisciplinary approach. They mostly focus on the schematization process [11]. Nollenburg [12] [13] makes a deep research on the discrete mathematical foundations which are the basis of the algorithms used in the drawing of schematic maps and makes some brief considerations about their implementation. Nevertheless, his studies do not cover the human perception factors nor a concrete computer framework for drawing schematic maps. Silvana Avelar [1] [4] presents a broader study, by including some human perception factors and studies the schematic maps on demand. She goes further on by presenting a framework for electronic schematic maps which can answer user queries and studies the automated generation of schematic maps. Nevertheless, the study of the human perception factors is limited to what she calls the "aesthetic factors". Most of the algorithms to design schematic maps retain a common structure [14]. They use a graph to model the transportation network, in which the vertices represent stops or turning points and the edges represent the paths between two turning points.

3 The GeneX Framework

In this section, we present the GeneX framework which was developed through a collaboration research performed by a team involving collaborators from FEUP *(http://www.fe.up.pt)*, OPT *(http://www.opt.pt)*, STCP *(http://www.stcp.pt)*, FWT *(http://www.fwt.co.uk)*, INEGI *(www.inegi.up.pt)*, and that was funded by INEGI. The GeneX framework is a software application designed to support the following objectives:

1. The automatic generation of electronic schematic maps for complex transportation networks in bounded time, through the flexible use and parameterization of schematization algorithms
2. To serve as a test lab to support the research of schematic maps.

By merging and processing different kinds of external information (transportation networks, geographic and constraint information) through the use of state-of-the-art algorithms, the framework generates schematic maps automaticaly. The framework produces an SVG [3] file which can be used directly, printed in paper or further processed. The framework alsos produce a statistics file to measures several parameters about the framework functioning.

[3] The Scalable Vectorial Graphic is XML language to describe vectorial bidimentional graphics. It is an open format created by the World Wide Web Consortium

3.1 Software Engineering Life Cycle model

The requirements elicitation and validation was performed together with the project stakeholders, as part of a Joint Application Development (JAD) model [15]. The contributions provided by the partners were highly regarded: from the experts in information systems for transportation services (OPT), in optimization algorithms (INEGI) and map design (FWT), to the final users of the system (STCP). The development of this framework was considered, since the begining, an interdisciplinar subject in which knowledge from several areas of the science need to be integrated. Regarding functional requirements, the framework should be capable of producing schematic and spider maps in a fully automated way, about any location the user may select, using several schematization algorithms. The framework should be able to obtain data through the use of a common standard protocol data schema shared by the stakeholders. Another requirement was that the producton of schematic maps should be time-bounded (by setting deadlines or iterations number), to make it able to support Location-based services. In order to serve as a test lab, the framework should allow the choice of the algorithm and its parameters. The framework should also support different schematization algorithms (genetic, linear, tabu search, GRASP, etc) and provide a common protocol to implement them. This set of functional requirements needs to be supported by a set of non-funcional requirements, which comprehends usability, performance and interoperability. Usability is fundamental as the schematic maps produced by the framework have a strong user learning component: the maps produced, as well all possible interactions should be as intuitive as possible to be quickly learned and understood by their users. The designers and the final user team members provide insightful highlights in this area. To be able to support location-based services, the framework should execute the algorithm and produce the correspondent schematic map in near real time. Therefore, the framework was designed to perform as a soft real-time system [16]. A standardized transport network data specificication was implemented and a common interface for the algorithms was designed to achieve interoperability. Extensive use of reusable components [17] [18] was also made. The framework was developed by using C# Language, as a modern Object Oriented language which supported the requirement list.

3.2 Architecture and Data Model

The architecture of the framework follows a modular structure, with two main modules: the data preparation module and the algorithm execution module. Figure 2 shows the GeneX framwork package diagram, depicting its components.

The data aquisition and preparation module is responsible for preparing the data to be used by the algorithm execution module. The user selects graphically the location where he wants the spider/schematic map to be centered (hub). The module then extracts raw data from the transport network database and organizes it into a data structure by using the Spider Map Library. The spider

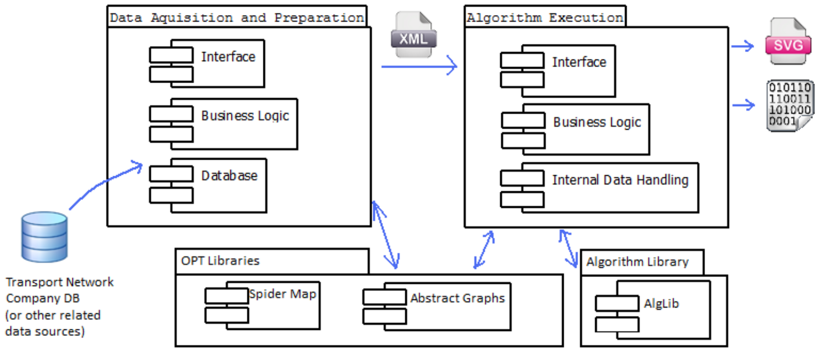

Fig. 2. GeneX Framework Package Diagram

map library is a complex set of C# classes that support the serialization and communicaton of the spider and schematic map structure. The algorithm execution module reads the XML file and transforms the data into an internal data structure (to improve component reuse by the schematization algorithms). At the user interface, the user can choose the schematization algorithm (from the AlgLib algorithms package) to execute and its options and performance measurement metrics. The business logic is then responsible for calling and executing the algorithm and to produce the final result, which can be an SVG (Scalable Vector Graphics) or a binary file containing serialized data. The SVG file is produced by using a library that allows the conversion of a spider map data structure in an SVG file called Abstract Graphs Library [6].

3.3 The HPPO Algorithm

The AlgLib package provides a foundation for the execution and configuration of the schematization algorithms. Each algorithm has to implement the same communication interface functions, in order to be used by the execution module:

- ***spiderMap*** **execute(*spiderMap*, *parameterList*)** performs the execution of the algorithm. The arguments are the spider map XML structure that was opened by the execution module, and the algorithm parameter list, already set up by the user. This function returns a spiderMap data structure which is the processed spider/schematic map. That structure can then be output to a SVG or serialized binary file.
- ***parameterList*** **getParameters()** the module calls this parameter function to get the list of the algorithm parameters that can be set by the user.

We have implemented a preliminary schematization algorithm that we called "Heuristic Point Placement Optimization" (HPPO). HPPO, aligns map points (corresponding to the transportation network stops and stations) to a regular grid, by positioning each map point in the nearest grid intersection. For high density or non regular density transportation maps, map station contention for grid intersections will happen. In this case, the HPPO smartly solves the contention through an heuristic algorithm. By using regular expressions, the point labels are also taken into account when placing network vertices, by determining the similarity degree of node labels in conjunction with its geographic location in order to produce a decision about the location to plot the node where the contention arises. The use of regular expressions is based in a finite-state automaton which scans strings in order to find the degree of similarity. We are not only relying in geographical information, but merging knowledge from different science fields theory to make a higher quality judgement on how to solve the contention, such as computer science and operations research. For each map point, HPPO starts by checking if the nearest grid intersection is empty. If it is, then there is no contention and the point is plotted there. If the grid intersection is not empty, then we have a contention. This means that the geographical coordinates of the nodes are equal, or at least, within the same decision range concerning the square grid resolution. The automaton also tells us the degree of similarity. If the degree of similarity is higher than the predefined threshold level, then we assume that both nodes refer to the same location. In this case, they should be both plotted in the same grid intersection. If the degree of similarity is lower than the threshold, then it is important to distinguish them and plot them in different grid intersections while maintaining the topological relation between them. In this case, we get the topological relation between the two points (based on their coordinates) and try to move the node to the adequate grid intersection. It was found that preserving the topological relation between map points is of fundamental importance when developing map design frameworks. If contention happens again (what can happen in a highly crowded map or with a loose square grid), then we have two options: or we may continue this cycle recursively until the topological relation is violated, or we may decide if we shall plot the node into the suggested grid intersection. To limit the processing time, we discarded the recursive approach in our algorithm. Being so, we analyse the proposed grid intersection. If contention happens again we check the degree of label similarity through our automaton, and if is higher than the threshold, we plot the point there. If not, we analyse both the first and the actual grid intersections suggested and we add the node where there are less nodes plotted. The pseudoalgorithm is described in figure 3.

4 Results

The GeneX Framework was able to generate in a fully automated way schematic and spider maps for every location requested. It is also a very important tool as a test lab for the schematization algorithms that are being developed. Although the framework is still under development and the algorithms in the AlgLib are being improved and enhanced, it is already being used for the production of schematic and spider maps. Maps produced with this framework are already available for public use in the cities of Porto and Lisbon. Concerning our HPPO algorithm, it showed good results, as it can be observed in figure 4. Other advantage of HPPO is that if different nodes refer to the same place, this algorithm can ignore

```
Foreach Node(CoordX, CoordY, Label) In Transportation network
    Get candidate grid intersection C based in node coordinates
    If C is empty
        Then Plot Node on C
    Else
        If degree of similarity between the node(s) already plot on C and the node
        being analysed > threshold
            Then Plot Node on C
        Else
            Calculate the topological relation between the node being plotted and
            the node(s) already plotted in C and get the candidate related grid
            intersection D
            If D is empty
                Then Plot Node in D
            Else
                If degree of similarity between the node(s) already plot on D
                and the node being analyzed > threshold
                    Then Plot Node in D
                Else
                    Compare number of nodes plot in C and D and plot the
                    node in C if |C| <= |D| or in D if |D| < |C|
```

Fig. 3. HPPO pseudoalgorithm description.

contention and plots them correctly in the same grid intersection (grouping). In addition, all the topological relations are still preserved.

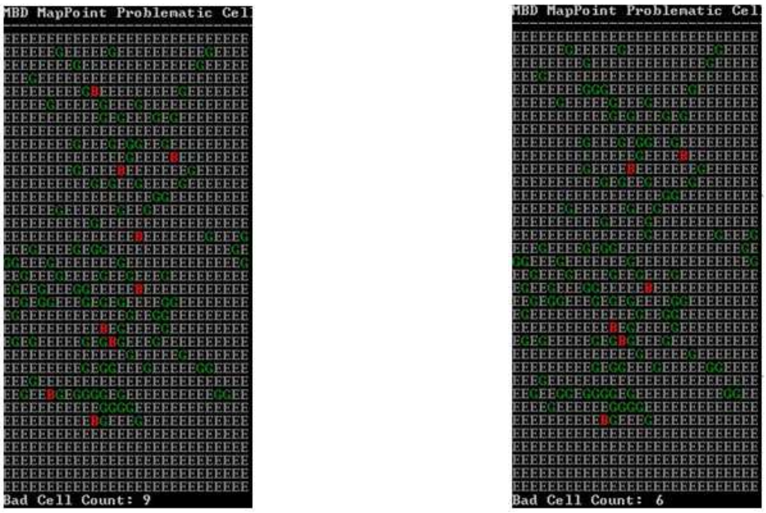

Fig. 4. Frequency matrixes showing Bad Cells (in red) when not using HPPO(left) and using HPPO(right) for the Porto downtown spider map. HPPO reduced Bad Cells from 9 no 6, while preserving topological relations

5 Conclusions and Future Work

The GeneX framework was used to produce spider maps that are being used in the cities Porto and Lisbon, it has proved effective in generating spider maps. Nevertheless, these maps were not completely produced by an automated process, as some manual changes were necessary to improve the visual appearance, which is the most difficult aspect to model in the algorithms. The quality of the results of the HPPO algorithm is quite good, showing a significant decrease in bad cells. The algorithms need to be perfected in order to increase the quality of the results and to make them directly usable (without any manual processing). The algorithms need to further improve aspects such as visual line distiction, stop label organization and geographical constraints. Some development of the framework is also needed to support Location-based services, such a "request manager" which can feed user requests to the framework and reply to them. Other issues that need further study are the adaptation of the resulting maps to different devices. All this work also needs to be complemented and validated with usability tests and analysis.

References

1. Avelar, S.: Schematic Maps on Demand: Design, Modeling and Visualization. PhD thesis, Swiss Federal Institute of Technology Zurich (2002)
2. Porathe, T.: User-Centered Map Design. Design (2007)
3. Steiniger, S., Neun, M., Edwardes, A.: Foundations of location based services. (2006)
4. Avelar, S., Hurni, L.: On the Design of Schematic Transport Maps. Cartographica: The International Journal for Geographic Information and Geovisualization **41** (2006) 217–228
5. Bogen, S., Brandes, U., Ziezold, H.: Visual Navigation with Schematic Maps. Visual Information Communication (2010) 65–84
6. OPT: Abstract Graphs Library Specification (Internal Report) (2009)
7. Cabello, S., Deberg, M., Vankreveld, M.: Schematization of networks. Computational Geometry **30** (2005) 223–238
8. Cabello, S., Kreveld, M.V., Sciences, C., Box, P.O.: Schematic Networks: An Algorithm and its Implementation. In Richardson, D., Oosterom, P., eds.: 10th International Symposium on Spatial Data Handling (SDH), Ottawa, Springer (2002) 475–486
9. Barkowsky, T., Latecki, L.J., Richter, K.f.: Schematizing Maps : Simplification of Geographic. Cognition **8** (2000) 41–53
10. Anand, S., Avelar, S., Ware, J.M., Jackson, M.: Automated schematic map production using simulated annealing and gradient descent approaches. Technology (2000)
11. Ware, J.M., Taylor, G.E., Thomas, N.: Automated Production of Schematic Maps for Mobile Applications. Transactions in GIS **10** (2006) 25– 42
12. Nöllenburg, M.: Automated drawing of metro maps. PhD thesis, Universitat Karlsruhe (2005)
13. Nöllenburg, M., Wolff, A.: A Mixed-Integer Program for Drawing High-Quality Metro Maps. Graph Drawing **3843** (2006) 321–333
14. Dong, W., Guo, Q., Liu, J.: Schematic road network map progressive generalization based on multiple constraints. Geo-spatial Information Science **11** (2008) 215–220
15. Scacchi, W.: Process Models in Software Engineering. October (2001) 1–24
16. Laurini, R., Servigne, S., Nol, G.: Soft real-time GIS for disaster monitoring, Springer-Verlag (2005) 465–480
17. Neighbors, J.M.: The Draco approach to constructing software from reusable components. IEEE Transactions on Software Engineering **10** (1984) 567–574
18. Goguen, J.A.: Reusing and interconnecting software components. Computer **19** (1986) 16–27