Mapping Tasks to Interactions for Graph Exploration and Editing

Stefan Gladisch[*] University of Rostock
Ulrike Kister[†] Technische Universität Dresden
Christian Tominski[‡] University of Rostock
Raimund Dachselt[§] Technische Universität Dresden
Heidrun Schumann[¶] University of Rostock

Abstract

Graph exploration and graph editing are still mostly considered independently and the available software solutions are typically not designed for today's interactive surfaces such as tablets or tabletops. This work is a step toward interactive systems for both graph exploration and graph editing. We deal with three questions. What tasks need to be supported? What interactions can be used? How to can tasks be mapped to interactions? We contribute a systematic review for the first two questions and present a formal mapping approach to address the third.

1 Introduction

Working with graphs includes tasks for exploring *and* editing the data. However, we are lacking visualization systems that support both aspects to the same extent. Users often have to operate two systems side-by-side and switch between them frequently, which complicates their work. Another drawback of existing systems is that they still mostly rely on mouse and keyboard interaction and do not take advantage of today's interactive surfaces. Our assumption is that seamlessly integrating graph exploration *and* graph editing and utilizing interactive surfaces would outperform existing solutions. In this light, three questions need to be addressed:

1. What tasks need to be supported?
2. What interactions can be used?
3. How to map tasks to interactions?

Here, we present first answers to these questions based on a recent technical report [2]. Our goal is to discuss the developed ideas and concepts with VIS attendees.

2 Tasks for Graph Exploration and Graph Editing

An integrated graph exploration-and-editing system needs to support a variety of tasks. To develop a systematic view on what tasks are relevant, we analyzed existing solutions for graph exploration (Gephi, Tulip, Cytoscape, Pajek, Nodes3D, CGV) and graph editing (yEd Graph Editor, MS Visio, GoVisual Diagram Editor, Dia Diagram Editor, Visual Paradigm for UML, Enterprise Architect). To categorize the tasks, we follow Yi et al. [4] for the exploration part and define new categories for the editing part. Next we briefly summarize the tasks supported by the analyzed systems.

Tasks of Graph Exploration

- Select: Tasks to mark nodes or edges of interest and to keep track of them.
- Explore: Tasks to examine global and local characteristics of the graph, for instance a subgraph of interest.
- Reconfigure: Tasks to create a different perspective on a subset of the data, for instance by applying a different layout.
- Encode: Tasks to alter the visual encoding of graph elements and associated data attributes.
- Abstract & elaborate: Tasks to adjust the level of abstraction of a graph representation.
- Filter: Tasks to change the set of displayed nodes and edges based on specific conditions.
- Connect: Tasks to establish relationships among nodes, edges, and associated data attributes, and sets thereof.

Tasks of Graph Editing

- Create: Tasks to create empty documents for new graph data sets.
- Insert: Tasks to add new nodes, edges, or data attributes to a graph.
- Delete: Tasks to remove existing nodes, edges, or data attributes from a graph.
- Update: Tasks to change attribute values of existing nodes or edges of a graph.
- Navigate: Tasks to navigate to different subgraphs in order to edit them.
- Select: Tasks to mark nodes or edges of interest to be affected by an edit operation.
- Miscellaneous: Additional tasks that do not fit in the above categories, for example cut and copy nodes or edges.

Most of the analyzed systems cover several categories, but there is no system that comprehensively supports all of them. Noteworthy, the category select exists for exploration and editing. The explore and navigate categories are quite similar as well. This indicates that basic selection and navigation are cross-cutting when working with graphs.

3 Interaction Vocabulary

After identifying relevant tasks for graph exploration and graph editing, the next step is to determine what interactions can be used to accomplish these tasks. An interaction vocabulary is needed. Typically, graph exploration and graph editing already use a high degree of interaction on their own. An integrated exploration-and-editing system therefore requires a rich interaction vocabulary.

Today's interactive surfaces with touch, pen, and tangible interaction (see Figure 1) provide such rich interaction vocabularies. Compared to traditional mouse and keyboard interaction, these interaction modalities provide more degrees of freedom and thus more distinguishable interactions, which is necessary in our scenario. Degrees of freedom include the input device (e.g., different fingers, pens, or tangibles), the user who is interacting (as identified by appropriate sensory devices), the applied gesture (e.g., tap vs. long press in touch interaction), and the affected object or world feature (e.g., tap on node vs. tap on edge). The interaction with touch, pen, and tangibles is also more reality-oriented and direct, which can contribute to a more effective graph exploration and editing.

There are further aspects that need to be taken into account when compiling an interaction vocabulary. We have to analyze whether the simultaneous use of different interaction modalities is suitable for graph exploration and editing. Moreover, there might be interaction modalities that are particularly useful (or unfitting) for specific tasks. A first indication that pen interaction is especially suitable

[*]e-mail: stefan.gladisch@uni-rostock.de
[†]e-mail: ulrike.kister@tu-dresden.de
[‡]e-mail: ct@informatik.uni-rostock.de
[§]e-mail: raimund.dachselt@tu-dresden.de
[¶]e-mail: schumann@informatik.uni-rostock.de

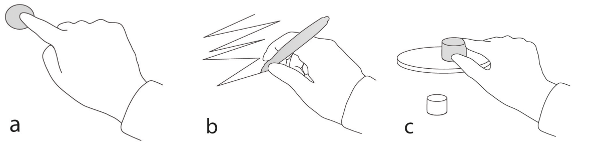

Figure 1: Common interaction modalities for interactive surfaces: a) touch interaction, b) pen interaction, and c) tangible interaction.

for graph editing tasks is presented in [1]. To give precise answers to these questions, more research is needed. Our future work is headed in this direction.

4 Mapping Tasks to Interactions

Having defined a set of exploration and editing tasks $T = \{t_1, t_2, ..., t_k\}$ as well as an interaction vocabulary $I = \{i_1, i_2, ..., i_l\}$, the next step is to decide which interaction shall be used to accomplish which task. To this end, a mapping $m : T \rightarrow I$ from tasks to interactions is needed. To avoid ambiguities, m must be injective. Furthermore, m has to be defined under consideration of accepted interaction criteria. The human-computer interaction literature lists a number of such criteria, including:

- Predictability: Interaction should always exhibit deterministic behavior for the user.
- Consistency: Similar interactions should be used for similar functions.
- Familiarity: Interaction should map as closely as possible to the real world or to known metaphors.
- Generalizability: Interactions should be as specific to the context as necessary, but as basic as possible to be reusable.
- Viscosity: Frequently used functions should map to interactions with lowest effort.
- Recoverability: Users should be able to easily undo and redo interactions.
- Directness: Interaction should rather be directly applied to the affected virtual object than on separate control panels.
- Continuity: Combination of basic interaction steps should be possible to form an interaction flow without discontinuities.

A difficulty is that these criteria are not free of conflicts. Moreover, the suitability of a mapping m with respect to a criterion is often difficult to determine and has to be estimated by the system designer.

For an example, we look at possible interactions for a basic zoom task. On interactive surfaces, zoom is usually accomplished with touch interaction using a pinch gesture (see Figure 2 a)). So, using pinch for zooming would yield a high quality in terms of the familiarity criterion. A recent study [3] found that zooming can be carried out much faster with tangible interaction where the interactive surface is moved in physical space (see Figure 2 b)). This means that tangible interaction would yield a better quality with respect to the viscosity criterion, but less quality concerning familiarity. In this case, familiarity and viscosity are conflicting. Even if there were only these two criteria to consider, it would not be clear how to decide which interaction to use, because there is no consensus about which criterion has a higher priority.

The larger number of tasks, the richer interaction vocabulary, and the conflicting criteria lead to a vast number of possible mappings $m : T \rightarrow I$. The fact that all task-interaction pairs must be considered as a whole make it very difficult for a system designer to set up a suitable mapping. For this reason, we propose the following formal approach.

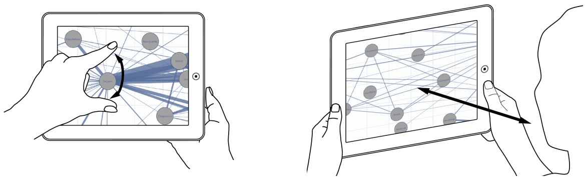

Figure 2: Different interactions for zoom: a) touch interaction: pinch gesture and b) tangible interaction: moving the device itself.

Let M be the set of all possible mappings $T \rightarrow I$ and C be a set of interaction criteria. The quality of a mapping $m \in M$ concerning a specific criteria $c \in C$ can be expressed with a quality function $q : M \times C \rightarrow [0,1]$. Then, the overall quality of a mapping m concerning C can be expressed with the normalized sum $\hat{q} = \frac{\sum_{i=1}^{n} q(m,c_i)}{n}$. Inserting weights allows us to prioritize the interaction criteria leading to a weighted variant of the overall quality $\hat{q} = \frac{\sum_{i=1}^{n} \alpha_i \cdot q(m,c_i)}{n}$, with $\alpha_i \in [0,1]$. With the definitions of the quality function q and the weights α_i, finding a "good" or "the best" mapping m can now be handled as an optimization problem where $\hat{q}$ is maximized. An optimization algorithm can be applied to converge to a mapping with an optimal overall quality.

To come up with an application-specific task-interaction mapping for graph exploration and graph editing, the following steps need to be carried out. First, k tasks from the identified categories must be specified depending on the application scenario. Next, interaction modalitities along with a set of $l > k$ distinguishable interactions have to be selected according to devices available in the application context. Then, relevant interaction criteria for the application have to be determined and dedicated quality functions have to be defined. Finally, the formal mapping approach can be applied. The result would be a quality task-interaction mapping that can guide the implementation of an integrated graph exploration-and-editing system.

5 Summary

In this work, we argue for supporting graph exploration and editing in a single interactive system. We categorized tasks that should be supported, indicated the benefit of today's interactive surfaces for accomplishing these tasks, and proposed a formal approach for mapping tasks to interactions. More details are available in [2].

Finding suitable quality functions and weights for graph exploration and graph editing on interactive surfaces is still a complex endeavor. Based on our formal approach, we hope to be able to reduce the complexity allowing us to come up with practical solutions in the future to be evaluated in real user experiments.

Acknowledgements

This research has been supported by the German Research Foundation (DFG) in the context of the project GEMS – graph exploration and manipulation on interactive surfaces (DA 1319/3-1).

References

[1] M. Frisch, J. Heydekorn, and R. Dachselt. Investigating Multi-touch and Pen Gestures for Diagram Editing on Interactive Surfaces. In *Proc. of ITS*. ACM, 2009.

[2] S. Gladisch, U. Kister, C. Tominski, R. Dachselt, and H. Schumann. Mapping Tasks to Interactions for Graph Exploration and Graph Editing on Interactive Surfaces. arXiv:1504.07844 [cs.HC], 2015.

[3] M. Spindler, M. Schuessler, M. Martsch, and R. Dachselt. Pinch-Drag-Flick vs. Spatial Input: Rethinking Zoom & Pan on Mobile Displays. In *Proc. of CHI*. ACM, 2014.

[4] J. S. Yi, Y. ah Kang, J. T. Stasko, and J. A. Jacko. Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *TVCG*, 13(6), 2007.