

# Glidgets: Interactive Glyphs for Exploring Dynamic Graphs

Brittany Kondo\*

Hrim Mehta†

Christopher Collins‡

University of Ontario Institute of Technology

## ABSTRACT

We present Glidgets, a combined direct manipulation and visualization technique for exploring and querying changes of elements in dynamic graphs. Traditional approaches provide an indirect time slider and employ visual cues such as global change highlighting. Our work merges temporal navigation and the visual representation of graph dynamics into interactive visual glyphs on nodes and edges. Our time line glyphs reveal the presence and absence of graph elements, and node degree. Using sketch-based interaction, the glyphs are used to create queries and navigate time directly on graph nodes and edges. This enables one-stroke gestures to answer questions such as “Are these nodes ever connected?” or “When is this node present in the network?” Analysts can directly query changing graph elements and investigate those changes by navigating time, while focusing on the element.

## 1 INTRODUCTION

In dynamic graphs, low-level changes, such as element addition and removal directly contribute to higher level changes, therefore, it is important to understand how elements change at a low level [2]. We designed *Glidgets*, an interface for exploring low-level changes in dynamic graphs using interactive glyph representations of element changes. Specifically, the changes visualized by our technique are: element presence and absence, as well as node degree. These topological changes are visualized in time line glyphs, acting as time sliders embedded in the graph element. This allows a user to first query graph element changes and then investigate those changes by navigating time, while remaining focused on the element of interest. For interacting with graph elements, Glidgets employs pen-based gestures for flexible and precise interaction [3]. For instance, using Glidgets to answer “Are two nodes ever connected over time?”, one can first draw a line connecting the nodes to reveal an edge glyph that shows if and when the nodes were connected. Furthermore, navigating to any time point can be done by dragging along the glyph.

One of the simplest techniques to show how a graph changes over time is to display images of the graph at each time slice (small multiples [5]). Alternatively, animation illustrates temporal changes through motion and is typically paired with a time slider for controlling the animation speed. These techniques can be cumbersome and inaccurate for observing and tracing evolving, local graph structures mainly due to simultaneous topological changes [1]. Sometimes, these techniques are paired with difference highlights to visualize an aggregation of the temporal changes that occurred between two time points (e.g., [4]).

Temporal navigation controls such as the time slider are separated from the graph, requiring the user to shift their focus between the changing element of interest and navigating time. Examining all low-level changes occurring to an individual element is difficult

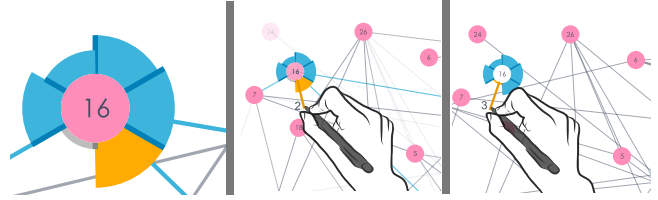


Figure 1: A node glyph is revealed by tapping any node (left). Dragging along the glyph moves in time and other elements are faded in and out (middle). When a selected node disappears, its glyph remains visible until it is de-selected (right).

to track manually. Moreover, visual effects highlighting changes only temporarily draw attention to a change and are separated from navigation controls (e.g., [2, 4]). In Glidgets, temporal changes of an element are visualized in a small time line glyph and navigation can be performed directly on the element in focus, unifying the navigation control and highlighting technique.

## 2 INTERACTIVE CHANGE GLYPHS

Using Glidgets, a user can first query an element’s changes by selecting it to reveal its glyph, and then use a slider embedded in the glyph to navigate time. Here, the glyph guides temporal navigation. The change glyphs visualize three low-level changes: node and edge presence (addition and removal) and node degree. Our glyphs were designed to have a minimal, yet useful representation for visualizing temporal changes, and arrange changes using an intuitive, time line visual metaphor. The glyphs represent small time lines and are evenly divided into segments for each time slice. Element presence is encoded with colour; grey for absence and blue for presence. To integrate the glyphs with the graph elements, they are drawn as highlights on existing elements.

### 2.1 Node Glyphs

A node’s glyph is revealed by tapping it. A clock metaphor is used to arrange the time slices on the glyph. The time line begins at the top of node and is wrapped around the node (Figure 1(left)). The glyph is divided into equal-sized annulus segments, coloured according to the node’s presence (blue for presence, grey for absence). One segment is assigned for each time point, and the start points of time intervals are marked by a darker coloured line at the start of glyph segments. The varying heights of the glyph segments encode the relative node degree. The height is proportional to the ratio of current degree at a segment and max degree of node over time. The maximum and minimum heights are globally normalized across all nodes. The node’s time slider is activated by dragging around the glyph and an elastic tether is drawn, connecting the pen tip to the glyph (Figure 1(middle, right)). At the same time, the rest of graph is updated to reflect the time change. Corresponding to the clock metaphor, the temporal direction is designed to be intuitive: rotate clockwise to go forward in time. To improve visual tracking during navigation, the node’s incident edges are highlighted in blue, illustrating the degree changes more prominently. The current time slice segment is highlighted in orange on the glyph and the

\*e-mail:brittany.kondo@uoit.ca

†e-mail:hrim.mehta@uoit.ca

‡e-mail:christopher.collins@uoit.ca

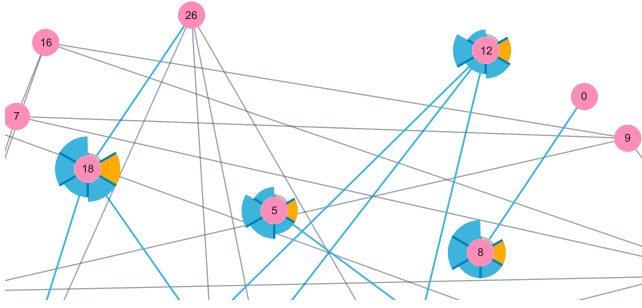


Figure 2: The aggregated glyphs for four selected nodes. The glyphs reveal that the nodes are present together in all time slices, except for the first one. The orange segment indicates the current time.

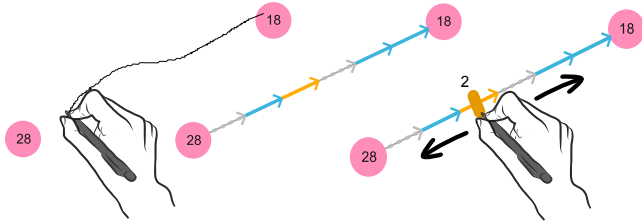


Figure 3: Sketching a line between two nodes (left) reveals the glyph of an edge (middle). Dragging along the glyph moves in time (right).

time slice number is displayed as a label next to the pen tip (Figure 1(middle)).

## 2.2 Aggregated Glyphs

To support exploring and comparing temporal changes of element groups, when multiple elements are selected the change glyphs are automatically aggregated to show when multiple elements appear together over time, or, the *intersection of presence* (Figure 2). Therefore, a blue segment means all selected elements are present at that time slice, whereas a grey segment means at least one element has disappeared. Otherwise, the visual encoding for both nodes and edges is the same as the single glyph, and embedded time sliders can be activated on any of the glyphs.

## 2.3 Edge Glyphs

To select an edge and reveal its glyph, a line can be sketched between any two nodes (Figure 3(left)). The glyph is drawn as a temporary highlight on top of the edge (Figure 3(middle)). A linear time line metaphor is used to arrange the time slices (i.e., time begins at left-most node). The segments of the glyph are equally divided into time steps. The beginning of each time step is marked with an arrow pointing forward in time, indicating the temporal direction. The glyph segments are coloured according to the edge’s presence. When the edge is not present during a time step, a grey dotted line is drawn allowing the user to see the underlying edge. The edge time slider is activated by dragging anywhere along the glyph. During dragging, a short sliding bar appears perpendicular to the glyph similar to a traditional time slider (Figure 3(right)).

## 2.4 Temporal Overview

To support global analysis of changes over time, we created a *global view*. This view uses *global change glyphs* to show presence changes of all element, across all time steps. The global change glyph design is similar to the regular change glyphs, except only presence of an element is represented. Users can select elements of interest in the global view, however the global change glyphs are

independent: presence is not aggregated. By using a slider on a selected element, the user can transition directly to any time point in the regular graph view.

## 3 EVALUATION

We performed a comparative, exploratory evaluation between Glidgets and the regular time slider. We designed a set of tasks requiring participants to locate a moment in time when the low-level changes of an element, or set of elements, had a certain behaviour, such as: “Find the first moment when a node disappears”. To create tasks, we used an altered version of the Van De Bunt network [6], consisting of 31 nodes, 111 edges and 6 time slices. The graph layout was force-directed and node positions were fixed.

Using a within-subjects design, we invited eight participants to complete fourteen different tasks using Glidgets and the regular time slider. In the Glidgets condition, participants could use all interactive glyphs, the global view and regular time slider. For each task we measured completion time and error rate, and asked participants to subjectively rate each technique. The study lasted approximately one hour. While Glidgets did not significantly outperform the regular time slider, subjective feedback strongly favoured the glyphs for completing tasks. In particular, there was a high preference for the node glyphs.

When using Glidgets to solve tasks, many participants combined multiple techniques (e.g., alternating between the node and regular time sliders). These combinations varied for nearly every participant, in all the tasks. Unexpectedly, some participants used the edge glyph and slider to solve tasks regarding node presence changes, while other participants used only the global view to solve presence tasks. While this finding demonstrated the diverse ways Glidgets techniques can be combined to explore element changes, the use of multiple techniques for solving a single task negatively impacted completion times and was overwhelming for some participants.

## 4 CONCLUSION

We presented Glidgets, a glyph-based interface for querying element changes and navigating time, while focusing on an element of interest. We performed an exploratory evaluation comparing the performance of Glidgets and a regular time slider. While our quantitative measures did not favour Glidgets, participants preferred to use the glyphs for completing the tasks. As future work, we plan to conduct a follow-up evaluation comparing sub-sets of Glidgets techniques to the regular time slider for certain types of tasks (e.g., use only the edge slider and glyph to solve tasks regarding edge presence). Lastly, alternative designs for the glyphs should be explored to improve scalability to longer time lines. For example, the node glyph could be layered to aggregate and show different levels of time.

## REFERENCES

- [1] D. Archambault, H. C. Purchase, and B. Pinaud. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):539–552, 2011.
- [2] B. Bach, E. Pietriga, and J. Fekete. Graphdiaries: Animated transitions and temporal navigation for dynamic networks. *IEEE Transactions on Visualization and Computer Graphics*, 20(5):740–754, 2013.
- [3] G. Goth. Brave NUI world. *Communications of the ACM*, 54(12):117–119, 2011.
- [4] S. Rufiange and M. J. McGuffin. Diffani: Visualizing dynamic graphs with a hybrid of difference maps and animation. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2556–2565, 2013.
- [5] E. R. Tufte. *Envisioning Information*. Graphics Press, Cheshire, USA, 1990.
- [6] M. A. Van Duijn, E. P. Zeggelink, M. Huisman, F. N. Stokman, and F. W. Wasseur. Evolution of sociology freshmen into a friendship network. *Journal of Mathematical Sociology*, 27(2-3):153–191, 2003.