Running head: IMPROVING GRAPH VISUALIZATION

Improving Graph Visualization

in TechNotes’ GraphView

for

Braden MacDonald

CEO

TechNotes

Vancouver, British Columbia

By

Konstantin Mestnikov

ENGL301 Technical Writing Student

July 20, 2022

Table of Contents

Abstract…………………….……………….…………………….………………iii

Introduction……………………….……………….……………...………………1

Data Section……………………….………………….…………...………………2

Conclusion ………….…….…………………….……………...…………………8

References ………..……….…………………….……………...…………………9

Figures and Tables

# **Introduction**

*Background on TechNotes database and graph view issues*

The Graph View in TechNotes allows for the data to be represented as a graph, where nodes and edges that stand for entities and relationships in the database. While potentially beneficial to browsing information efficiently and deriving new insights from the data, graphs can quickly become cluttered and hard to interpret.

Visualizations of graphs can quickly become complex and confusing especially when the graph connects entities of different types (companies, concepts, products, etc.) as is the case in TechNotes. Without properly handling the case when the number of edges becomes too large, proper database navigation and data exploration in the graph view become hard or even impossible.

*Purpose of this report*

This report aims at evaluating various possibilities for graph visualization in order to reduce graph cluttering, analyzing the feasibility of implementation of chosen graph decluttering features, and at proposing recommendations for Graph View product development based on the findings.

*Method of Inquiry*

Potential graph decluttering techniques to be tested in this report were identified via secondary sources or designed for the purposes of this study. In addition, X members of the general population responded to a survey designed to analyze how various ways of graph layout and design affect what the graphs convey in terms of aesthetics, information, and comprehension. Additionally, monetary and time expenditures for implementing the graph decluttering techniques were analyzed and estimated to assess feasibility and priorities of implementing the various techniques.

*Scope of the inquiry*

The inquiry will cover the following questions:

* how well does the currently used graph visualization conveys information and how is the graph visualization perceived?
* Which graph decluttering techniques would be most effective based on participants’ views?

**Data Section**

Several strategies to achieve graph decluttering are tested in this report, namely: 1) altered graph layout, 2) altered representation of edges, 3) higher-order graph modifications. The first set of strategies simply involve altering the layout of the graph as it is currently presented in Graph View of TechNotes. The altered layout is evaluated against the original layout. The second type of strategies targets the way edges are represented in the graph. Particularly, a type of edge bundling developed by Bach et al. and called confluence drawing is assessed. The last set of strategies involve higher-order graph modifications that are aimed at highlighting specific dimensions of a graph. Sorting nodes into communities algorithmically according to graph topology, highlighting cliques (tight interconnected clusters) or allowing to hide nodes of specific type are all assessed in an example use case of visualizing a research network.

**Alternative graph layout strategy**

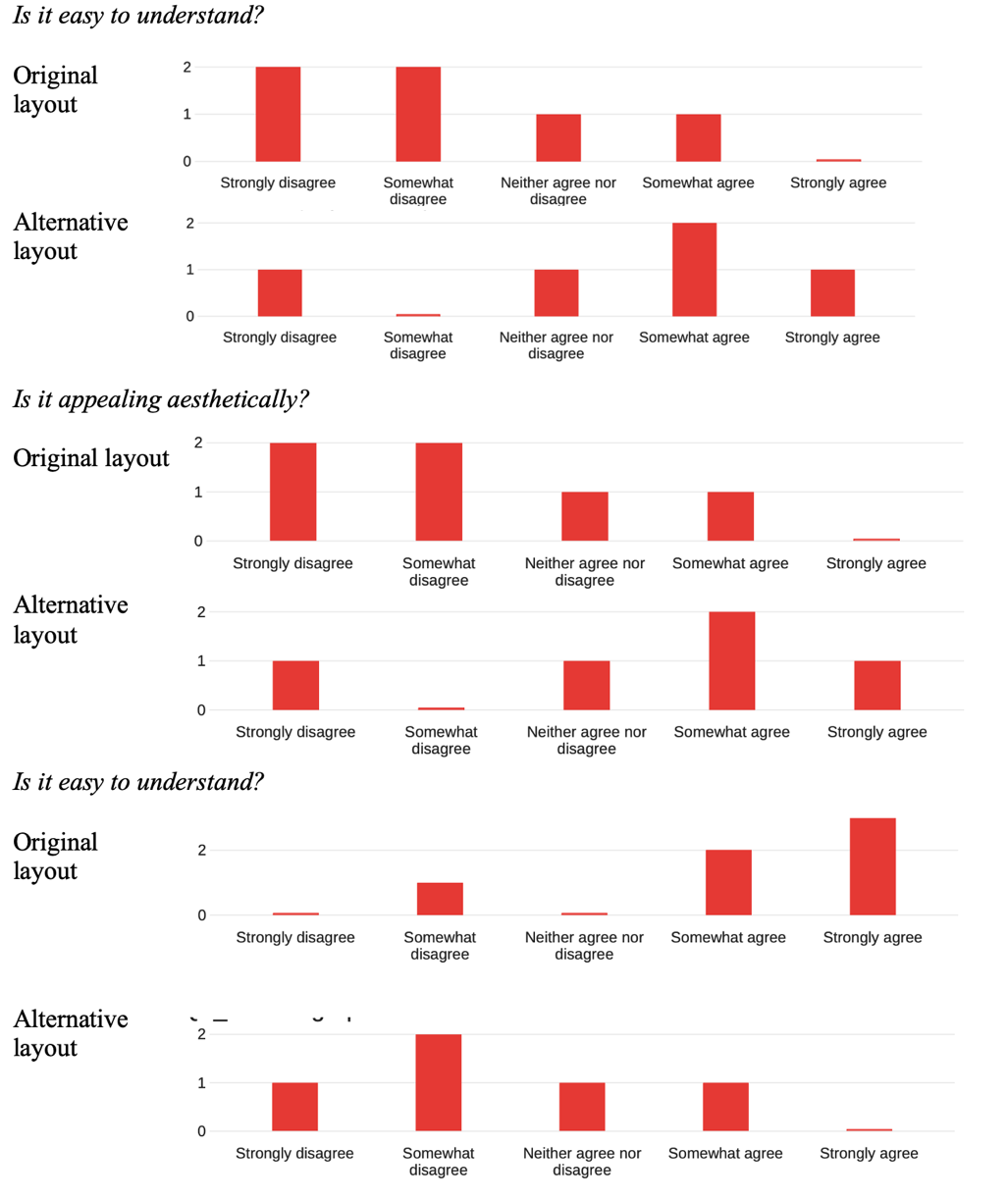
Current graph layout is computed using a standard forced layout algorithm. The forced algorithm does not consider the different node or relationship types that exist in TechNotes data, and thus produces a layout purely out of surface topology of the graph. This, for example, leads to node arrangements, in which TechConcepts hierarchy is not shown explicitly, or in which products, TechConcepts and companies are not arranged in any meaningful way relative to each other. To address these issues, an alternative layout is devised and contains the following features: 1) TechConcept hierarchy tree is put in a central region of the graph, 2) products are arranged on either side of the TechConcept tree, 3) manufacturers are arranged on both peripheries of the graph canvas. To test how this alternative layout compares to the original forced layout, several survey questions were designed that are aimed at assessing participants’ perception of aesthetic, complexity, and information factors.

Figure 1 Participants reply distributions for perception-related questions

*Survey results on effectiveness of decluttering techniques*

 As graph 1 shows, the graph layout is generally perceived as more aesthetically pleasing, easy to understand and less cluttered as the original layout. A further question was aimed at graph comprehension. The participants were asked to name the TechConcept that has the greatest number of products that belong to its type. While using the original layout 100% of participants responded “Lithium-ion cell”, 66% of participants responded the correct answer “Battery Energy Storage” when prompted with the alternative layout.

*Discussion of results*

In a striking result, a simple rearrangement of nodes produced a graph that performed better in all perception characteristics that were tested – aesthetics, cluttering and comprehensibility. Other than showing a clear advantage of the alternative graph layout, it shows that perhaps aesthetics and comprehensibility have an underlying relationship that can be further explored in future studies. More than that, alternating the layout allowed for an actual improvement of performance in terms of comprehension, as tested by the “Battery Energy Storage” question. That happened perhaps because arranging various TechNotes entities in a consistent way improved the big picture view and made it easier to distinguish various relationship types as well as amount of a particular entity type.

**Confluence drawings**

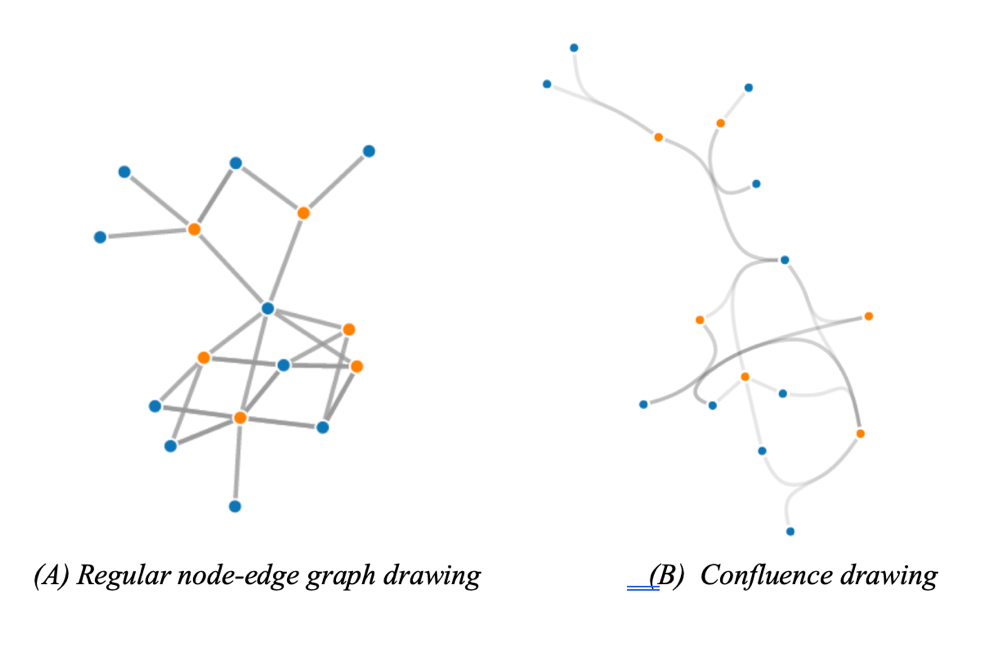
A different approach to graph decluttering tackles the way edges are represented in a graph. Because the maximum number of edges in a simple graph is proportional to the square of number of nodes in the graph, high edge density and high number of edge crossing are common problems that occur even in relatively small graphs. Various edge bundling techniques are developed to group edges together to reduce redundancy, edge crossing and overall cluttering. In this study, a relatively new way of edge bundling called confluence drawing developed by Bach et al. are assessed. In confluence drawings, edges are processed with an algorithm that bundles the edges and represents them as smooth curves in such a way that if any two edges are connected by one of such smooth curves, it means that they have an edge in between. This leads to a representation where a single curved edge may represent multiple edges at the same time (Figure 2).

Figure 2 Comparison of Regular and Confluence drawing graph representations (adapted from Bach et al.)

*Survey results on perceived information from graphs*

Confluence drawing were tested for comprehension by participants via two survey questions. In the first question, participants are presented with a simple node-edge graph without edge bundling and are asked a question. In the second question, participants are presented with a confluence drawing of the identical graph and are asked the same question. As can be seen from Table 1, 60% of participants answered the question correctly when presented with a regular graph, while only 20% answered correctly when presented with the confluence drawing version of the graph.

*Discussion of results*

Despite the appealing concept of confluence drawings, they fail to provide a less cluttered version of the graph while maintaining comprehensibility. The participants performed much better when presented with a regular graph, than confluence drawing graph, meaning perhaps that confluence drawings are harder to understand and/or take to get accustomed to. This can be explained by the fact that confluence drawing were originally developed for purposes of network science to represent larger graphs, where individual connections are important but not as important as overall structure of the network.

**Type hiding, cliques, and communities**

The last set of decluttering techniques relate to graph modifications that are specific to graphs that represent networks of individuals, companies, organizations. In this case, one is interested in asking questions about a network that refers to one type of entities as opposed to many. The techniques used in this section have already been implemented in the system, but their relative effectiveness has not been assessed. The first technique refers to simply hiding every other type of node other than the one under consideration, and thus is called node hiding. This technique can aid in focusing a particular node type and eliminate all other information altogether. The second technique employs node hiding and uses an algorithmic way to automatically detect communities within a given network using Louvain method. This technique is potentially beneficial as it gives a visual aid in the form of graph nodes coloured by communities for a quick grasp of the topology of the network. The last techniques involve detecting tight node clusters called cliques in addition to detecting communities and node hiding.

*Survey results on perceived information from graphs*

Survey questions were designed to assess difficulty and clutter level of the four graph options. Interestingly, the original layout and the clique layout are equally divided on both sides of the difficult-easy, cluttered-non-cluttered spectra. At the same time, the node-hiding and community versions of layout were in the middle of the spectra. Furthermore, the other questions of the survey aimed at assessing how the different graph layouts are perceived on the scale of usefulness in answering questions about finding different connections in the network. The clique and community layouts both outperformed the original and node-hiding layout.

*Discussion of results*

It is interesting to see how original layout and clique layout both occurs as most difficult and easy/cluttered and non-cluttered according to the survey results. It is possible that the level to which an individual perceives or understands the underlying graph representation determines their preference. In other words, a more complex layout (clique-based layout) may be seen as easy and neat only if one understands it well. So even a little gap in understanding the layout could significantly decrease the ease of perception. In this case, perhaps the middle options (node-hide and community-based) layouts can be fit for a wider range of audiences, as they avoid “all or none” problem with the clique layout.

Since community and clique-based layouts also performed better in terms of perceived usefulness, the community layout appears as a winner in terms of ease of use and usefulness. Its complexity is just right to be comprehensible and appealing and just right to be useful to answer network-based questions.

**Cost analysis of the tested decluttering techniques**

Since type-hiding, community detection and finding cliques are already implemented in the system, the cost of employing them is essentially 0. On the other hand, developing the alternative layout described in part a of the Data Section may take considerable time. That is because the layout may need to be adapted to varying and complex data types found in TechNotes and consideration of importance, space, edge crossing and etc. will be important in devising the corresponding algorithm. Thus, in our opinion, the layout may take approximately 1-1.5 month of work for a single developer. The implementation of confluence drawing may be more challenging for several reasons. First, the system on which GraphView is build may not allow such algorithm to be incorporated easily, and in such a case, the entire GraphView may need to be re-implemented using a more flexible framework. Secondly, the algorithm may be difficult to optimize for large-node networks and given that TechNotes database values fast response-speed, it may be challenging and time-consuming to achieve. In our opinion, it may take anywhere from 1 month, if no changes to the underlying system are necessary, anywhere to 3 months, if a new framework needs to be used and optimizations for performance made.

**Conclusion**

*Summary and interpretation of findings*

Several decluttering techniques were examined in this study. The alternative graph layout provides increased performance in graph comprehension as well as improved perception in terms of aesthetics, clutter level and comprehensibility. The confluence drawing approach, while having a good conceptual appeal, suffers from decreased comprehensibility. Among the tools for network analysis, the layout that includes node-hiding and community detection complements a balanced complexity of layout with perceived usefulness for answering network-related questions. On the other hand, the clique-based layout may appear as too complex despite highest ranking in perceived usefulness. The node-hiding only layout only shows marginal increase in performance compared to unaltered graph layout.

*Recommendations for TechNotes’ graph view*

Based on the findings as well as cost analysis, the following recommendations for GraphView improvement are made:

* Implementing the alternative graph is a single most impactful and cost-effective measure that can improve performance and perception of GraphView
* Making community-based layout a more prominent or default feature when showing entities of a single type can provide better user experience
* Improving clique-based layout by providing more intuitive documentation as to how to use it and making it an option only when the users need to ask specific questions about group clusters would make the best use of this complex layout

**References**

Bach, Benjamin, et al. “Towards Unambiguous Edge Bundling: Investigating Confluent Drawings for Network Visualization.” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, 2017, pp. 541–50. *Crossref*, https://doi.org/10.1109/tvcg.2016.2598958.