
GLOs: Graph-Level Operations for Exploratory Network Visualization

Charles D. Stolper
College of Computing
Georgia Tech, USA
chadstolper@gatech.edu

Florian Foerster
College of Computing
Georgia Tech, USA
florian.foerster@gatech.edu

Minsuk Kahng
College of Computing
Georgia Tech, USA
kahng@gatech.edu

Zhiyuan Lin
College of Computing
Georgia Tech, USA
zlin48@gatech.edu

Aakash Goel
College of Computing
Georgia Tech, USA
aakashgoel@gatech.edu

John Stasko
College of Computing
Georgia Tech, USA
stasko@cc.gatech.edu

Duen Horng Chau
College of Computing
Georgia Tech, USA
polo@gatech.edu

Abstract

There is a wealth of visualization techniques available for graph and network visualization. However, each of these techniques was designed for a specific task. Many graph visualization techniques and the transitions between them can be specified using a set of operations on the visualization elements such as positioning or resizing nodes, showing or hiding edges, or showing or hiding axes. We term these operations *Graph-Level Operations* or *GLOs*. Our goal is to identify and provide a comprehensive set of these operations in order to better support the broadest range of graph and network analysis tasks. Here we present early results of our work, including a preliminary set of operations and an example application of GLOs in transitioning between familiar graph visualization techniques.

Author Keywords

Graphs; visualization techniques; operations

ACM Classification Keywords

H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous.

General Terms

Design, Human Factors

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). Copyright is held by the author/owner(s).
CHI 2014, April 26–May 1, 2014, Toronto, Ontario, Canada.
ACM 978-1-4503-2474-8/14/04.
<http://dx.doi.org/10.1145/2559206.2581239>

Introduction

More and more often, analysts must understand connections within their data. The field of graph visualization has developed around this task, providing a wealth of techniques for communicating and exploring graph and network data sets. These techniques range from force-directed node-link diagrams to PivotGraphs [16] to matrix displays to more traditional visualization techniques such as scatterplots.

For the most part, each of these techniques is most effective for a single task. To accomplish a series of tasks, one can imagine transitioning between these different techniques. For example, starting with a force-directed layout, one could reposition the x coordinate of each node based on its betweenness centrality. One could then reposition the y coordinate of each node based on its degree. Adding x - and y -axes to the layout and hiding the links between nodes leaves us with a scatterplot of betweenness centrality vs. degree.

The above sequence demonstrates how these transitions (and the techniques themselves) can be decomposed into generalizable, atomic *graph-level operations (GLOs)* such as repositioning by a variable or adding an axis or removing all links. We can define techniques by a set of GLOs and efficiently transition between existing techniques by means of GLOs.

A vocabulary of graph visualization operations provides analysts and graph visualization technique designers the potential to identify new and effective views of graph data through novel combinations of these operations. From the graph visualization system-builders' perspective, rather than implement every graph visualization technique, they can alternatively implement every GLO and still provide the same (and potentially greater) coverage.

In this extended abstract, we present the following contributions to HCI and graph visualization:

- We introduce *graph-level operations (GLOs)* as atomic operations on all or partial graph visualization elements;
- We enumerate a preliminary set of GLOs derived from a set of node-centric visualization techniques;
- We provide an example of transitioning between a series of existing graph visualization techniques by means of GLOs;
- We describe the requirements of user interface for exploring a graph and creating new graph visualization techniques using GLOs.

Related Work

Software systems and tools have been developed for graph analysis. There are many software libraries for visualization, such as JUNG [11]. Some visualization toolkits, such as Protovis [7] and D3 [5], provide a declarative language for developing new visualizations, including network visualizations. GUESS [1] defines languages for navigating graphs. However, these tools require users to have programming skills, which makes difficult for many analysts to use them. Some graphical tools, such as UCINet [4], Pajek [3], and Gephi [2], make it possible to visualize graphs without programming. NodeXL [13] enables analysts to do so in a commercial spreadsheet. These systems tend to provide only a small number of visualization techniques, and it is impossible to define new techniques.

There has been work on defining common operations for the process of visualizing data, but the operations in these work focused more on filtering and grouping data than on placing and aligning visual elements. Polaris [14]

formalized steps for visualizing multidimensional data from relational databases. The Ploceus [10] and Orion [8] projects proposed several operations and methods for manipulating and transforming relational data into graphs. In our work, we would like to define operations for placing and arranging nodes and edges in the graph in addition to selecting them from raw data sets.

Graph-Level Operations

We hypothesize that all graph visualization techniques, such as force-directed node-link diagram, PivotGraphs, and scatterplots, as well as the transitions between any pair can be decomposed into a set of atomic operations on visualized graphs. We define *Graph-Level Operations* (GLOs) as these atomic operations on some or all graph visualization elements of a representation. Theoretically, one can compose any graph visualization technique and transition between them using different combinations of GLOs.

By defining GLOs as atomic, we indicate that each GLO is independent from each other and represents a basic unit of operation for visualizing a graph. Similar to atoms themselves, GLOs can be further decomposed. However, we expect that such micro-operations do not represent meaningful operations for the task of analyzing a graph using visualization. For example, transitioning from a force-directed node-link diagram to scatterplot view of the same graph is *not* a GLO itself since it can be broken down into a set of smaller, basic operations such as transitioning x and y by attributes as we show below. These operations could be decomposed into even smaller operations such as calculating the value of the attribute for a node and editing the x or y position of the node according to the attribute's recalculated value. Although these micro-operations are important on an algorithmic

level, they would not add value for visualizing a graph from the user's perspective.

Graph-Level Operations can be defined broadly. Those operations can be not only modifying existing visual elements but also add or remove visual elements in a display. Take drawing an x axis as an example. Though this operation does not modify any of the existing visual elements (the nodes and links), it does introduce new visual elements into the visualization that, along with other visual elements, comprises a scatterplot.

Our Approach

In order to identify as many Graph-Level Operations as possible, we have followed a simple method. First, we identify as many different graph visualization techniques as possible. Second, we form all pairs of techniques. For each pair, we transition one technique to and from the other, recording each step. Each step of these transitions is a single GLO. As this is a work-in-progress, this method is still ongoing. We include here a preliminary set of GLOs that we have identified by applying the above method to four node-centric graph visualization techniques: force-directed node-link diagrams, semantic substrates [12], arc diagrams [15], and scatterplots.

The following GLOs are related to changing the *position* of nodes:

- Align Nodes {Left,Center,Right} on x
- Align Nodes {Top,Middle,Bottom} on y
- Evenly Position Nodes on x or y
- Apply a Force-Directed Algorithm to the Nodes

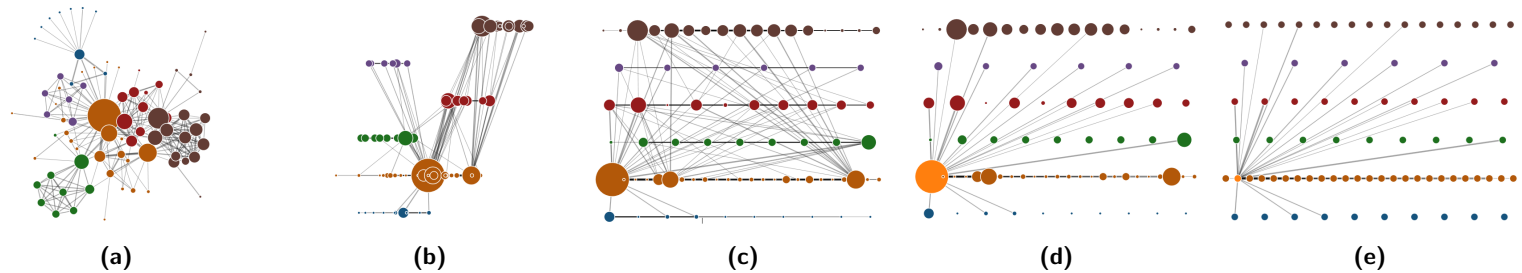


Figure 1: Results of the GLOs making up the transition from a force-directed layout to semantic substrates

The following GLOs position nodes utilizing their *attributes*:

- Substrate Nodes on x or y by {attribute}
- Position Nodes on x or y by {attribute}

Furthermore, there are GLOs related to changing the *properties* of node or edge glyphs:

- Size Nodes by a Constant or {attribute}
- Show or Hide Links
- Show Links as {Straight,Curved,Circles}

Finally, there are operations unrelated to the nodes and edges:

- Show/Hide x or y Axis

We have implemented these GLOs in a browser-based prototype system using D3.js [5].

GLOs in Practice

In this section, we demonstrate how transitions between graph visualization techniques can be defined by sets of graph-level operations (GLOs).

We begin with the layout in Figure 1a: the nodes in a graph are positioned according to a force-directed layout algorithm and are sized according to their degree. Figure 1b shows the result of applying our first graph-level operation (GLO): substrate nodes on y by category. This GLO positions nodes along the y dimension according to their categories (here double-encoded with color). Applying a GLO to evenly distribute the nodes of each category along the x axis results in Figure 1c. Next, we apply a GLO that adjusts the interaction to only show edges of one node at a time (Figure 1d). Finally, we apply a GLO to size the nodes to a fixed circle size as opposed to being sized by the value of a particular attribute in order to minimize node occlusion. This results in the semantic substrates visualization technique [12] (Figure 1e/2a).

Continuing, we apply a GLO to relatively position the nodes along the x axis according to the value of their betweenness centralities (Figure 2b). We can then apply a GLO to display the axis itself (Figure 2c). Applying a related set of GLOs to relatively position the nodes on the y axis by their degrees and display that axis results in the

displays in Figures 2d and 2e. Applying a GLO to hide all edges results in the scatterplot shown in Figure 2f.

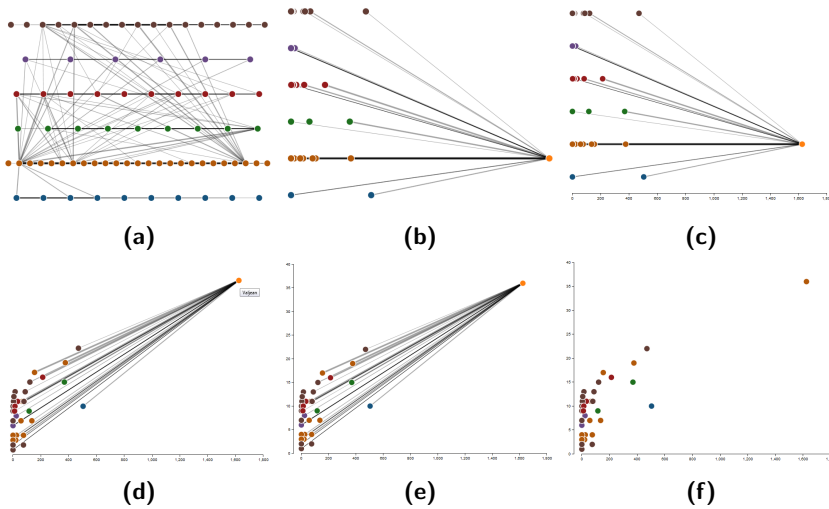


Figure 2: Results of the GLOs making up the transition from semantic substrates to a scatterplot

Thus, we have demonstrated transitions from a force-directed layout to semantic substrates to a scatterplot through a series of graph-level operations (GLOs).

User Interface

Any user interface for exploring graphs using GLOs should enable the user to transition between pre-defined techniques, such as the transitions we described above. However, we are especially interested in providing a user interface in order to enable network analysts to identify novel, effective graph visualization techniques.

Based on these two goals, we have identified a number of requirements for a user interface:

- Enable an analyst to apply individual GLOs to a graph
- Enable the analyst to highlight nodes or edges of interest in a graph in order to observe these through the course of the transitions
- Assist analysts by communicating potentially interesting GLOs
- Identify and communicate any past operations that no longer have any impact on visualization
- Enable an analyst to move backwards and forwards through the GLO history
- Enable an analyst to save a GLO history as a technique to apply to other graphs
- Enable an analyst to export an image of the current visualization along with its GLO history to enable the analyst to compare different techniques

Next Steps

As we move forward with this project, we will continue to grow and refine the set of graph-level operations by iteratively expanding our set of existing techniques beyond node-centric techniques. We will consider the case of node duplication, which we have identified as necessary for techniques such as matrix layouts [6] and parallel coordinates [9]. We will also consider node and edge aggregation, necessary for techniques such as PivotGraphs [16]. We will implement and refine the user interface that we discussed above through a user-centered design approach. Finally, we will conduct a study of GLOs' utility in both exploring graphs and discovering novel, effective graph visualization techniques.

Acknowledgements

The authors would like to thank the WIP reviewers for their helpful feedback. Funding was provided by the U.S. Army Research Office (ARO) and Defense Advanced Research Projects Agency (DARPA) under Contract Number W911NF-11-C-0088 and the XDATA Project. The content of the information in this document does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

- [1] Adar, E. GUESS: a language and interface for graph exploration. In *CHI '06*, ACM (New York, NY, USA, 2006), 791800.
- [2] Bastian, M., Heymann, S., and Jacomy, M. Gephi: An open source software for exploring and manipulating networks. In *ICWSM '09*, AAAI (2009), 361–362.
- [3] Batagelj, V., and Mrvar, A. Pajek analysis and visualization of large networks. In *Graph Drawing Software*, Mathematics and Visualization. Springer Berlin Heidelberg, 2004, 77–103.
- [4] Borgatti, S. P., Everett, M. G., and Freeman, L. C. Ucinet for windows: Software for social network analysis, 2002.
- [5] Bostock, M., Ogievetsky, V., and Heer, J. D3: Data-driven documents. *IEEE TVCG* 17, 12 (Dec. 2011), 2301–2309.
- [6] Fekete, J. Visualizing networks using adjacency matrices: Progresses and challenges. In *IEEE CAD/Graphics '09* (2009), 636–638.
- [7] Heer, J., and Bostock, M. Declarative language design for interactive visualization. *IEEE TVCG* 16, 6 (2010), 1149–1156.
- [8] Heer, J., and Perer, A. Orion: A system for modeling, transformation and visualization of multidimensional heterogeneous networks. In *VAST 2011* (2011), 51–60.
- [9] Inselberg, A. Multidimensional detective. In *IEEE Infovis 1997* (Oct. 1997), 100–107.
- [10] Liu, Z., Navathe, S., and Stasko, J. Network-based visual analysis of tabular data. In *VAST 2011* (Oct. 2011), 41–50.
- [11] O'Madadhain, J., Fisher, D., Smyth, P., White, S., and Boey, Y.-B. Analysis and visualization of network data using JUNG. *Journal of Statistical Software* 10, 2 (2005), 135.
- [12] Shneiderman, B., and Aris, A. Network visualization by semantic substrates. *IEEE TVCG* 12, 5 (2006), 733–740.
- [13] Smith, M. A., Shneiderman, B., Milic-Frayling, N., Mendes Rodrigues, E., Barash, V., Dunne, C., Capone, T., Perer, A., and Gleave, E. Analyzing (social media) networks with NodeXL. In *C&T '09*, ACM (New York, NY, USA, 2009), 255264.
- [14] Stolte, C., Tang, D., and Hanrahan, P. Polaris: a system for query, analysis, and visualization of multidimensional relational databases. *IEEE TVCG* 8, 1 (2002), 52–65.
- [15] Wattenberg, M. Arc diagrams: visualizing structure in strings. In *IEEE Infovis 2002* (2002), 110–116.
- [16] Wattenberg, M. Visual exploration of multivariate graphs. In *CHI '06*, ACM (New York, NY, USA, 2006), 811819.