

Toward Integrated Exploration and Manipulation of Data Attributes in Graphs

Stefan Gladisch*
University of Rostock

Christian Tominski†
University of Rostock

ABSTRACT

Data exploration and manipulation are traditionally considered independently. In this work, we argue to integrate exploration and manipulation in order to smooth user workflows. Our focus is on changing the visual encoding of graph attributes as an exploration task and on changing attribute values as a manipulation task. We present a novel approach for integrating both tasks via a unified visual interface. This interface is based on the classic node-link diagram extended by radial menus and touch gestures for interaction. Our integrated solution provides the functionality for changing both the visual encoding and the data values. Hence, switching between different software tools is no longer necessary.

1 INTRODUCTION

In many applications data exploration and manipulation are key tasks. To accomplish these tasks, interactive, visual solutions are widely used as a basis, because they offer visual access to the data and also an interface between the user and the computer. However, traditionally exploration and manipulation are considered independently and carried out using separate tools.

In previous work [3], we gave a first example indicating that there is potential in integrating exploration and manipulation. While our previous work addressed the *graph structure*, the present work is going to investigate the exploration and manipulation of *data attributes* associated with graph elements.

We start with motivating why integrating exploration and manipulation makes sense and discuss challenges. Then we present work in progress on how the visual encoding of the data and the data themselves can be adjusted using a novel integrated interface.

2 MOTIVATION

Why should we care about integrating exploration and manipulation? Imagine yourself exploring a node-link diagram of a graph. During exploration you change parameters of the visual encoding to gain different views on the data. Through these changes you stumble upon a node that obviously has erroneous attribute values. As a consequence you switch to a data editing tool that allows you to correct the values. Afterward you go back to the node-link diagram and continue exploring it.

The problem with this traditional procedure is that the switch between exploration and manipulation and also changing visual encoding parameters through external GUI controls interrupts the user's workflow and produces unnecessarily high cognitive load. As different tools typically use different visual layouts of the data, users have to locate the data item to be corrected again when switching from exploration to manipulation. Furthermore, it can be necessary to switch back and forth multiple times to acquire enough information about the data (e.g. values in the neighborhood, extreme values, value distribution) to carry out the correction properly.

*e-mail: stefan.gladisch@uni-rostock.de

†e-mail: christian.tominski@uni-rostock.de

From these considerations we see that exploration and manipulation are not uncoupled. Data exploration can lead to data manipulation and the other way around. We think that by integrating exploration and manipulation we can smooth such workflows.

Integrating exploration and manipulation is a challenging problem. The reason is that an integrated approach has to cover all interaction tasks that are otherwise offered in exploration and manipulation tools individually. With the increased number of tasks comes the need for a carefully designed user interface allowing easy access to the functionalities needed. Further, the user must at all times be aware of the effect of an interaction operation: Does it serve exploration or manipulation? With these requirements in mind, we are working on a solution as presented next.

3 APPROACH

Exploring and manipulating graphs is a broad topic [6]. In fact, it is too broad to be addressed here in its entirety. Therefore, we restrict our consideration to the following user tasks:

1. Change the visual encoding of node attributes
2. Change the data values associated with nodes

According to the model of visualization exploration [4], task 1 can be characterized as an exploration task. Changing values of a data set as for task 2 is part of a data manipulation process.

The starting point for our work are graphs with data attributes also known as multivariate graphs [5]. Attributes can be quantitative or categorical. As a common ground for exploration and manipulation, we use a node-link diagram to visualize the graph structure. Data attributes of nodes are visually encoded by varying the size and color of nodes. Additionally, small symbols can be attached to nodes to visualize categorical data values. To facilitate standard interaction needed for data exploration, the visual representation is embedded in a zoomable visualization space.

Taking first steps toward integrated exploration and manipulation according to the aforementioned two tasks, we developed a unified visual interface. To support these tasks and to communicate their differences, we need an interface structured according to exploration and manipulation. The interface's functionality must be self-revealing and the interaction should be smooth requiring only little effort from the user.

On the visual side of our interface, we build on the concept of radial menus. The interaction side is served by modern touch interaction. In the following we explain our interface in detail and indicate how users can apply it to change the visual encoding and change data values in an integrated fashion.

Interface Design The zoomable node-link diagram is the basis for our integrated approach. We use direct interaction and radial menus to enhance the node-link diagram with the functionality required for our purposes. The general idea is to use touch gestures on the node-link diagram to invoke and interact with the radial menus. To simplify these interaction, we limit our design to tap, long tap, drag, and pinch gestures.

There are two different menus: One menu allows users to change the visual encoding of nodes, another supports changing the attribute values of nodes. To let the user "feel" the difference between

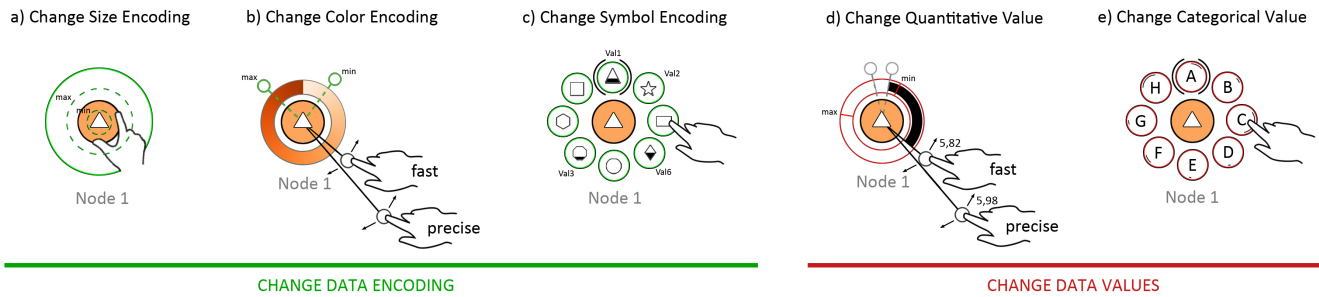


Figure 1: Interactive controls for changing the visual encoding (exploration task) and the data attribute values (manipulation task).

both menus, we use the following interaction concept. A *short tap* on *any node* invokes the menu for changing the visual encoding of node attributes. A *long tap* on a *specific node* invokes the menu for changing the data values of the tapped node.

Once invoked, the radial menu is centered on the selected node. We maintain a view to the node and its connected edges by leaving free space in the center of the menu. This way, the menu (where the interaction takes place) and the visualization (where the effect will become visible) are tightly integrated. The short distances in the radial menus further reduce the need for finger movement, which is beneficial for touch interaction.

The radial menus can accommodate up to 8 visual entries. Scrolling and nesting are applied where more than 8 logical menu entries are needed. According to existing conventions for touch interaction we ensure a minimal size of 10 mm for entries and preserve sufficient gaps between them. To clearly communicate the purpose of a menu, we use different outline colors: Green for the menu changing the visual encoding and signaling red for the menu for manipulating data values. Furthermore, visual feedback is provided via highlighting and through animation to facilitate *fluid interaction* [1]. Labeled menu options, a low depth of the menu, and the possibility to navigate to sibling entries, all strive to provide what has been termed *effective view navigation* [2].

While the idea of using radial menus centered on nodes is our general approach, it has to be refined to support either changing the visual encoding or changing data values. To this end, the menus are equipped with dedicated entries and controls to be described next.

Change Visual Encoding The radial menu for changing the visual encoding presents at its first level the visual variables available (size, color, symbol). Upon selection of a visual variable V , the second level of the menu is presented. It contains the data attributes that are compatible to be encoded with V . Selecting a data attribute D establishes a visual mapping $D \rightarrow V$ and the entire node-link visualization will be updated accordingly.

The final step is to adjust the mapping. Depending on the visual variable being used, different interactive controls are provided to carry out the adjustment. A direct pinch gesture is applied to enlarge or shrink the representation of nodes (Fig. 1 a). A simple adjustment of the color coding can be realized by dragging handles of a radial color scale control (Fig. 1 b). In order to let users change the mapping of categorical values to symbols, we show another level of the radial menu from which the user simply chooses one of the symbols available (Fig. 1 c).

With these controls, we enable the user to explore different visual encodings of different graph attributes. The interaction is carried out directly with the radial menus embedded in the node-link diagram. Indirections to external GUI elements are not necessary.

Change Data Values This menu contains a list of the attributes available for the selected node. By tapping an attribute the user expresses the intent to edit it. Depending on whether the attribute is quantitative or categorical, different editing controls will

be presented. For editing quantitative attributes, we designed a radial bar chart showing the attribute’s value range as well as the current data value of the node being edited (Fig. 1 d). By dragging the handles of the bar chart, the user can manipulate the data value at varying degrees of precision. If a categorical attribute has been selected, a menu level opens up for the user to select a new categorical value from the value range of the attribute (Fig. 1 e).

The described controls provide additional context information to support the user in making informed decisions during data exploration and manipulation. The previous paragraphs provided a brief overview of the functionality of our approach. As it is not possible to discuss all details here, it is planned to demo our solution at the poster session and discuss directly with the attendees.

4 SUMMARY AND FUTURE WORK

In this work, we argued for an integrated exploration and manipulation process. As a preliminary result, we presented a novel integrated visual interface for changing both the visual encoding of data attributes (exploration task) and the data values themselves (manipulation task). So far, we support basic operations using radial menus and dedicated controls embedded directly into the visualization. Touch gestures are employed to simplify interaction.

For future work we plan to extend our approach to more complex exploration and manipulation tasks. Simple extensions would be to support additional visual variables or the manipulation of groups of nodes. Another interesting question is to investigate alternative visual representations (e.g., matrix views) as the common basis for exploration and manipulation. We can imagine that matrices might be more suited when it comes to working with edges of graphs. It is also necessary to address scalability issues to support graphs with a large number of data attributes. Finally, we have to carry out user studies to evaluate the usability of our design.

REFERENCES

- [1] N. Elmqvist, A. Vande Moere, H.-C. Jetter, D. Cernea, H. Reiterer, and T. J. Jankun-Kelly. Fluid Interaction for Information Visualization. *Information Visualization*, 10(4):327–340, 2011.
- [2] G. W. Furnas. Effective View Navigation. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 367–374. ACM, 1997.
- [3] S. Gladisch, H. Schumann, M. Ernst, G. Füllen, and C. Tominski. Semi-Automatic Editing of Graphs with Customized Layouts. *Computer Graphics Forum*, 33(3):381–390, 2014.
- [4] T. J. Jankun-Kelly, K.-L. Ma, and M. Gertz. A model and framework for visualization exploration. *IEEE Trans. Vis. Comput. Graph.*, 13(2):357–369, 2007.
- [5] A. Kerren, P. Helen C, and M. O. Ward. *Multivariate Network Visualization*. Springer International Publishing, 2013.
- [6] M. J. McGuffin and I. Jurisica. Interaction Techniques for Selecting and Manipulating Subgraphs in Network Visualizations. *IEEE Trans. Vis. Comput. Graph.*, 15(6):937–944, 2009.