An Approach to Prepare Data to Feed Visual Metaphors in a Multiple View Interactive Environment

Ronaldo de Matos Nascimento Filho, Glauco de Figueiredo Carneiro
Universidade Salvador (UNIFACS), Salvador, Bahia, Brazil
rronaldodematos@hotmail.com, glauco.carneiro@unifacs.br
Miguel Monteiro
NOVA LINCS, Universidade Nova de Lisboa (FCT/UNL), Portugal
mtpm@fct.unl.pt

***Abstract*—This paper presents an approach to prepare data to feed visual metaphors in a multiple view interactive environment. We implemented a tool that supports programmers and users to prepare datasets from different domains to feed visual metaphors. To analyze the effectiveness of the approach, we conducted a case study with the data of the Brazilian National Health System (known as *SUS - Sistema Unico de Saude* in Portuguese). The results obtained are an initial evidence of the feasibility of the approach that support the preparation of data to a format suitable to the characteristics of visual metaphors. The case study illustrates scenarios in which both programmers and users are able to prepare datasets from different domains to feed visual metaphors that comprise a multiple view interactive visualization infrastructure.**

***Index Terms*—cloud computing; legacy systems; cloud migration; tools.**

I. Introduction

According to [1], data visualizations are algorithmically generated and can be easily regenerated with different data. Moreover, they are usually data-rich, and are often aesthetically shallow [1]. Popular tools for data visualization still have limitations, yielding visualizations of somewhat poor-quality, with little or no interactivity with the tool user.

Practitioners have claimed that although the number of data file formats is unbounded, the structure of any data could be described using a small number of parameters [2]. The range of visualization algorithms applicable to a given type of data is narrow and the subset used within a given scientific discipline is even narrower. Thus, hardly all possible data structures need be covered, with the consequence that the solution space tailored to these activities should be relatively simple and easy to maintain. Researchers have also claimed that data visualization can point to new questions and to additional exploration directions, help identify sub-problems, show great volumes of data and leverage the human cognitive capabilities to identify patterns and communicate relationships and meaning [1]. Thus, it is desirable - and also feasible - to build a data visualization tool targeted for users that are not knowledgeable of these matters but need to deal with data in their daily activities.

The Problem. To build graphical representations of data and make it available in the internet, tools such as HTML, CSS are usually used. Depending on the visualization metaphor, Scalable Vector Graphics (SVG) elements may also be required. Though preparation can be carried out manually, such a task would be laborious, time consuming and not scalable to large amounts of data. In addition, reuse requires considerable effort to be achieved. On the other hand, processes for building a graphical representation of data can be automated - with the aid of tools such as Javascript - thus facilitating the regeneration of the representation from different data sets (provided they are of similar structure). Automation makes it feasible to represent large quantities of data. These approaches are classified under *data visualization* [1]. The drawback is that build work will be larger, to prepare the whole dataset.

To tackle this problem, we decided to select tools, toolkits and frameworks that could aid in the aforementioned dataset preparation, so that users can focus on the problems and corresponding solutions. Taking this into account, we proposed an approach to prepare data to feed visual metaphors through the implementation of a multiple view interactive environment. Through it, users can explore and analyze the data by selecting the visual metaphors that best fit the tasks at hand.

The rest of this paper is organized as follows. The next section presents the research background. Section 3 describes the methodology of the study. Section 4 analyzes the results. Section 5 presents the conclusion and scope for future research.

II. Research Background

Regarding data visualization, unlike the traditional tree diagram, the *treemap* visual metaphor provides a graphical representation of nested (hierarchical) data and their respective values. These values can be represented by both the area of each rectangle and its color intensity. This visual metaphor provides an overview of all the data and their descendants, as well as their value [3]. On the other hand, the *sunburst* visual metaphor presents the same aspects of data as with the *treemap*, but with a different approach, facilitating the analysis of the hierarchical levels of the data, without one sub-level interfering in the visualization of another. It also provides a panoramic view of all levels [4].

The *circle packing* metaphor was designed for the visualization of data in categorized form, suitable for hierarchical data with many levels. It enables the recursive navigation through the various levels of a dataset structure.

Unlike the aforementioned views, which focus on hierarchical data, the *chord* view provides an alternative for representing relationships between entities, making it possible to see which entities are related to each other, as well as the direction of that relationship. Depending on the data, the understanding of this representation can be compromised, because if there are too many relations between the entities, some of the representations of the relations may overlap one another. For this reason, an interactive solution was adopted.

III. Proposed Solution

To perform the proposed approach, we implemented a multiple view interactive environment in which the visual metaphors are instantiated to represent data provided in JavaScript Object Notation (JSON). The graphical representations enable users to interact through filters, so as more detailed data display features.

Functional Requirements. The environment should support the generation of interactive graphical representations, based on data that can be filtered according to users' needs. For this purpose, the data sent to the systems must be in a structure, defined here as a characteristic of the data, to be exposed during the process. The environment should provide filters on the data sent by the user.

Non-Functional Requirements. For scalability purposes, propagation and ease of use, the system must be accessible through a web browser. Regarding the architecture of the environment, it should allow the processing and manipulation of data on the server. This targets a thin web browser client. The environment can be extended to support data manipulation, validation, and restructuring. Views should be built in SVG and the interface should be responsive. The environment should also support a large amount of data as input. Moreover, it should be extensible to enable the inclusion of new implementations of data characteristics, visual metaphors and their respective filters.

Steps to Prepare Data to Feed Visual Metaphors The activity diagram shown in figure 1 presents suggested steps to select and adjust the visual metaphors in the proposed multiple view interactive environment. As an initial requirement, the user select a dataset with the JSON format. Note the system was designed with the intention of adding future implementations, and the options shown here refer to just the visualization metaphors implemented so far. In Table I, we present a short explanation of the suggested steps:

TABLE I
Steps to Prepare Data to Feed Visual Metaphors

Step	Description
1	Select the Type of Data
2	Select the Visual Metaphor
3	Provide Data in the JSON format
4	Indicate which visual attributes will represent data real attributes in the selected visual metaphor [5]
5	Filter data to be presented in the visual metaphor
6	Build the Visual Metaphor

Proposed solution. The proposed architecture focus on the principle of the *thin client* architecture. This enables a minimal set of software requirement to function as a front-end user interface for a web application. On the other hand, this approach is limited to support an interactive real-time visual data manipulation and graphical visualization environment. Considering this scenario, we decided to implement the solution through a ”thin” client. As a result, we needed to tailor the implementation of the visual metaphors to allow a thin client (i.e., any web browser) to send data and configurations for processing and instantiation of visualization metaphors, with the aid of filters and data treatments. This way, it is possible to transfer the heavier steps to the server whenever possible. Only the pre-processed data for rendering the visualization is returned to the client.

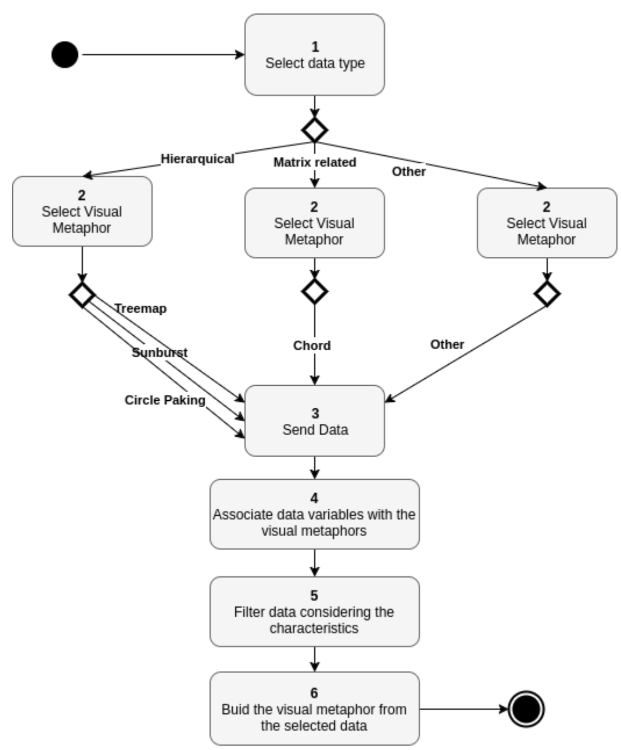

Fig. 1. Steps to Prepare Data to Feed Visual Metaphors

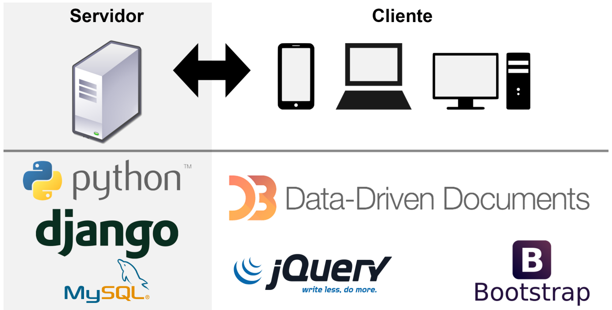

Fig. 2. Components of the Proposed Solution

In order to meet the requirements of the proposed solution, we planned the architecture presented in Figure 2. In the following paragraphs, we describe the components that comprise this architecture.

Python and Django The Python programming language was chosen for its scalability, support for productivity, simplicity and its open source nature. Django is an open source framework for Python, through which it is possible to focus directly on the solution. It offers us a whole structure for the creation of internet systems, in a simple and easy way, using an MVT (Model, View , Template), which resembles the well-known MVC (Model, View , Controller) pattern. The main differences can be found in the View layer. Django also brings an Object-relational mapping (ORM), allowing the creation of database-linked applications, simply and quickly, abstracting any SQL line of code. Another advantage of Django is its automatically generated administrative interface, which can be easily configured from the classes created in the Model layer. This interface allows us to perform basic operations such as CRUD (Create, Read, Update, Delete), which may also be extended and/or modified separately.

MySql. Considering that the proposed environment does not require specific database features, we decided to use MySql. It is a widely used database that provides all the basic requirements of a database, and it can be easily integrated with a Django application.

Javascript and Jquery. Current browsers have adopted Javascript as the main programming language. For this reason, we decided to adopt in the proposed environment JavaScript snippets and also snippets of Jquery code - a popular Javascript framework [6]. This framework allows for code simplification, enhancing code productivity and quality.

D3.js We selected the D3.js toolkit due to its ease of use, diversity and quality of visual metaphors already implemented [6] The toolkit is not limited to data visualization. It also includes facilities for object manipulation, requests, conversions, interactivity control, Document Object Model (DOM) manipulators and more.

CSS3 and Bootstrap CSS became the *de facto* standard for styling on the internet, despite minor differences in rendering between browsers. Its interface and important elements such as margins and spacing are not disrupted. However, the implementation of responsive elements via CSS are inherently verbose. For this reason, the most commonly used front-end framework - Bootstrap - was adopted. This tool supports the construction of responsive interfaces, besides many other possibilities of stylization.

IV. The Case Study

In this section, we present a case study conducted to analyze the effectiveness of the solution proposed in this paper regarding the preparation of data to feed visual metaphors in a multiple view interactive environment.

A. *Problems to be addressed*

One way to extract information from a large amount of data is by analyzing a graphical representation of the dataset that represent it. To tackle this problem, we implemented the proposed solution presented in Section III.

The solution can be used by non-programmers as long as the visual metaphors they need were implemented prior to use and made available in the environment. In that case, the user just has to convert the data to be analyzed to an appropriate format (JSON). Considering that the environment is extensible, a visual metaphor can be included in the environment by a programmer, in case is not available.

B. *Aims*

The purpose of this case study is to illustrate the viability of use of an approach to prepare data to feed visual metaphors to in a multiple view interactive environment.

C. *Research Questions*

Research Question 1 (RQ1): To which extent can non-programmers generate a graphical representation of data in JSON format in the proposed multiple view environment?

Research Question 2 (RQ2): To which extent can non-programmers interact with the selected visual metaphors and analyze data provided to the environment in JSON format?

D. *Tasks and Used Data*

In this case study, we illustrated the viability of the multiple view interactive environment to prepare data to feed visual metaphors using data from the Brazilian Unified Health System (SUS) publicly available by the Brazilian government[1]. Public health plays an important role for citizens and has drawn attention of both policymakers and the public at large. Concerns about gaps in both the availability and quality of public health services have hit the headlines in response to both new and persistent health risks [7]. For this reason, decision makers and practitioners must be careful about the allocation, management, and administration of public health resources [8].

Brazil's citizens expect the public health system to make appropriate use of out-of-pocket health expenses. In 1988, within the new Constitution, the Brazilian Unified Health System (SUS) was created by the Brazilian Ministry of Health to offer free health care to the population, based on three pillars: universal coverage, integral health care and equity [9].

[1] datasus.saude.gov.br

E. Strategy to perform the tasks

The activities executed in the case study followed the steps described in Table I and illustrated in Figure 1. All the steps were previously tested by the first author, as well as the appropriateness of the data to instantiate and interact with the visual metaphors in the multiple view interactive environment.

1) **Step 1 - Select Data Type:** According to Figure 3, the user inform the type of data to feed the visual metaphors in the environment. The current version provides support for two types of data structure. The first is related to the relationship among data entities such as the one represented in a graph. For example, a graph that represent a standard graph of a social network. The second is related to hierarchical data as essentially a specialized form of network data. The difference from the first type of data structure is that in the second type they are all related to each other by the principle of containment. They, unlike standard data networks, do not use the principle of connection.

2) **Step 2 - Select the Visual Metaphor:** the user selects the visual metaphor according to the data structure of the data he/she provided (Figure 4). In this case, the treemap, sunburst and circle parking visual metaphors are related to hierarchical data representation. On the other hand, the network related data can be represented by the chord visual metaphor.

3) **Step 3 - Provide Data in the JSON format:** the user select the path or url that contains the data to feed the selected visual metaphor in the environment (Figure 5).

4) **Step 4 - Indicate which Visual Attributes will Represent Data Real Attributes in the Selected Visual Metaphor [5]:** in this step the user indicates how entities from the data structure will be visually represented using the data attributes (Figure 6).

5) **Step 5 - Filter Data to be Represented in the Visual Metaphor:** in this step the user indicates the range of values to filter data to be visually represented (Figures 7, 8 and 9).

6) **Step 6 - Build the Visual Metaphor:** in this step the user requests the tool to carry out the build of the selected visual metaphor (Figures 10 and 11).

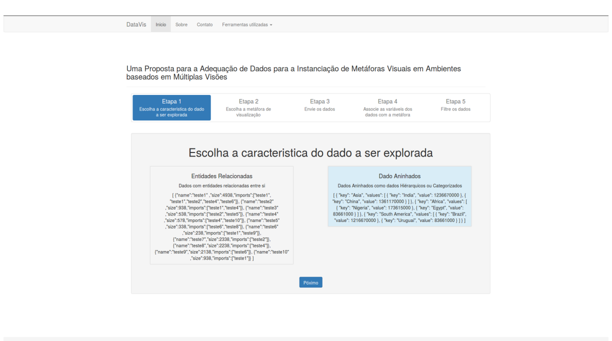

Fig. 3. Step 1: Select the type of data

Fig. 4. Step 2 - Select the Visual Metaphor

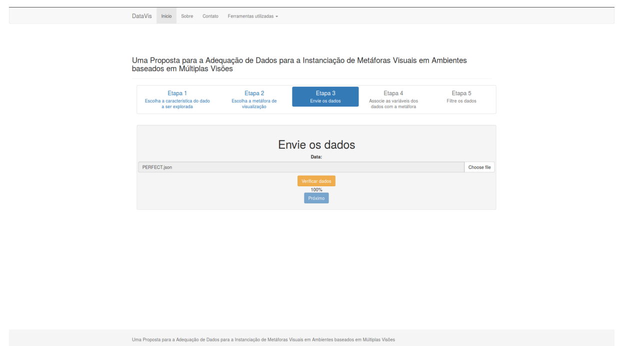

Fig. 5. Step 3 - Provide Data in the JSON format

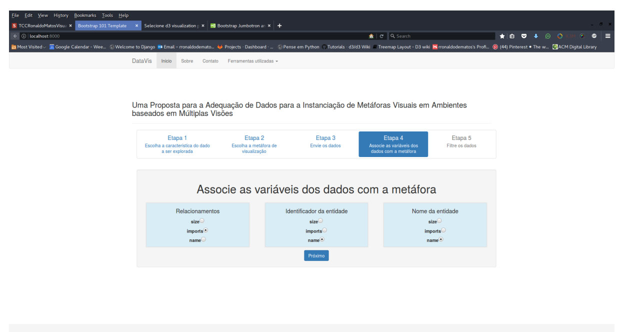

Fig. 6. Step 4 - Indicate which Visual Attributes will Represent Data Real Attributes in the Selected Visual Metaphor

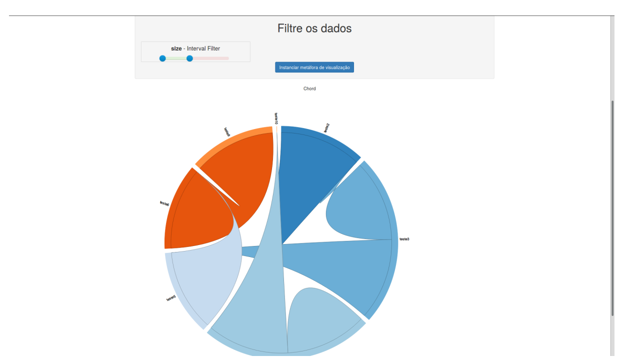

Fig. 7. Step 5 - Filter Data to be Represented in the Chord Visual Metaphor

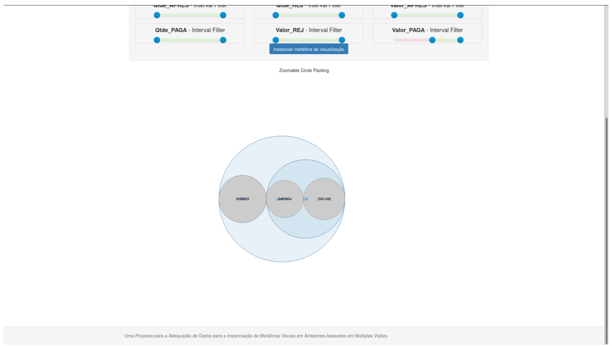

Fig. 8. Step 5 - Filter Data to be Represented in the Circle Packing Visual Metaphor

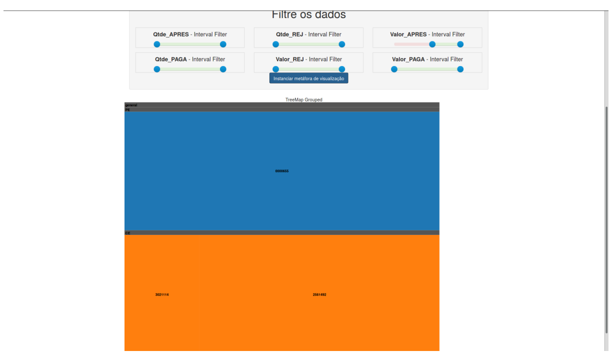

Fig. 9. Step 5 - Filter Data to be Represented in the Treemap Visual Metaphor

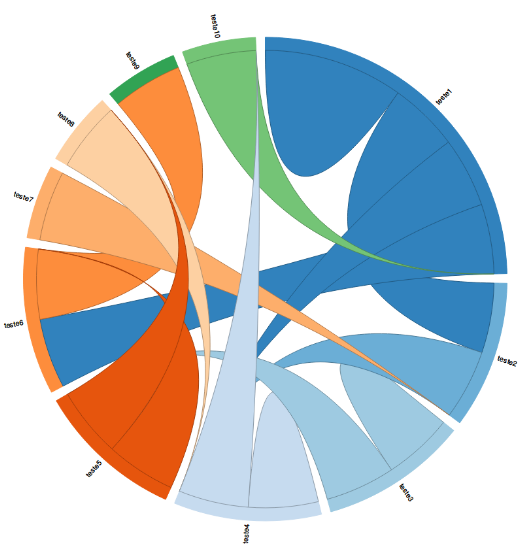

Fig. 10. Step 6 - Build the Visual Metaphor (The Chord Visual Metaphor)

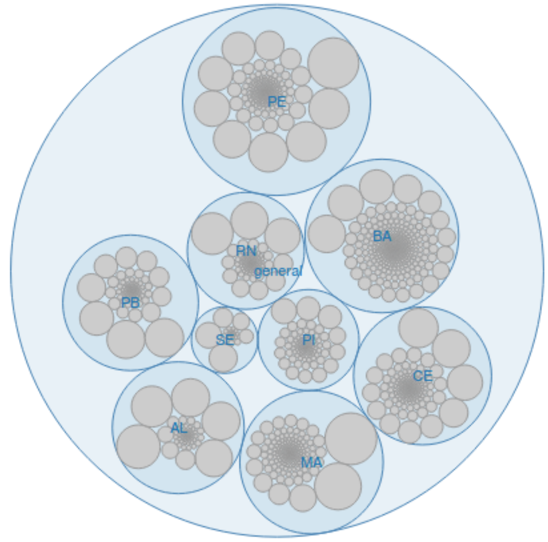

Fig. 11. Step 6 - Build the Visual Metaphor (The Circlepaking Visual Metaphor)

Fig. 12. Step 6 - Build the Visual Metaphor (The Treemap Visual Metaphor)

F. Results

Considering the steps described before, we present the results using data from the Brazilian National Health System.

The participant combined the use of the available visual metaphors (treemap, chord, sunburst and circle packing) to accomplish the following tasks: **Task 1 (T1)**: *What are the top three Brazilian Northeast States in terms of highest approved values for performed procedures?* **Task 2 (T2)**:*Considering the States selected from Q1, which month had the highest approved value for performed procedures?*

The panoramic view provided by treemap (Fig. 12) can initially represent data related to the first level, i.e. the states in terms of approved accounts and the respective budget allocated for them. On the other hand, the circle packing visual metaphor not only provides a panoramic view of data (Fig. 11), but it also represents the subsequent levels at the same time. This helps the user to know to which extent the set of analyzed states, period of time (in this case, month), and health unit contribute in the amount of approved accounts.

To perform **Tasks 1 (T1) and 2 (T2)**, the participant identified the Brazilian Northeast States that had the highest approved values for performed procedures. In this version of treemap, the rectangle visual attribute is associated with the amount of approved accounts of a given State.

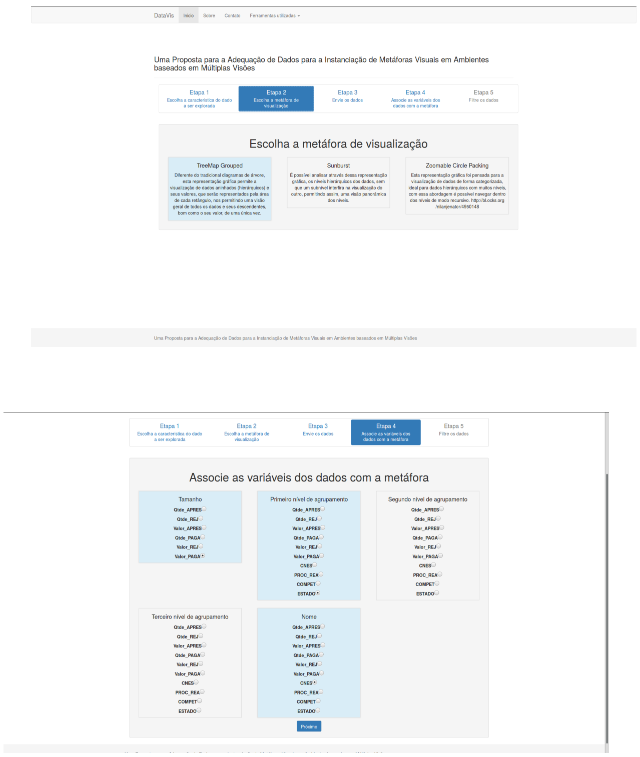

Fig. 13. Estudo de caso da Step 4

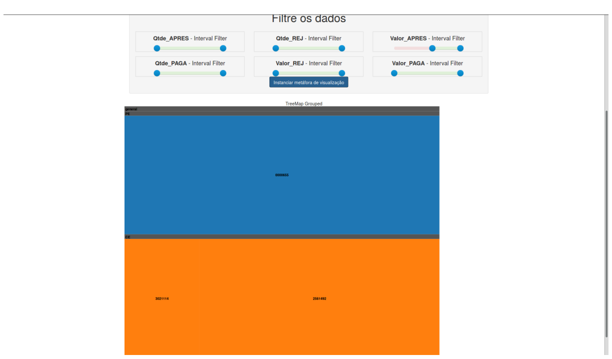

Fig. 14. Estudo de caso da Step 5

V. Conclusion

With this case study with three visual metaphors, it was possible to analyze the adequacy of data for these models, as well as the approach to prepare data to feed visual metaphors in a multiple view interactive environment. It was also possible to check the interactivity with the filters, zoom, and highlighting for selected object.

With this approach, the process of understanding the data through graphic representation was summarized in six steps, using only the web interface, without the need to program a single line of code.

References

[1] J. Iliinsky, Noah; Steele, *Designing Data Visualizations*, 1st ed. O'Reilly, 2011.

[2] S. A. W. B. M. J. G. S. Johnson, Gregory P.; Mock, "Envision: A web-based tool for scientific visualization," in *EnVision: A Web-Based Tool for Scientific Visualization*. 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009.

[3] B. B. Bederson, B. Shneiderman, and M. Wattenberg, "Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies," *AcM Transactions on Graphics (TOG)*, vol. 21, no. 4, pp. 833–854, 2002.

[4] J. Stasko and E. Zhang, "Focus+ context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations," in *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*. IEEE, 2000, pp. 57–65.

[5] M. Carpendale, "Considering visual variables as a basis for information visualisation," 2003.

[6] V. H. J. Bostock, Michael; Ogievetsky, "D3 data-driven documents," in *D3 Data-Driven Documents*. IEEE Transactions on Visualization and Computer Graphics, Vol. 17, 2011.

[7] G. P. Mays, F. D. Scutchfield, M. W. Bhandari, and S. A. Smith, "Understanding the organization of public health delivery systems: an empirical typology," *Milbank Quarterly*, vol. 88, no. 1, pp. 81–111, 2010.

[8] F. D. Scutchfield, M. W. Bhandari, N. A. Lawhorn, C. D. Lamberth, and R. C. Ingram, "Public health performance," *American Journal of Preventive Medicine*, vol. 36, no. 3, pp. 266–272, 2009.

[9] A. J. Barros and A. D. Bertoldi, "Out-of-pocket health expenditure in a population covered by the family health program in brazil," *International journal of epidemiology*, vol. 37, no. 4, pp. 758–765, 2008.