

A Highly Customizable Information Visualization Framework

Luís Spínola, Daniel Castro Silva, and Luís Paulo Reis

Faculty of Engineering, University of Porto (FEUP)
Artificial Intelligence and Computer Science Laboratory (LIACC)
Rua Dr. Roberto Frias s/n 4200-465 Porto, Portugal
(up201405907, dcs, lpreis)[@fe.up.pt](mailto:fe.up.pt)

Abstract. The human brain can quickly become overwhelmed by the amounts of data computers can process. Consequently, data abstraction is necessary for a user to grasp information and identify valuable patterns. Data is usually abstracted in a pictorial or graphical format. Nowadays, users demand more personalization from the systems they use. This work proposes a user-centered framework that aims to ease creating visualizations for the developers of a platform while offering the end-user a highly customizable experience. The conceptualized solution was prototyped and tested to ensure the information about the data is transmitted to the user in a quick and effective manner. The results of a user study showed that users are pleased with the usability of the prototype and prove that they desire control over the configuration of their visualizations. This work not only confirmed the usefulness of previously explored personalization options for visual representations, but also explored promising new personalization options.

Keywords: Information Visualization · Visual Analytics · Visualization Personalization.

1 Introduction

The visualization process centers on converting data into graphics, and there are numerous methods to do so [7]. Information Visualization (InfoVis) systems conventionally follow a model offering a one-size-fits-all solution, meaning representations do not change, ignoring user preferences, capacities and overall circumstances [11]. When developing a platform to expose visualizations, there needs to be a concern with the overall design, usability and interactivity, besides the efficacy of the methods used [4] [8]. However, the responsibility of selecting the best approach for each set of information usually falls under the developer.

This work emerged to explore an InfoVis process to implement graphic platforms devoted to analyzing and monitoring data. It contributes with a scalable framework offering: Visualizations of performance indicators, common graphics and means to display geo-referenced information; A method to configure and customize visualizations to cater to user preferences; Means for a developer to easily implement a visualization in a system, providing a simple workflow.

2 Background

While the amount of data keeps growing, human attention and time do not expand with that trend, instead remaining constant [2]. With larger sets of data, users need to test more attributes and experiment more before coming up with suitable abstractions [3]. Furthermore, more people are performing data analysis, and business users, while experts in their area, don't typically possess the skills to choose a suitable visual metaphor from available options [7].

Research has proven that users behave as individuals with completely different desires – their needs, abilities, and preferences have a significant impact on their performance and satisfaction when using InfoVis techniques [6]. Thus, techniques should consider the reader [12]. On that account, it is crucial to scrutinize the possibilities of new kinds of user-centered InfoVis systems. One example that supports this hypothesis is [10], where several studies were conducted: Experts were asked to choose a type of encoding for different data. Results show that there is in fact a substantial variability in visualization preferences for the same data set; The second focused on user's prior experience, visual literacy and cognitive capabilities, by trying to identify what types of graphical representations are more adequate for each type of user; A third, tried to understand how the purpose with the data could take place to change the visualization requirements. Two states were taken in consideration — analyzing and monitoring.

In [5] the concept of graphical overlays was introduced, and a system was developed to allow the dynamic addition of the proposed overlays: **Reference structures** - to assist the reader in comparing and extracting values, for instance by using grid lines; **Highlight** - to emphasize some aspect of a visualization; **Redundant encoding** - used to aid in the extraction of values or to better portray tendencies; (for example, showing data labels directly on the graphic); **Summary statistics** - helpful to portray useful statistical information about a data set. The paper present the mean, median, and maximum as examples.

A concept similar to graphical overlays was introduced (visualization aids) [11], which, much like overlays, can be included in a representation without having previous insights of user's intent. To evaluate the visual prompts, a study was conducted using two techniques, namely bar and line charts, alongside checkboxes allowing to toggle on and off five different aids. A task consisted of showing graphics to a user, alongside questions that required interpreting some data. With 40 participants and 50 tasks to be completed, the mean accuracy was 87% and showed that a user turned on an aid with an average of around 49 times. The most selected option was 'show data', followed by 'horizontal line grid'; 'vertical grid line', 'fill area' and 'dot grid' were barely used.

Similar work proposes options that are used as overlays in InfoVis techniques [5] [11]. That fact makes it viable to be used in previously done visualizations, but limits the amount of overlays that can be applied and studied. This work proposes a distinctive approach that follows the trends of research on InfoVis aids by suggesting a framework for producing visualizations that offers personalizations that are proven to add value to the readability of a representation, also serving as a platform to investigate further aids.

3 Solution Conceptualization

A family of graphics is a set of visualizations serving a purpose. When seeding data to start the process, the programmer needs to specify which family the data belongs to. The families introduced so far in the prototype are: **One Numerical** - Simple categorical data (bar and pie chart); **Two Numerical** - To compare two numerical values (side-by-side comparison bar, scatter, and bar chart); **N Numerical** - Data distribution when more than two categories are present (bar, line, and area charts). **Time Series** - Data over time (line, area, and bar charts); **Performance** - Display KPIs (gauge, bullet plot); **Geo Quantities** - Values at coordinates (column and bubble map); **Geo Densities** - Density graphs (heat map, hexagonal map); **Geo Path** - Paths. A temporal sequence of coordinates traveled, with icons displaying stops. This division provides flexibility to the developer. When seeding data, the developer knows what graphics are available to the user. Families in this prototype follow the most common norms for each type of data. Still, families can be added by using existing graphics, granting them a different objective and combining them with new techniques.

The framework is divided into three simple layers. Figure 1 depicts the flow between layers. The first layer, **Data Input**, receives the data in JSON format. The metadata of the JSON object is created and appended to the data fetched. The developer queries the data and appends the metadata, which is used to specify the graphic family, as well as inform the next layer of which data key corresponds to what (for example which key corresponds to the date in a Time Series or which key corresponds to the coordinates in a georeferenced representation). The developer can also specify a user ID and a graph ID that in conjunction are unique and are needed to make user configurations persistent.

The second layer, **Graph Chooser**, uses the information from the first layer and selects the adequate graphic family for the data. User personalizations and graph selection made in the following layer are persistent. As such, this layer queries the database to find out if the visual representation is being revisited and, if so, loads previous graph selection and personalizations.

The final layer, **View/Personalization**, is the stage that handles the visualization and the process of personalization configuration by the user. This layer also communicates with the database to store individual graphic information and, if desired, can also store user metrics such as preferred graph per family and the amount of times each personalization option was used.

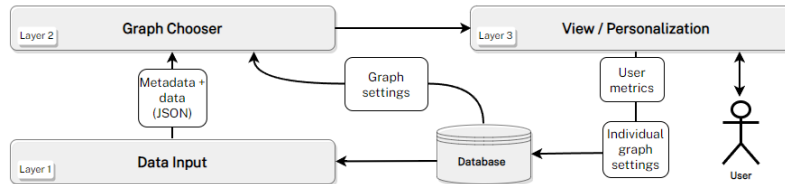


Fig. 1: Framework Architecture Diagram.

4 Implementation Details

Visualization on the web cannot solely offer a static representation as interaction eventually triggers some change, and although treated as different components, representation and interaction are intrinsically connected and positively contribute to the final experience [14]. Thus, all graphics provided offer interaction.

When evaluating a tool, some prominent factors are task completion time and task completion correctness [9]. JavaScript, widely used to produce interactive web-applications, was the chosen language. For the overall framework, **React**, an open-source web framework, was chosen, being used in conjunction with **Material-UI**, a set of components easing the process of the design aspects of a system. For InfoVis techniques, **Recharts** was used alongside **Deck.gl**¹.

The visualization provides the user with an icon giving access to a sidebar (see Fig. 2), which can be docked to the right or left side. The first option is a selection with the available techniques. Other options are organized in sections:

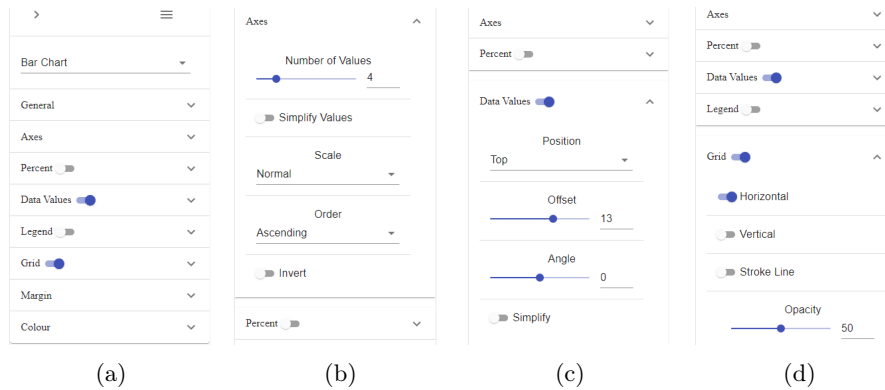


Fig. 2: Personalization sidebar. a) Default state. b) Axes. c) Data values. d) Grid.

General – General definitions, such as **height**. An option allows data to be **grouped** in one graphic or distributed among several (as dimensions on the same graphic encourage analysis and comparison, while separation encourages monitoring [10]). For line/area charts, we can change the type of **interpolation**. For line charts, the aid of using dots and the accessibility option of tuning the **line width** exists. For bar/area plots, there is the option to display the values **stacked**. For geovisualisations, the user can change the **background map** between several options (normal, terrain, relief, satellite, or none). There’s also an option to show **borders** and **text layers** (e.g. country districts’ borders and names); this option can be configured according to the scope of the visualization.

Axes (for graphs contained in a Cartesian plane) – Options include the **number of values** in the numerical axis, and the option to **simplify** those

¹ Official websites: reactjs.org, material-ui.com, recharts.org, deck.gl

values (for example showing 5500 as 5.5k). It is also possible to change the **scale** between normal and logarithmic and, when possible, to choose the **order** in which data is displayed. It is also possible to **invert axis** (for example, turning a vertical bar chart into a horizontal one).

Data Values – When activated, exact values are shown directly onto the representation. Options include changing the **position** where values are displayed relative to the graphic elements and adjusting the **displacement** and **angle**. Another option is to **simplify** the displayed values.

Percentage – Changes visual representation to a percentage. Options include to display the percentage in **relation to the max value** on a representation, to display the percentage in **relation to the total value**, or to display the percentage in **relation to a custom value**.

Grid, Legend, and Margin – There is an option to toggle **grid lines**; **horizontal**, **vertical** or both, to change the **grid opacity** and to have a **stroke line**. The graphic legend (hide or show the legend and to tweak its **position** and **alignment**); and the representation margins.

Color – Colors serve different purposes in visualizations (for instance, a sequential color scheme and qualitative color scheme). A user can choose to change the whole color scheme of a platform or that of individual graphics – this option can be very useful for users who suffer from some type of color vision deficiency [13]. It is also possible to alter the **opacity**.

5 Experimental Setup and Results

For each family, tasks were given to participants, consisting of a multiple-choice question related to the data represented in a graphic. The tasks were conceived to offer different complexity levels and could be completed without using the personalization options. The taxonomy for these tasks was introduced in [1].

The test started with a demographic survey, followed by questions using a five-value scale, collecting data on participant’s experience with data analysis and InfoVis techniques. After completing the tasks for each visualization, users were asked the preferred visual representation, and an open question to express opinions and problems found. There were a total of 18 tasks. After completion, a post-survey was presented to inquire about the usability of the options, including questions about being allowed to display the same data in different graphics and asking if the user interacted with the graphic or used the personalization options to complete the tasks. The metrics collected were time, type of graphic selected, and the number of times each option was used.

The study comprised 37 participants, 21 male and 16 female, with ages ranging from 18 to 64 years old. 24 participants had a college graduation, of which 11 had a master or doctorate, and the remaining 13 had basic education. Participants rated themselves on handling data through graphic visualizations, with an average of 3.65 out of 5. They showed a strong opinion towards having the option to personalize their platforms and systems, with an average of 4.38 out of 5, and 4.24 out of 5 in relation to the personalization of graphic visualizations.

Time and Accuracy – There was an average of 15.3 successfully completed tasks (correct answers) per participant, which translates to an accuracy of 84.8%. The time taken to complete the study (not including initial and final questionnaires) was on average 21 minutes and 24 seconds.

Graph selection – 6 participants did not change the default visualization. Of the 31 participants that did, 7 only changed it between 1 and 5 times, while the remaining 24 changed over 5 times, and 9 visited all graphics available on the study. Results show that while participants felt the default graphic was the most useful one for each task and graphic family, always being the most voted (average of 64%), other selections were also popular.

Personalization Options – 9 out of the 37 participants did not make any personalization. These 9 participants had a completion time average of 13m,57s, which is considerably lower than the global average (21m,24s). However, these participants also showed lower accuracy compared to the global average, a value of 80.3% (14.5 out of 18). Some participants opted to only use the options that directly aided on the task completion; others explored the “aesthetic” options as well. This is probably the cause of a high discrepancy between the number of personalization options used by participants. Nonetheless, the options were changed a total of 2703 times, which translates to an average of 73 changes per user. The most popular option was to show data values (29% of changes) directly on the visualization. By order of popularity, other options were: grouped (13%), percentage (11%), order (9%), number of values on an axis (9%), invert axis (6%), stacked (4%) and simplify data values (3%). Grid options were not toggled as much as expected, probably due to it being enabled by default. Users found the interface pleasing with a score of 4.16 (out of 5) and that most people would have an easy time learning to use it, with a score of 4.29, they also found the personalization options explicit and overall useful, with a score of 4.19.

6 Conclusions and Future Work

Results show that there is usefulness in giving users the option to personalize visualizations. As shown, personalization is desired by the common user. We confirmed the usefulness of some options and proposed new ones with potential. We also show that a higher successful task rate can be achieved with some options, and although users that did not use options completed tasks faster with a slightly lower accuracy rate, it did not account for the fact that users were not yet familiar with the system. This paper hopes to encourage and motivate building of systems having personalization in mind from the start. It continues research on forms of personalizing visualizations, and proposes a framework that can be included in web platforms for translating data into visualizations.

While the results are very promising, this work can be further improved by several factors. Users tend to use the same options from graphic to graphic, and on big platforms with many dashboards, personalizing each graphic might become a tedious process. Such a fact opens up the possibilities for personalization-based adaptation. The framework was conceptualized to facilitate the process of

data exploration for the user, but also to ease the process of introducing visualizations by developers. So, another interesting path for future work is to conduct tests to prove the framework effectiveness from the programmer's perspective. There is also the perspective of creating mechanisms to provide the user with the power of making their own queries to the database and, in conjunction with the proposed framework, create their own dashboards and similar solutions.

Acknowledgements This work was financially supported by Base Funding - UIDB/00027/2020 of the Artificial Intelligence and Computer Science Laboratory - LIACC - funded by national funds through the FCT/MCTES (PIDDAC).

References

1. Amar, R., Eagan, J., Stasko, J.: Low-level components of analytic activity in information visualization. In: IEEE Symposium on Information Visualization, 2005. INFOVIS 2005. pp. 111–117. IEEE (2005)
2. Chittaro, L.: Information visualization and its application to medicine. *Artificial Intelligence in Medicine* **22**, 81–88 (5 2001)
3. Keim, D.: Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics* **8**, 1–8 (1 2002)
4. Koh, L.C., Slingsby, A., Dykes, J., Kam, T.S.: Developing and applying a user-centered model for the design and implementation of information visualization tools. pp. 90–95. IEEE (7 2011)
5. Kong, N., Agrawala, M.: Graphical overlays: Using layered elements to aid chart reading. *IEEE Transactions on Visualization and Computer Graphics* **18**, 2631–2638 (12 2012)
6. Lallé, S., Conati, C., Carenini, G.: Prediction of individual learning curves across information visualizations. *User Modeling and User-Adapted Interaction* **26**, 307–345 (10 2016)
7. Liu, S., Cui, W., Wu, Y., Liu, M.: A survey on information visualization: recent advances and challenges. *The Visual Computer* **30**, 1373–1393 (12 2014)
8. Moere, A.V., Purchase, H.: On the role of design in information visualization. *Information Visualization* **10**, 356–371 (10 2011)
9. Nazemi, K., Burkhardt, D., Hoppe, D., Nazemi, M., Kohlhammer, J.: Web-based evaluation of information visualization. *Procedia Manufacturing* **3**, 5527–5534 (2015)
10. Poetzsch, T., Germanakos, P., Huestegge, L.: Toward a taxonomy for adaptive data visualization in analytics applications. *Frontiers in Artificial Intelligence* **3** (3 2020)
11. Steichen, B., Fu, B.: Towards adaptive information visualization - a study of information visualization aids and the role of user cognitive style. *Frontiers in Artificial Intelligence* **2** (11 2019)
12. Toker, D., Conati, C., Steichen, B., Carenini, G.: Individual user characteristics and information visualization. pp. 295–304. ACM (4 2013)
13. Wexler, S., Shaffer, J., Cotgreave, A.: *The big book of dashboards: visualizing your data using real-world business scenarios*. John Wiley & Sons (2017)
14. Yi, J.S., Kang, Y., Stasko, J., Jacko, J.A.: Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics* **13**, 1224–1231 (11 2007)