ABSTRACT

Title of Thesis: DATA-DRIVEN STORYTELLING
IN DYNAMIC GRAPH COMICS
THROUGH HIERARCHICAL CLUSTERING

Abhinav Kannan
Master of Science, 2023

Thesis Directed by: Dr. Niklas Elmqvist
College of Information Studies

In this work, we explore a tool to generate a dynamic graph comic given dense, time-series edge-vertex data. Prior research has demonstrated the effectiveness of node-link diagrams as an expressive medium for storytelling with dynamic graphs, and in this work, we develop an interface that generates a customizable comic strip. We use hierarchical aggregation to cluster and pile graphs based on the number of frames a user may wish to see, with each frame depicting a snapshot in time. We validate the interface for application with real world datasets to understand temporal changes in a graph network, and evaluate the interface against an expert audience. Finally, we propose a path forward for improvement of dynamic graph comics as a storytelling medium.

DATA-DRIVEN STORYTELLING IN DYNAMIC GRAPH COMICS THROUGH HIERARCHICAL CLUSTERING

by

Abhinav Kannan

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2023

Advisory Committee:
Dr. Niklas Elmqvist, Chair/Advisor
Dr. Tamara Clegg
Dr. Joel Chan

Dedication

To all the teachers in my life.

Acknowledgments

I owe my gratitude to my parents, without whose love and support I wouldn't have come this far in life. Amma and Appa, thank you for everything.

I would like to thank my elder brother, Sanjeev, who has saved me from financial crisis on multiple occasions. Time and again, he has believed in my goal to study well, even if that means that he may sink financially to keep me afloat.

I owe my endless gratitude to my guide and advisor, Dr. Niklas Elmqvist, for entrusting me with this deeply exciting work. His time, insights and feedback have made me a better person. His patience in putting up with me is a trait that will inspire me for decades to come.

To all my teachers and Gurus, who have taught me skills to get to where I am today.

I would like to thank my extended family, and relatives for all of the support they've given me. This list of people is endless, and could constitute a master's thesis in itself.

All the friends I have made along the way – at every class and school I have attended, and beyond – have been an endless source of support in this journey. Even if I haven't been in touch with everyone, my sincere thanks and best wishes certainly go out to them.

I would like to thank Dr. Tamara Clegg and Dr. Joel Chan, for taking the time to serve as the committee members for this thesis. Their guidance and encouragement was inspirational in the improvement this work.

Apart from the authors at the University of Maryland who have contributed to comics and

visualization, my thanks go out to a number of authors in the field who have inspired me to take up this work – Dr. Benjamin Bach, Dr. Nathalie Henry Riche, Dr. Hanspeter Pfister, just to name a few – and their collaborators who have contributed with some fabulous literature in data comics.

Finally, my gratitude to the divine energies of the universe for making this journey happen.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

CSS	Cascading Style Sheets
CVD	Color Vision Deficiency
D3	Data-Driven Documents
HTML	HyperText Markup Language
JS	JavaScript
JSON	JavaScript Object Notation
UML	Unified Modeling Language

Chapter 1: Introduction

In today's digital era, we are seeing an increasing role of data and devices that are able to capture information to the tenth of a second. There are more cameras, more smartphones, and a number of sensors and methods to record information and track movements in time – ranging from small and highly portable electronics to elaborate lab apparatuses. There is an abundance of time-series data coming in from a range of such sources. Tools developed by creators across academia and industry have been able to help study and analyze temporal patterns, and journalistic media sources have played an important role in attempting to tell stories with these complex time-series datasets to the common public.

One such category of data has been the temporal study of relationships between entities, sometimes known as "network analysis" [3] to help study the evolution and structure of relations and interactions, and particularly between humans, animals and companies to name a few. We have come a long way in terms of technological progress to have devices help us today in capturing such interactions, as opposed to human observation which can often be slower and unreliable. A few notable enabling technologies in this end include wearable sensors and advanced computer vision.

Across various forms of journalism, node-link diagrams have become an effective medium to represent analytical insights from the study of networks [1]. A node-link diagram is a visual

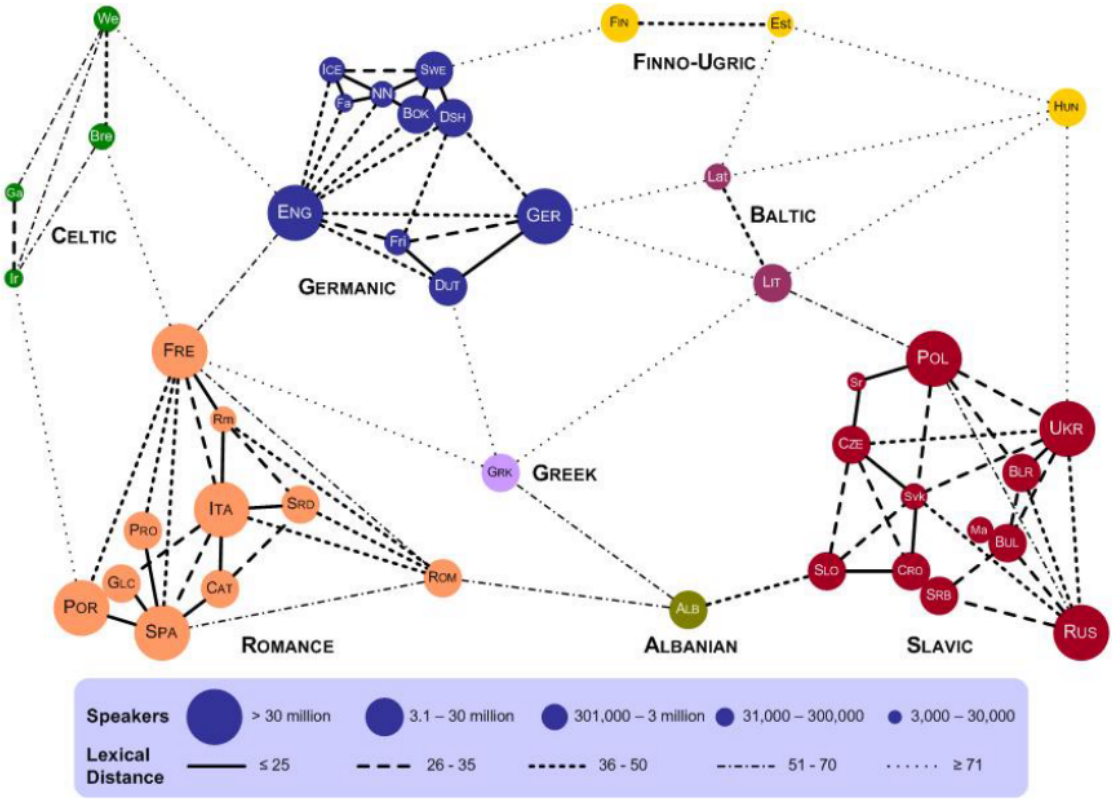


Figure 1.1: Node-link diagram depicting language relationships by Spritzer et al. [1].

representation whose vertices are visual "nodes", connected by lines commonly termed "edges" (or simply "links"). At the same time, there has been a number of constrains imposed by said forms of journalism, along with significant drawbacks in creating these diagrams. Print media, in particular, presents space constrains across formats such as magazines and newspapers. Creating node-link diagrams across time for these formats is a challenge that is not only time and labor intensive, but also often involves either programming or design skills (or both) which is not available commonly as a skill. The problem is exacerbated further when the data presented is dense across a large period of time, and there is a lack of means to intelligently pick the most suitable timestamps (or combined ranges of them) from the dataset.

Given that there is a growing need to convey such data quickly and meaningfully, we

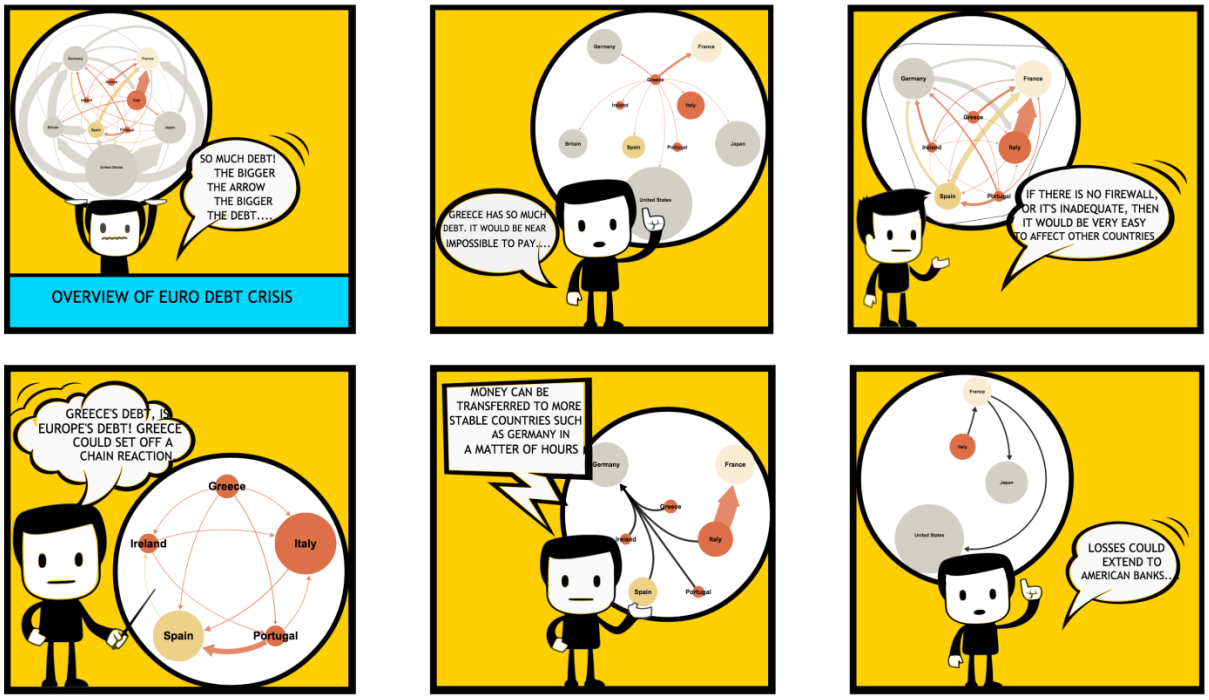


Figure 1.2: Dynamic Graph Comic on Euro Debt Crisis (Zhao et al.) [2].

explore the representation of node-link diagrams as dynamic graph networks, which is a form of visualization whose relations between objects change over time [4]. Studies have shown that depicting dynamic graphs as node-link diagrams can be familiar to a wide audience with minimal training, and therefore a preferable medium for storytelling [5]. However, a tool that facilitates the automatic creation of node-link diagrams remains an under-explored area in storytelling with visualization.

Based on this gap, we propose an interface that combines comic-style storytelling and hierarchical aggregation, that supports the creation and export of dynamic graphs as a series of node-link diagrams.

1.1 Purpose of Study

The purpose of this study was to develop an application prototype to help observe changes in complex, dense dynamic graph networks to better understand large time-series edge-vertex data.

1.2 Research Questions

While ideating the goals of our interface, we came up with the following research questions:

- RQ1: What are the interactions and visual cues that may potentially be useful in storytelling with dynamic graph networks?
- RQ2: How can we effectively communicate change in patterns between panels?
- RQ3: How can we automatically generate a comic strip based on the choice of frames specified by a user?

Keeping the above RQs in mind, we decided to design and develop a web application for dynamic graph visualization.

Chapter 2: Related Work

Our work on this application lies at the intersection of visualizing data-driven comics, dynamic graphs, followed by piling and aggregation. This chapter addresses the related work in these areas. In the first section, we review research studies linked with data-driven storytelling and its effectiveness in communication and interpretation. In the second section, we present a discussion on dynamic graph networks and node-link diagrams, particularly on tools that facilitate graph generation. Lastly, we dedicate a section for algorithms that facilitate piling, clustering and aggregation as method for summarization and selection in visualization.

2.1 Data-Driven Storytelling

There has been a growing interest in incorporating narrative storytelling in order to make statistical representations more engaging, and concepts such as Gestalt grouping [6] and vectorial reference have become powerful techniques in interface design as well as numeric visualization [7]. McCloud's taxonomy on narration with visual elements has also been of interest in making visualizations more story-driven [8, 9].

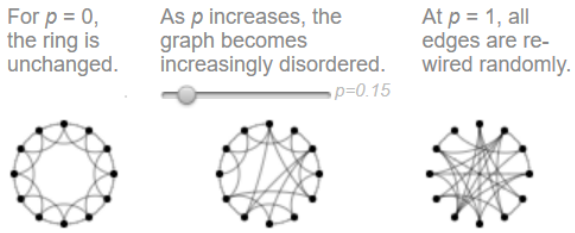
Storytelling in visualizations has come a long way since being highlighted as an engaging area of exploration by Gershon and Page [10], while Segel and Heer in 2010 have described comics as one among seven genres of narrative visualization [7]. A number of tools for comic

creation for visualization have emerged in recent years, with the work of Zhao, Marr and Elmvist inspiring an entire area of visualization research commonly addressed as data-driven comics (or simply, "data comics") [2,11]. Bach et al. have further established this field by exploring trends in narration that prove helpful for visualization [12], and provided design considerations for creators to produce visualizations with these comics [5, 13, 14].

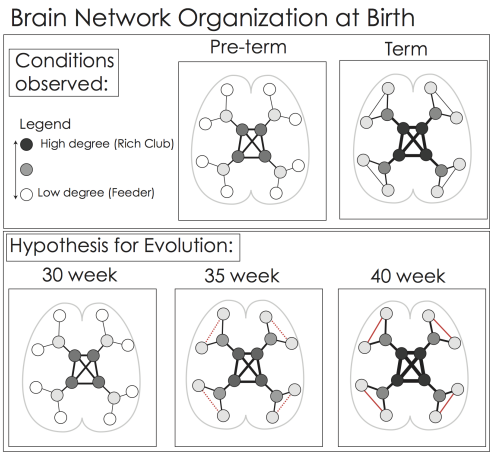
Given the limitations present in the area of creating time-series graph visualizations, Segel and Heer observe that the creation of such charts often requires "a diverse set of skills", while also suggesting that tools can be powerful for analysis but "do little to aid narrative communication" [7]. When it comes to dynamic graphs, not only can their diagrams be hard to create, but their connection topologies in a real world setting can range anywhere between minimal to extensively dense, or even between regular to completely random connections [15]. To illustrate this scenario further, Bret Victor has been influential in presenting "Scientific Communication As Sequential Art" – a show and tell method to describe and depict scientific literature in comic-style [16] 2.1a. Several studies and tools have attempted to use such style address the temporal aspect of data-driven stories for dynamic graphs [5, 13, 17, 18] 2.1. However, many of these studies are yet to address the visualization of dense temporal edge-vertex datasets that take advantage of the storytelling medium.

2.2 Dynamic Graph & Node-Link Representations

A number of graph generating tools that provide fine-grain control to generate node-link diagrams have emerged in both open-source and enterprise formats. A few such tools that offer a high degree of manipulation are Gephi, GraphCoiffure and yED [1, 19, 20]. GraphCoiffure, in



(a) Interactive Graph Comic by Victor [16]



(b) Brain Network Evolution by Bach et al. [5]

Figure 2.1: Graph Comic Examples.

particular, uses physics-based concepts for rendering and layout manipulation, taking inspiration from magnetism and rotational dynamics. In addition, a user can provide a CSS-stylesheet integration for finer control over visual elements. However, the tool lacks a comprehensive support mechanism for time-series datasets, given that only one graph can be viewed at a time. On the other hand, Gephi incorporates a time-series slider, allowing a user to view the evolution of a graph through dragging. Both Gephi and GraphCoiffure have a wide set of applications and provide interactions for filtering, ranking and grouping, along with customization for line strokes and color. yED provides pre-defined shapes to create UML and network diagrams, using drag-and-drop interactions. A powerful benefit of the yED application is its seamless integration with Microsoft Word, along with ability to modify shapes and export in a variety of file formats.

While these tools offer a high-level of interactivity, functionality and manipulation, only Gephi provides a time-series slider, and none of them support the ability to view more than one graph at a time. While generating multiple dynamic graphs with these tools is certainly achievable, doing so can be time-consuming and labor-intensive.

2.3 Piling, Clustering and Aggregation of Visual Data

A number of attempts have been made to drive storytelling using clustering and aggregation algorithms. One such attempt is the work by Zhao et al. on Sketcholution, an interface that visualizes frames of sketch histories [21]. This interface gives the user the option to display the number of sketch frames. Our work is similar to Sketcholution in terms of how the clustering tree is formed (which subsequently helps decide the number of frames), however, our work combines and aggregates frames differently by performing a Cartesian union of nodes and links in the network.

Bach et al. explore temporal changes in brain connectivity networks by using hierarchical clustering to pile snapshots over a large period of time [22]. This interface provides interaction techniques with piling as a design metaphor, in addition to creating a summarized aggregated frame (or "pile") as a cover matrix for a bundle of frames. This system also proves particularly useful in providing visual cues to identify fuzzy brain states, where each state is a dynamic network between different regions of the brain represented as an adjacency matrix. Our system employs a similar approach in piling to visualize state transitions, although in our case, the time-series data being visualized is an edge-vertex dataset as opposed to adjacency matrix. Our application also provides a text summary as a caption for each frame, describing the changes between time states.

A tool proposed by Burch and Weiskopf [23] comes close to our proposed work, and is notable in using small multiples to visualize dense dynamic graphs. This work uses a flipbook interaction technique to visualize dense temporal dynamic graph data. While this work uses edge-splatted radial layouts to reduce visual clutter for node-link diagrams along with aggregation, our

work emphasizes the need for data-driven storytelling with captions, node-labels and comic-style visual elements for enhanced engagement.

2.4 Summary

We have explored some of the relevant literature with regards to our work and identified a gap to justify our use case. Taking inspiration from a statement by Bach et al., who suggest that comics as a storytelling medium can help "explain complex processes by splitting them into less complex units" [12], we not only attempt to take advantage of comics as a storytelling medium, but develop an interface that would split complex graphs into smaller units in time. In the chapters to follow, we outline the ideas of our system and evaluate it against real world usage scenarios.

Chapter 3: Dynamic Graph Comics: System

3.1 Interface Design Goals

In designing our interface, we outline the following design requirements as below:

- **Graph layout:** Graphs should be displayed large enough such that its nodes, links and labels are easily visible in a panel. We depict a member of the network as a node, and a relationship between two members as a link. A label for a node is an identifier, ideally a name. We follow this criteria to connect and retain users' familiarity with a typical graph network.

- **Panel size and rendering:** The panels will need to adapt to the availability display space, and render as close as possible to a square. Connecting with existing work on comics and comic-based visualization, panels should be arranged into strips and separated by gutters. Each panel should follow a chronological order of layout by rendering snapshots in time from left to right.

- **Slider Control:** The user has the choice and control of deciding the number of frames to display, with the content of each frame being decided through hierarchical aggregation. The number of ticks displayed on the slider would be equal to the number of time instances in the dataset, with the default value being set to 10.

- **Visual cues:** We define and render the nodes and links in the graph as SVG elements. A node is represented as a colored circle, while a link is a stroke. The color of each element is chosen based on the following criteria:
  - Dark gray: If the element has an established presence in the graph, either at the start or from its preceding snapshot in time;
  - Green: If the element has just been added to the network compared to the frame preceding it; or
  - Faded gray: We show the element as faded gray if it has been removed from the network in the preceding snapshot.
- **Captions and Editable Text:** Our prototype generates captions that describe the changes in each frame. Each caption is an editable line of text describing the number of nodes and links added or removed, compared to the frame preceding it. This caption can be edited if the user wishes to do so. The node labels in each graph can also be edited.

.

Keeping the above goals in mind, we created our first prototype as shown in Fig. 3.1.

3.2 System Implementation

Our implementation begins with accepting an edge-vertex dataset selection from the user for visualization. In an ideal scenario, a typical user would provide a dataset to upload onto the application; however for the sake of simplicity and to demonstrate visual interpretation, we focus on two real-life use-cases with edge-vertex temporal datasets.

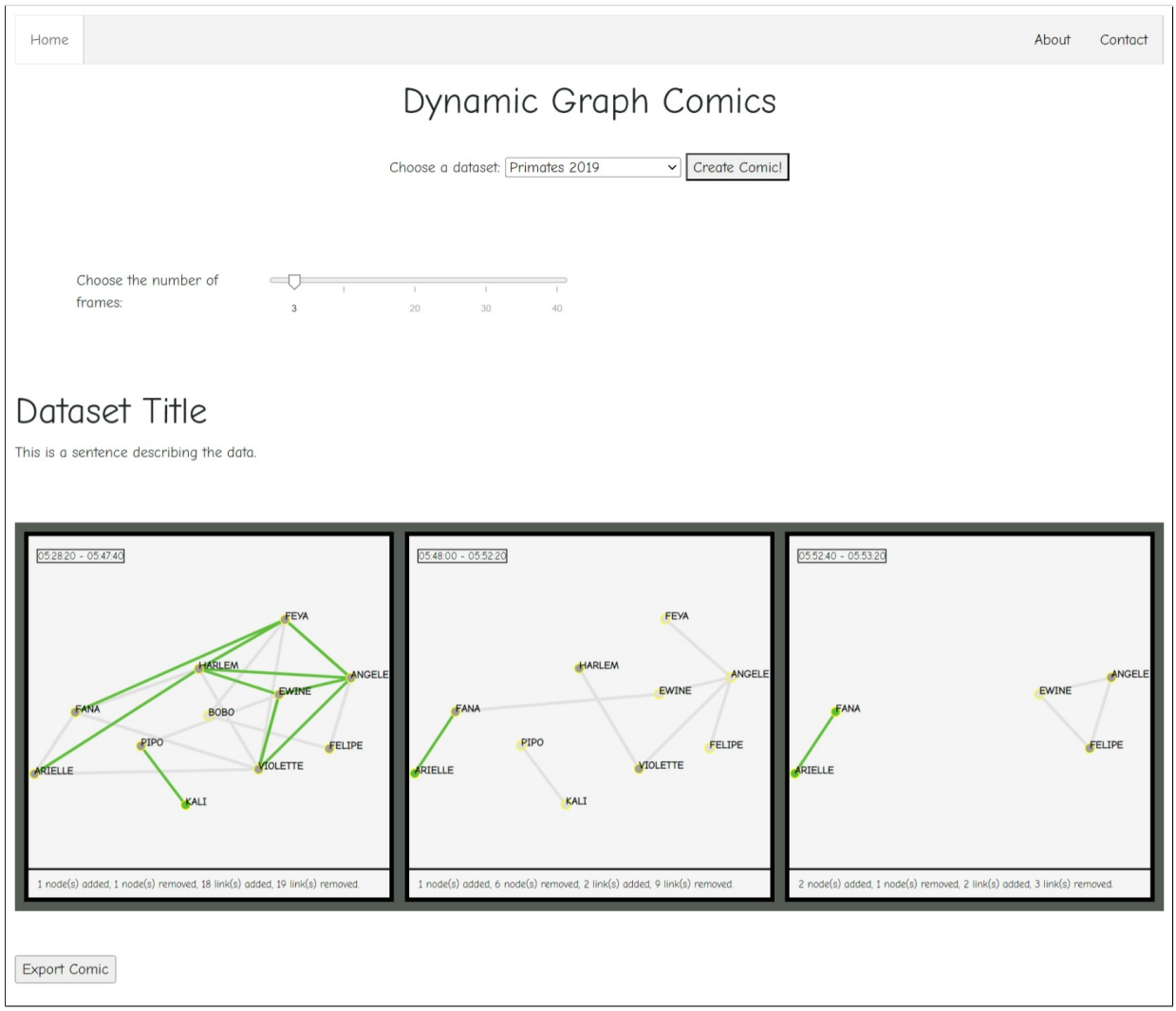


Figure 3.1: Dynamic Graph Comics App: A First Look.

3.2.1 Object Oriented Design

We take an Object Oriented Programming approach in engineering our dynamic graph network. This begins by defining a JavaScript class that would encapsulate the entire dynamic graph for all instances of time. We incorporate this method since each graph is a dynamic instance having similar properties – consisting of nodes, links and positional values – therefore, a JavaScript class with encapsulation and maintainability as its key advantages proves ideal to model our dynamic system.

3.2.2 Snapshot Structure

To create a new snapshot, we declare an instance of this class as a new dynamic graph object with time as an argument. Therefore, each snapshot for a particular time instance consists of the corresponding vertices and edges for the graph network during that time. This is then stored as a JSON object structure with primarily two data properties, one for storing node elements, and the other for links.

Each node object consists primarily of an identifier and name, with additional scope for adding properties that would support grouping, ranking and filtering. Each link object consists primarily of a node source, node target and time duration indicating presence in the graph network.

3.2.3 Hierarchical Aggregation

We follow a sequential method of hierarchical aggregation which is a simple and effective way to summarize a series of snapshots while preserving its chronological order and its main information. Summarizing a pair of frames implies creating one aggregated frame, and therefore

reduces the number of frames while keeping the main information and the chronological order.

Step 1. We create an array of all original snapshots as the input sequence.

Step 2. We compute the distance D between each pair of adjacent frames in the sequence based on a distance function that computes the change from $F(t)$ to $F(t+1)$.

Step 3. We find the pair of adjacent frames with the smallest distance and merge them into a single frame. The merged frame contains the information from both of the frames.

Step 4. We repeat steps 2 and 3 until all time instances are merged into one frame.

In our case, the distance function is the sum of all changes between the two time instances, in terms of node-link additions and removals to the network states. We formally define a high-level version of our hierarchical aggregation procedure under Algorithm 1.

Algorithm 1 Hierarchical Aggregation

Require: S, the set of all snapshots
Ensure: T, clustering tree formed from snapshot aggregation

```
distance_objects; // Array for storing distance objects
for all 0 ≤ i < S.length do
    child1 ← S[i]
    child2 ← S[i + 1]
    δ ← diff(child1, child2);
    Δ ← {δ, child1, child2};
    distance_objects[i] ← Δ;
end for

sort(distance_objects); // Sort objects in increasing order of δ

for all 0 ≤ j < distance_objects.length do
    child1 ← distance_objects[j].child1
    child2 ← distance_objects[j].child2
    parent ← aggregate(child1, child2)
    T ← formTree(parent, child1, child2)
end for
```

In the following section, we outline the mechanism behind merging two frames to create

an aggregated frame for rendering each graph visualization.

3.2.3.1 Combining Aggregated Frames

When creating a combined aggregated frame, we perform a pairwise unique addition (or Cartesian union) of nodes and links in both the graphs. In other words, the aggregated frame will contain the summarized data that pertains to both frames across the combined time period. To visualize the aggregation, we denote additions in green, while the removals are reduced in opacity. This visualization rule is applied to both node- and link-sets.

Since each snapshot is essentially a JSON object structure, we update its properties to denote changes from the earlier frame, in terms of addition and removal.

Finally, the aggregated frame is assigned as a parent to both of the frames in the pair. This process is recursively repeated until we have only one node at the highest level, i.e. the root node which would be the summarized parent of all frames in the entire time duration.

3.2.4 Technical Architecture

In this section, we briefly describe the technical stack of the application interface. Our aim from the start was to build a browser-based application that would ideally be accessible from any device. Therefore, we've prototyped a JavaScript web app that can presently run on a desktop environment or a mobile tablet.

With the key element of our application being visualization of the dynamic graphs, we picked D3 (Data-Driven Document) as our JavaScript library for rendering node-link diagrams.

The back-end environment is a Node JS implementation (Node 16.x) served up with Express

as our development server. Given that modularity is an important component of the app, running a Node-Express app on a browser with multiple JS files bundled requires a transpilation module. To perform this bundling, we picked Parcel as our bundling and transpilation library given its lightweight size and optimal performance. We use Git and GitHub as our tools for version control, with the latter hosting the server-side code and configuration dependencies.

Finally, we publicly deploy our app with Vercel as our deployment server.

Chapter 4: Usage Scenarios, Analysis & Validation

We take this section to validate our proof of concept prototype with one two real world datasets. The first dataset is from a Primate Center in France, and the second is from a digital contact tracing method at a conference. Following this validation, we formally evaluate the system with an expert review and analyze our findings.

4.1 Analysis from Primate Interactions: 2019

The social interactions between animals present an interesting case to be analyzed. We next explore and visualize node-link diagrams to simulate interactions between primates. Such analysis is of avid interest to primatologists and wildlife enthusiasts, and also useful for those tracking their health and emotional well-being.

We consider a dataset of 20 Guinea baboons residing in a Primate Center in France. This was performed as part of a study by Gelardi et al., recorded from June 13 – July 10, 2019 [3]. This is a rich dataset with a temporal resolution of 20 seconds, along with additional parameters indicating behavioral state. However, in our study, we only consider the first 17 minutes of this data for visualization, since (a) our prototype is currently not being evaluated for scalability since it is still proof-of-concept; and (b) this range of data itself presents for an interesting analysis, since we can obtain a sneak-peek of these primates' behavior in the early minutes of the morning.

To begin with, we obtained the data from `sociopatterns.org`. Since the dataset was a.txt file with tab-separated values, some additional processing was needed to be performed to convert it to JSON format. The seconds recorded by the sensors needed to be converted into clock-friendly timestamps, following which we ran a JavaScript snippet to finally obtain our JSON file with standard edge-vertex time-series data. The dataset has a temporal resolution of 20 seconds, and each tmestamp in the dataset consists of a set of members denoted as nodes, and relations between them denoted as links or edges in the network.

4.1.1 First Glance Observations

On taking a first glance, one can quickly observe some interesting patterns between the members' interactions.

Obs. 1. At 05.44 am, there is a sudden spike in activity. One may deduce here that the baboons have come together for an eventful gathering (Fig. 4.1). A series of events follow with additions and removals to the network.

Figure 4.1: Primates 2019: Spike in activity at 05.44 am.

Obs. 2. There is a sudden decline in interaction at around 05.48 am, where only 2 nodes remain (Fig. 4.2). However, this is again followed by a period of well-connected interactions for

the next 2 minutes.

Obs. 3. The baboons tend to be connected, and it can also be deduced from this timeline that more interactions are established when there are more members.

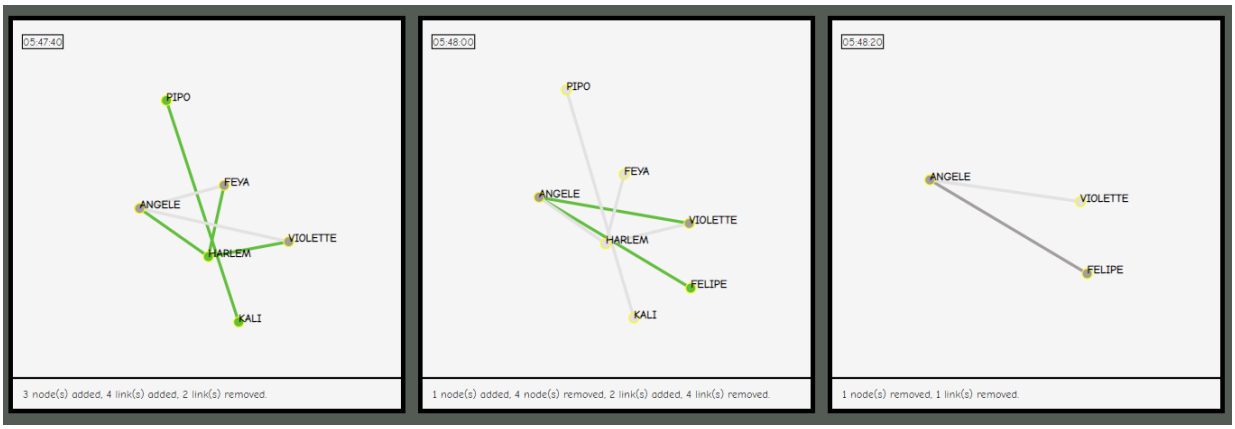


Figure 4.2: Primates 2019: Decline in activity at 05.48 am.

4.1.1.1 Validating First Glance Observations

For validation, we prepare three line charts describing the sequential activity in the network.

Fig. 4.3 describes the number of nodes and links present in the graph at a given time. Here, the solid blue curve describes the nodes; the solid green curve describes the links; and the solid black line describes the sum of the nodes and links, which is the output of our distance function during aggregation.

Figures 4.4 and 4.5 describe the number of changes in the network per unit time; the blue, green and black curves similarly correspond to nodes, links, and the sum of node-link changes, respectively. The solid lines in Fig. 4.4 show the node and link additions to the network, while the dashed lines in Fig. 4.5 show the node-link removals to the network. For better analysis, both figures Fig. 4.4 and Fig. 4.5 share the same black curve indicating the total number of changes.

We can validate Obs. 1 against Fig. 4.3, which shows the number of nodes and links present across time. Fig. 4.4 shows the number of changes in the graph across time, and also provides validation for Obs. 1.

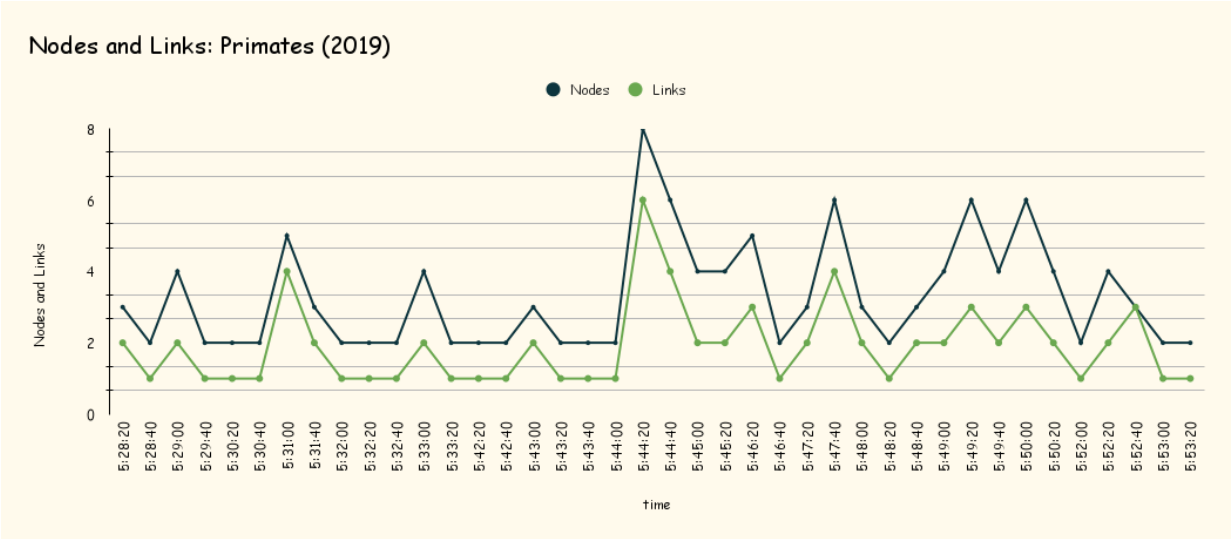


Figure 4.3: Primates 2019: Line graph showing the number of nodes and links in the network, across time.

Obs. 2 can be validated at 05.48 am in Fig. 4.5 and Fig. 4.3, to observe a sharp decline in members and interactions present at the time.

Obs. 3 is a trend that Fig. 4.3 validates easily, directly correlating the trends between number of nodes and relationships present across time.

On reviewing the same dataset with line charts, which is a common representation for visualizing dense, time-series data, we obtain several numerical insights; however, there are a number of shortcomings with doing so. Firstly, to obtain the same insights as above, we are having to generate at least 2 (if not 3) more charts (Figs .4.3, 4.4, 4.5). Secondly, we are unable to track which specific members have joined, and, which relations have been modified. Lastly, the line charts, while mathematically informative, do not come close to simulating a real-world

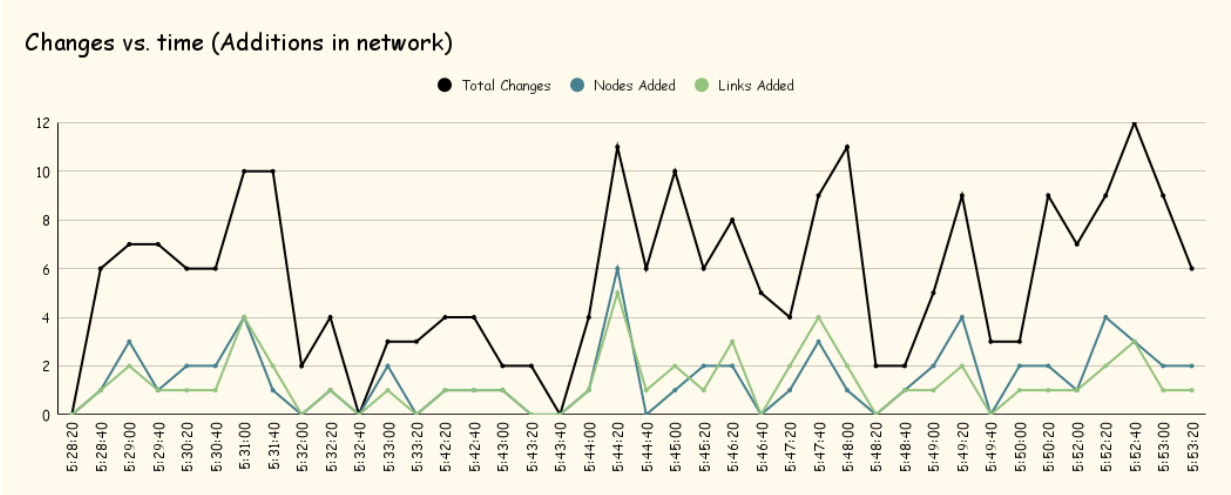


Figure 4.4: Primates 2019: Line graph showing the number of nodes and links changed versus time, with network additions.

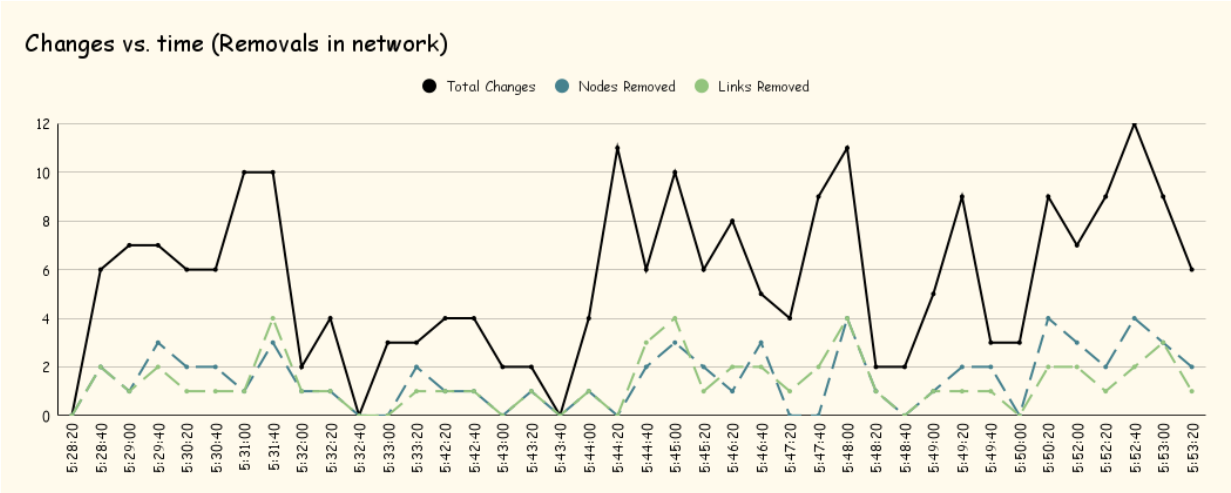


Figure 4.5: Primates 2019: Line graph showing the number of nodes and links changed versus time, with network removals.

network akin to the dynamic graphs.

In summary, while there are certain statistical benefits to using line-charts, we lose a number of informative findings that are provided to us at a first glance with our dynamic graph visualization.

4.1.2 Validating Hierarchical Aggregation

We can use the same curves (Figs. 4.3, 4.4, 4.5) for validating hierarchical aggregation in our application. We prove the results of our algorithm using bottom-up and top-down validation.

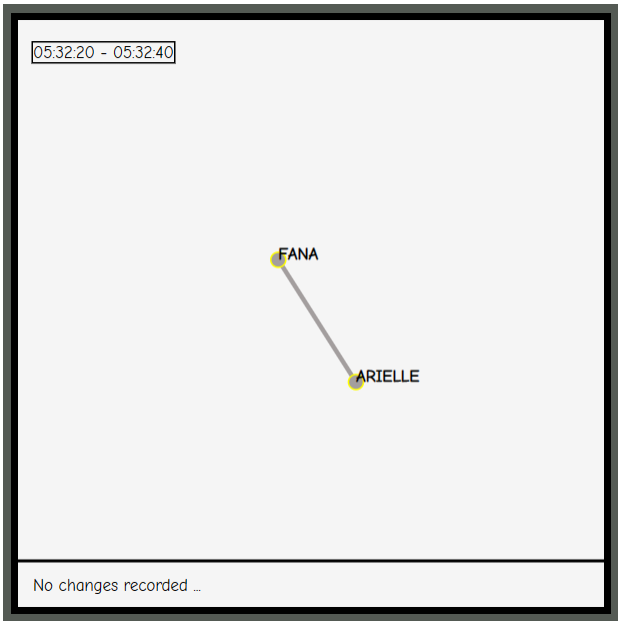


Figure 4.6: Primates 2019: Aggregation of frames between 05:32:20 – 05:32:40.

4.1.2.1 Bottom-up Validation

If we adjust the slider to its maximum value – 40 in this case – we obtain all of the snapshots in the network. If we reduce this value from 40 to 38, the sets of snapshots to be combined by a single step are frames F(05:32:20 – 05:32:40) and F(05:43:20 – 05:43:40) (Figs. 4.6 and 4.7).

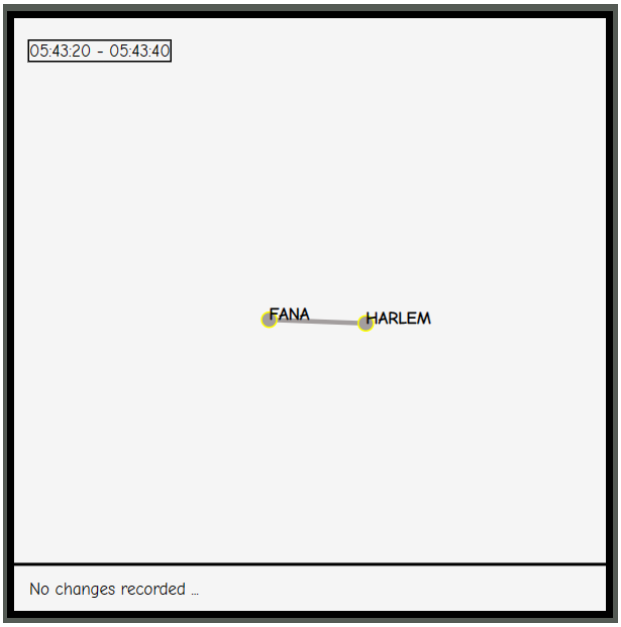


Figure 4.7: Primates 2019: Aggregation of frames between 05:43:20 – 05:43:40.

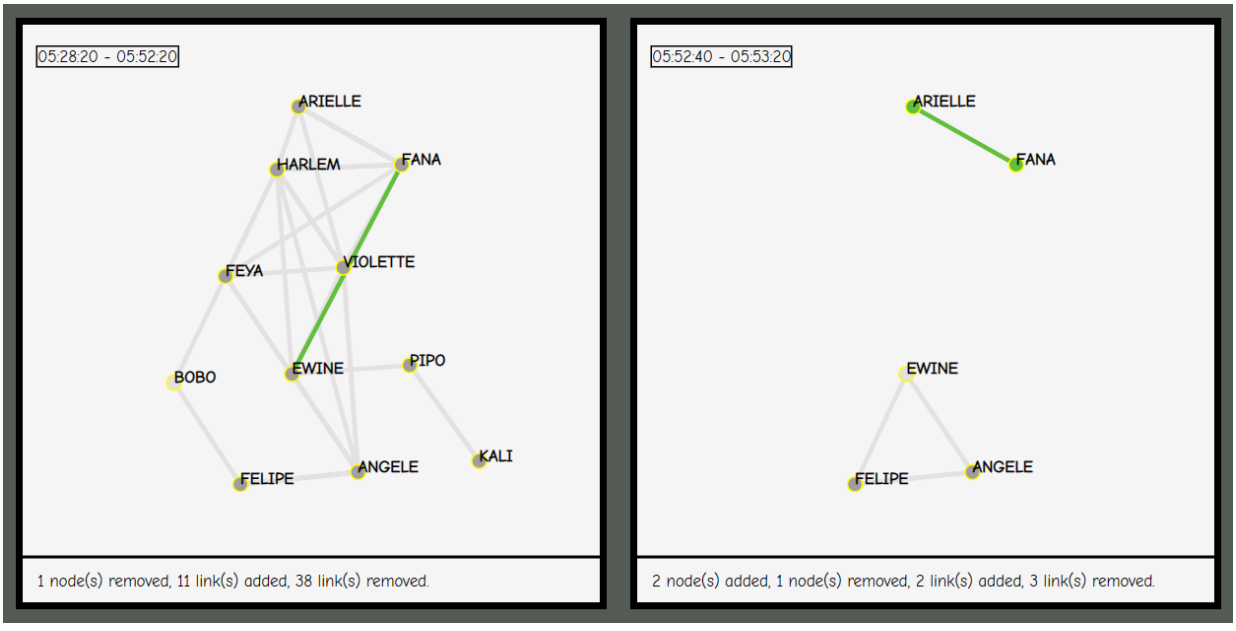


Figure 4.8: Primates 2019: Split in frames between 05:52:20 – 05:52:40.

This is because the distance between them is 0 – meaning that there were no changes between these sets of consecutive timestamps – which also is the lowest value of changes as validated from either Fig. 4.4 or 4.5. Furthermore, the programmatically generated caption in the panels state no changes to be recorded in the events between these timeframes. This ascertains our aggregation algorithm from the bottom-up.

4.1.2.2 Top-down Validaton

If we adjust the slider to a value of 1, we obtain one frame which summarizes all the changes in the network. This frame is selected from the top of the clustering tree. However, if we increase the slider value from 1 to 2, we obtain 2 frames: F(05:28:20 – 05:52:20), and F(05:52:40 – 05:53:20) (Fig. 4.8). Looking at either Fig. 4.4 or 4.5, we see the greatest number of changes occurring in the 20 seconds between 05:52:20 – 05:52:40. This means that frame1 and frame2 were the last to be aggregated given the large distance between them, to therefore ascertain our hierarchical aggregation algorithm as valid from the top-down.

4.2 Analysis: Digital Contact Tracing at SFHH 2009

We now consider a dataset to analyze an application in digital contact tracing, with data from the SFHH conference in Nice, France. This dataset consists of face-to-face interactions between 405 participants during the 2-day event (June 4–5, 2009), and was first released for open use by Génois and Barrat [24]. However, for simplicity, we consider only the first approximately 20 minutes of this data, for validating and illustrating our visualization. Such insights are useful in tracking people movement to combat disease spread, and several studies in recent literature

have explored network analysis with such data, to simulate person-to-person interactive activity and form models for infectious disease spread [25, 26]. The participant names are anonymized, and therefore the node labels displayed in the panels are participant IDs. This dataset has a temporal resolution of 20 seconds.

Similar to before, we obtained a processed version of this dataset from `sociopatterns.org`, which was available in DAT format. We processed the data further to convert the same into JSON format and include clock-friendly timestamps.

4.2.1 First Glance Observations

On first glance of the generated strip, the following observations described below are easily visible among the snapshots in the network:

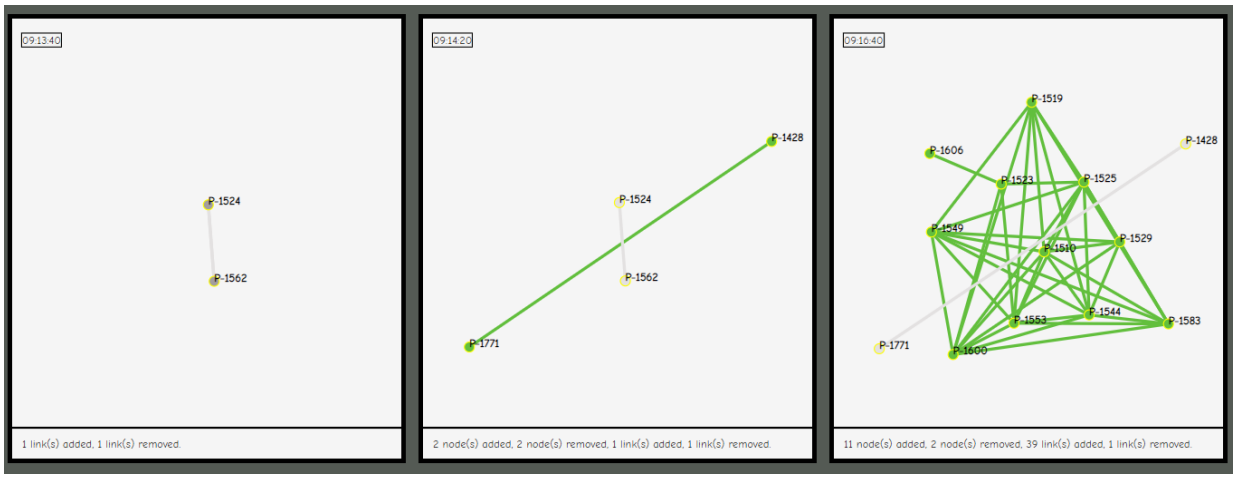


Figure 4.9: SFHH 2009: First major peak of activity at 09:16:40 am.

Obs. 1. Interaction activity in this case is quite slow starting from 09:02 am, with only 0-2 nodes being added for the first few minutes. However, there is a sudden spike at around 09:14 am (Fig. 4.9).

Obs. 2. The member interactions remain high from 09:14 am onwards, until a sudden

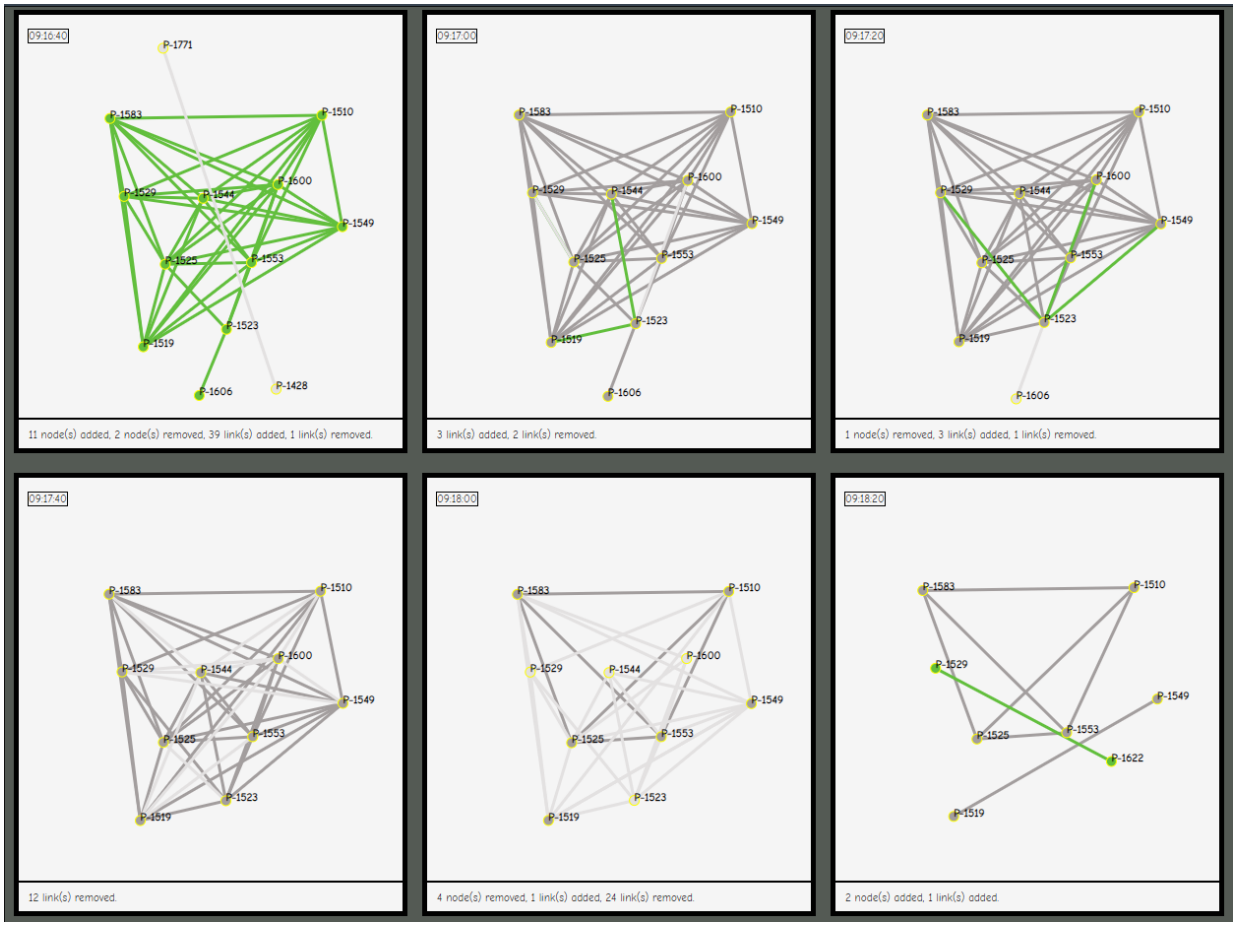


Figure 4.10: SFHH 2009: Activity until decline at 09:18 am.

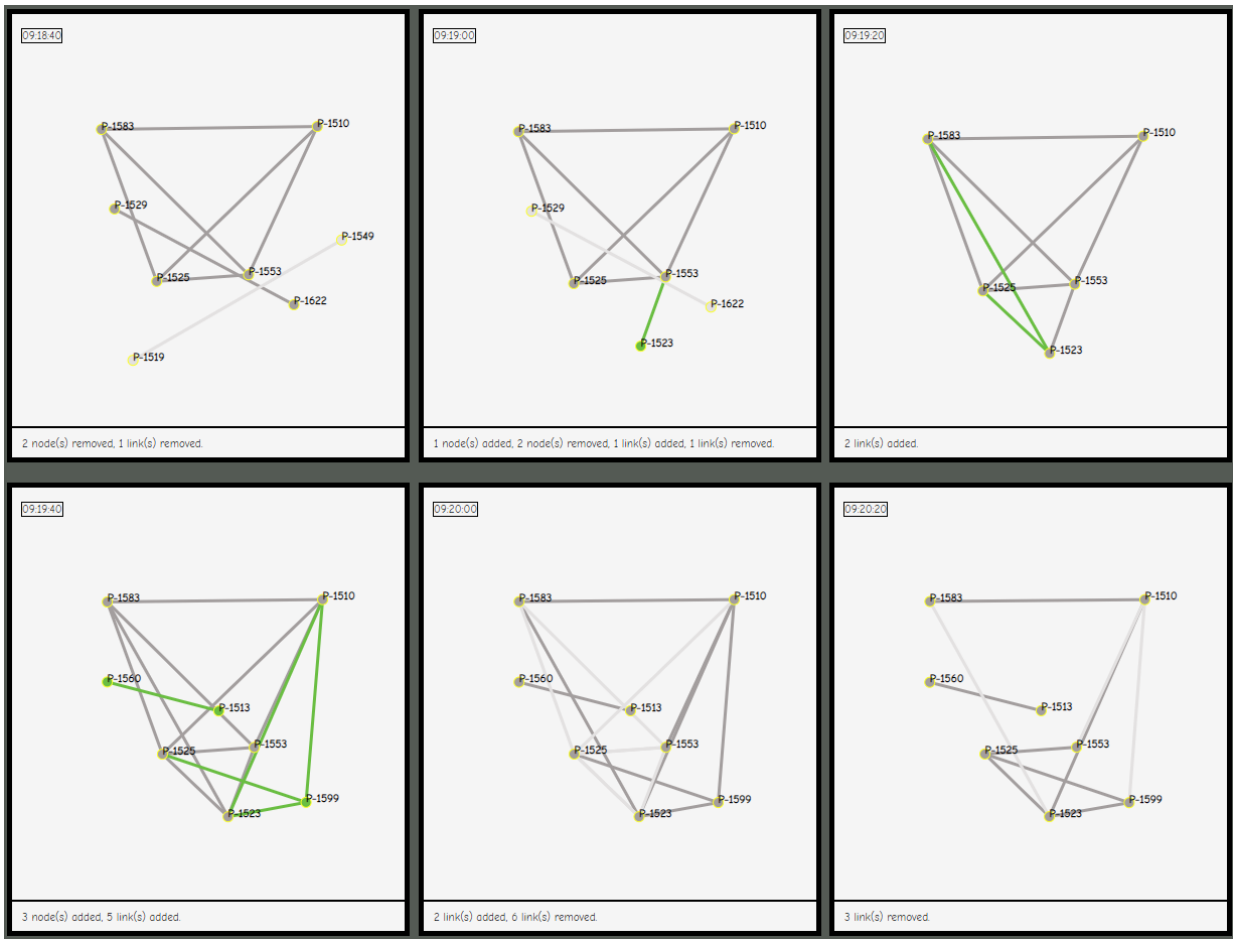


Figure 4.11: SFHH 2009: Spike in activity at 09:19:40 am.

decline at 09:18 am. At this time, a number of links have suddenly faded (Fig. 4.11).

Obs. 3. Activity remains fairly stable after 09:18 am, until another short spike at 09:19:40 am (Fig. 4.10).

The three observations can be validated against line charts depicting activity in the network (Figures 4.12, 4.13, 4.14), describing similarly in the previous section.

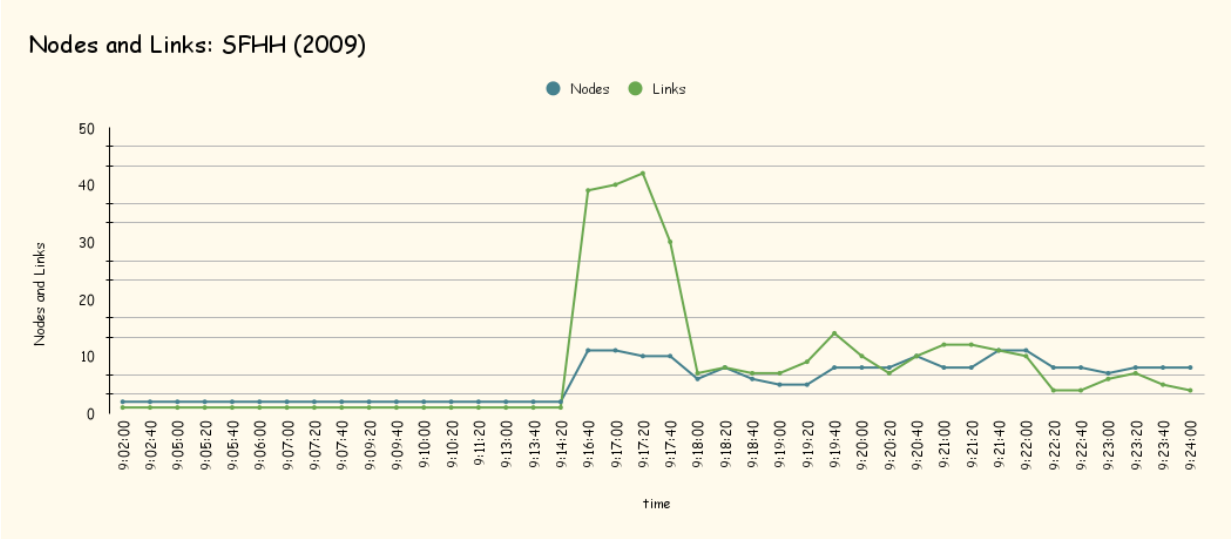


Figure 4.12: SFHH 2009: Line graph showing the number of nodes and links present, versus time.

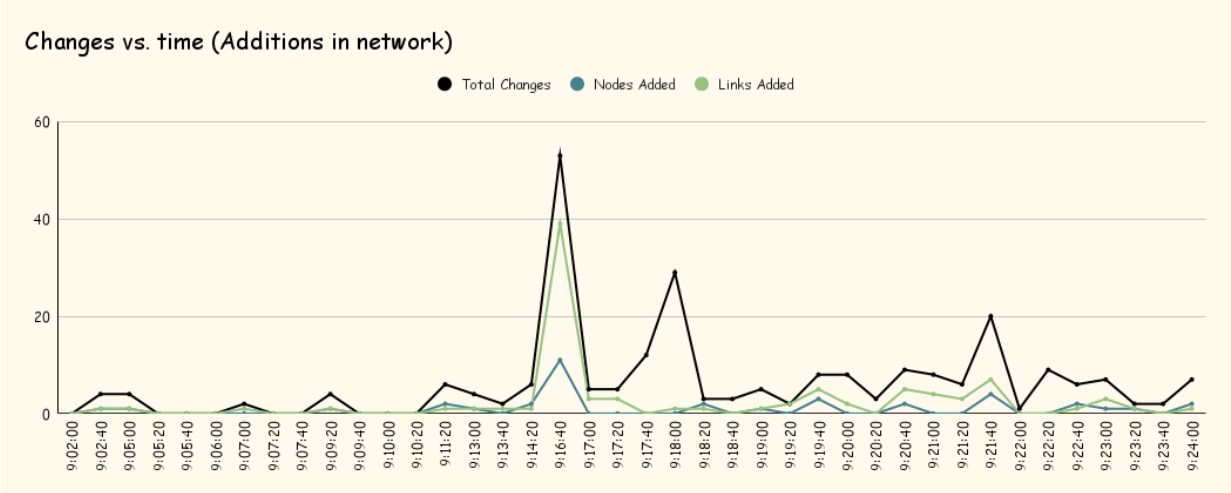


Figure 4.13: SFHH 2009: Line graph showing the change in number of nodes and links versus time, with network additions.

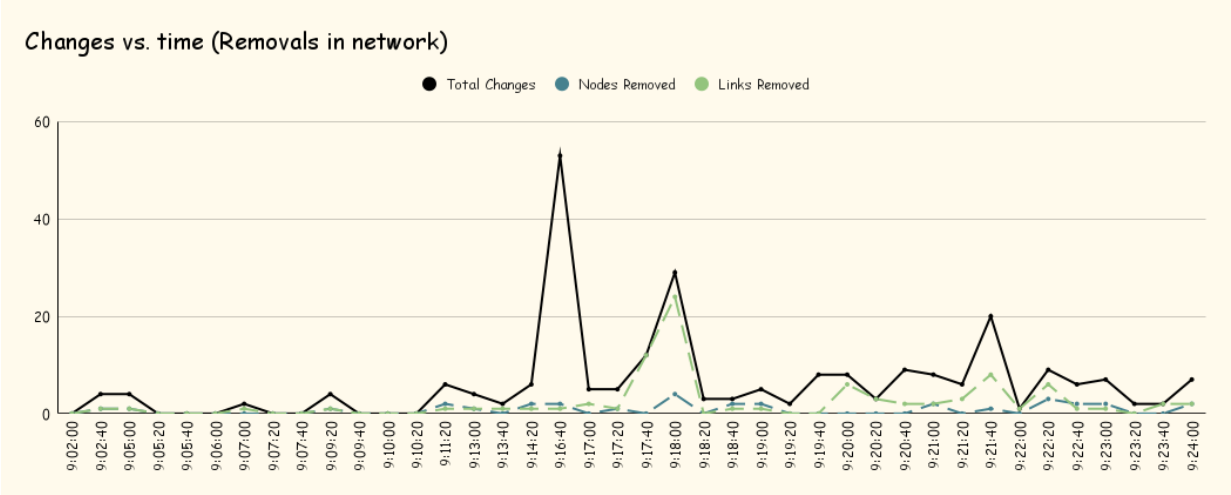


Figure 4.14: SFHH 2009: Line graph showing the change in number of nodes and links versus time, with network removals.

4.2.2 Validating Hierarchical Aggregation

We can use the same curves (Figs. 4.12, 4.13, 4.14) for validating hierarchical aggregation in our application. We prove the algorithm using bottom-up and top-down validation.

4.2.2.1 Bottom-up Validation

Upon adjusting the slider to its maximum value of 40, we are able to observe a complete view of all snapshots in this network. Since a number of events of duplicate activity and lesser significance can be noticed at the start, we can adjust the slider to reduce the number of frames value from 40 to, say, 33. This yields a strip of combined snapshots from 09:05:00 am until 09:10:20 am (Fig. 4.15), whose event activity can be validated against Figs. 4.13 or 4.14). Furthermore, the changes between the panels are correspondingly described in the captions. Therefore, our aggregation mechanism is validated from the bottom-up.

Figure 4.15: SFHH 2009: Aggregation of snapshots in initial phase of network activity.

4.2.2.2 Top-down Validaton

On adjusting the slider to a value of 1, we obtain a summary frame aggregating all the events in the network similar to before. Upon increasing the slider value to 2, we obtain a frame capturing all events until 09:14:20, and another capturing events from 09:16:40 upto the end (Fig. 4.16). This split can be verified upon referring to the activity in Figs. 4.13 or 4.14). The line charts show the highest peak occuring between the specified times in the network, once again validating our system for hierarchical aggregation from the top-down.

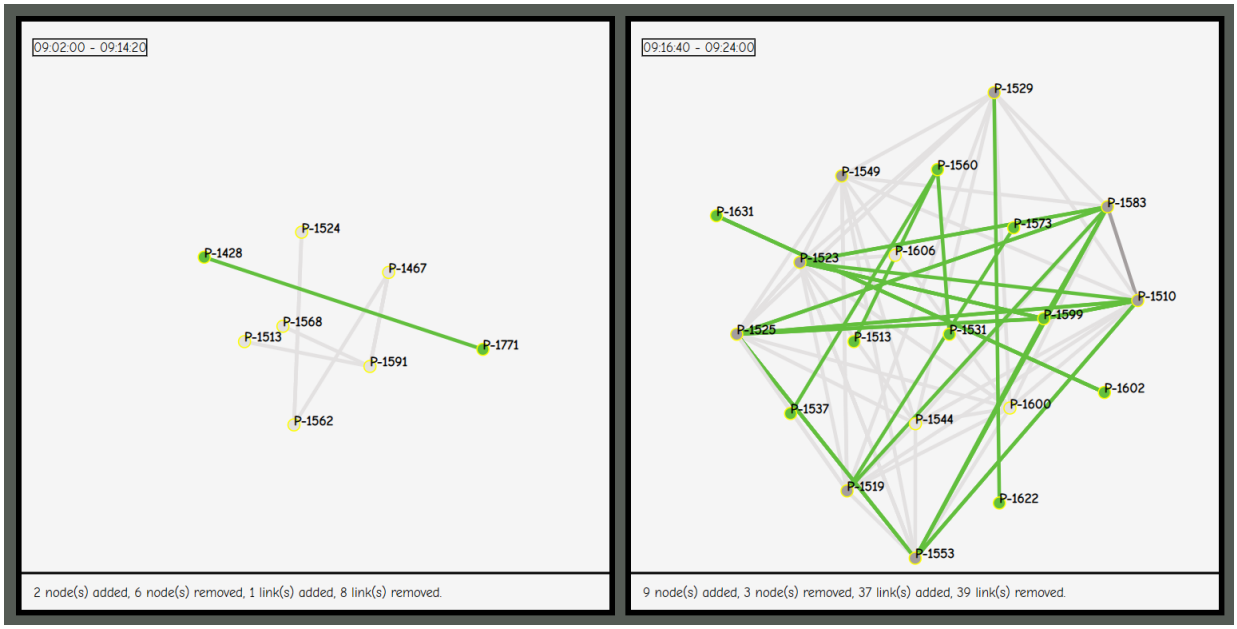


Figure 4.16: SFHH 2009: Split in frames between 09:14:20-09:16:40.

Chapter 5: Expert Review

We have been able to observe the effectiveness of our interface quantitatively against real-world data sets. In this section, we perform a qualitative study to assess the storytelling effectiveness of our interface. Some of these methods take inspiration with modifications from prior literature establishing data comics [2, 11].

5.1 Participants

We recruited a group of highly specific participants consisting primarily of visualization creators. We did so in order to gain feedback and a deeper level of understanding on the effectiveness of our tool for creation of dynamic graphs. We obtained 3 paid participants for our study who were selected through purposive sampling from the University of Maryland. An overview of the participant demographics is provided in table 5.1.

PARTICIPANT ID	OCCUPATION	AGE	GENDER	EDUCATION
P1	Master's Student	18-25	Male	Bachelor's in Computer Science
P2	Master's Student	18-25	Male	Bachelor's in Computer Science
P3	Doctoral Student	18-25	Male	Master's in Mech. Engineering

Table 5.1: **Participant Demographics.** Participant IDs, occupation, age, gender, and education background.

5.2 Apparatus

The setup consisted a standard laptop computer with a 14-inch screen and a browser running the application, along with an external mouse. We ran the application on the browser window in full screen mode.

5.3 Task and Datasets

In addition to an example dataset to illustrate the interface, we used two of the datasets we applied in our analysis earlier — Primate Interactions (D1), and Contact Tracing (D2). Each dataset upon selection generated all the snapshots in comic-style format, with the graph visualizations, timestamps and captions. Only a small portion of each dataset was considered in the application for simplicity. For each dataset, there was an accompanying set of 5 questions to facilitate engagement with the visualization, and these were provided to the users as soon as the visualizations were generated.

5.4 Metrics

Given that our objective was to obtain data on how visualization creators behaved with the interface for each dataset, we developed a post-completion survey in addition to the accompanying story-related questions. The survey consisted of 1-5 Likert scale questions, which were provided to participants after their analysis of both datasets. The survey questions were as follows:

- **Engagement**: How engaged and attentive did you feel by the story on the panel strip visualization?

- **Speed**: How did you feel about the speed of finding answers to questions? Were the visualizations helpful in being quick?
- **Visual Space**: How well does the story make use of visual space in the application?
- **Ease of Use**: How approachable and easy to use was the visualization strip to view and understand?
- **Enjoyability**: How enjoyable was the experience of viewing the strip?

5.5 Procedure

Our experiment session began with the participant arriving, signing a consent form and being assigned an identifier. An example dataset was visualized to demonstrate the interface, the meanings and visual cues in the generated graphs. The slider was used to demonstrate hierarchical aggregation, and the participant was encouraged to ask questions during this time. This process of demonstration lasted between 5-7 minutes, while questions and clarification lasted an additional time of up to 10 minutes.

Upon completion of the example demonstration, the participants were instructed to select the datasets to visualize, starting with the Contact Tracing (SFHH) dataset. Upon generating the visualization, the story-related questions were immediately provided to the participant on a separate sheet of paper along with stationery. The participants were encouraged to go back and forth between the interface and the questions to provide their answers suitably. This process was repeated for the Primates dataset.

For each dataset, the participants were encouraged to think aloud and express their thoughts

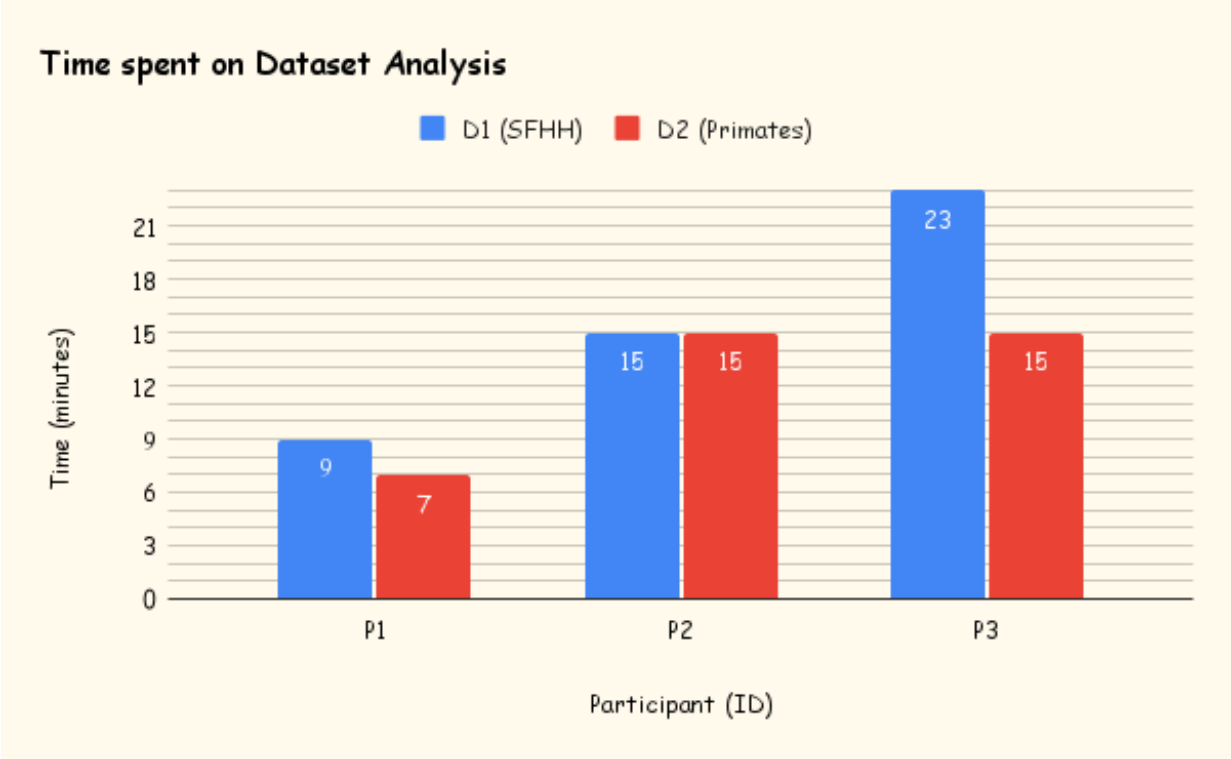


Figure 5.1: Time spent by participants on datasets D1 and D2, in minutes.

in their analysis of the visualization. Following the story analysis, the feedback survey was provided to obtain subjective feedback. A single experimental session lasted a maximum of 60 minutes.

5.6 Results

The results of this evaluation can be organized into quantitative and qualitative findings.

5.6.1 Quantitative Findings

Fig. 5.2 depicts a boxplot chart of subjective ratings obtained using the Likert scale from participants for the survey questionnaire. We find that enjoyability and engagement were rated significantly high among the participants, while the aspects of speed and visual space were

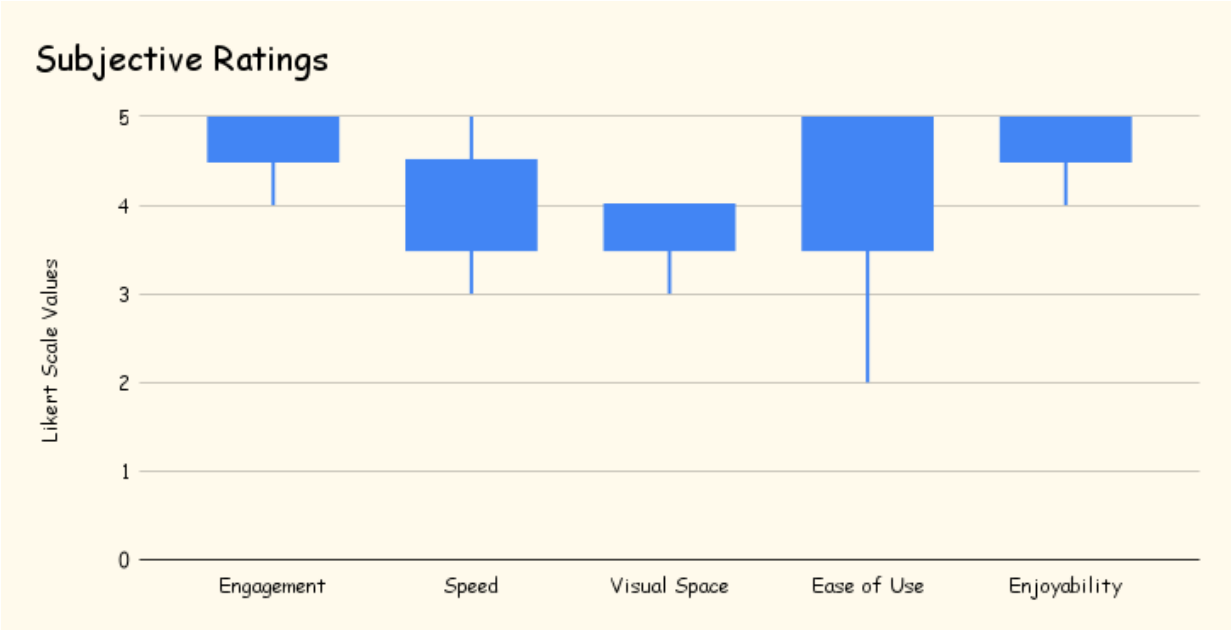


Figure 5.2: Expert Review: Likert Scale Ratings.

moderately favorable. We find that the ease of usage requires further introspection and discussion. We also have Fig. 5.1 that depicts the time spent by each participant on each of the two datasets (not including the example dataset demonstrated at the start of the session). Our analysis is that the time spent in analyzing the second dataset was less than or equal to the time spent on the first; both datasets had 40 time samples for consideration in our application.

5.6.2 Qualitative Findings

Based on our expert review, we broadly identify three major themes in our qualitative data, which are described below.

5.6.2.1 Analytical Communication

All three participants felt that the additive interactions in the graphs were conveyed very strongly, and emphasized that green as a color was a strong visual indicator for showing newly

formed nodes and links between snapshots. They also noted that the consistent location of nodes between panels helped track the network's activity across time. With these visual cues, the participants were able to imagine the scenarios from the network play out, and formed stories from the narrative. For instance, 2 participants coincidentally imagined – without cue – a set of nodes with high activity to be conference organizers in the SFHH dataset. In this regard, the captions in each panel were of relative importance; 1 participant made adequate use of captions in conjunction with the graphs to answer the story-related questions, while the remaining 2 participants did not feel the need to refer to them at all.

5.6.2.2 Active Engagement & Story Experience

As can be referred from Fig. 5.1, participants spent 7 minutes or more in analyzing each dataset. This took place with participants describing the visualization, and thinking aloud without distraction. All the participants described the major spikes in activity and non-major events in the datasets with the help of the visualization. The specific times of these active instances were noted correctly in answering the story-related questions. In fact, we found that participants clearly identified patterns, which they themselves validated with slider movement and hierarchical aggregation. As noted in Fig. 5.1, it is apparent that the interface becomes easier and quicker to use with time.

5.6.2.3 Hierarchical Aggregation

We found that the participants validated their own assumption about major points in the dataset storyline, using the slider for hierarchical aggregation. After viewing all the snapshots first, participants adjusted the slider to find out which frames would be combined. To provide

further context, P1 felt that the slider helped "average out" parts of the story, while P2 felt that adjusting the entire timeline to 3 frames helped identify "a clear start, middle and end" in the story. P3 felt that the repeated movement of the slider helped to "dive deep" into aggregated frames which were of interest. In a similar vein, P2 likened the analogy of slider movement for summarized and detailed views to that of swimming on the surface and seabed, respectively. In analyzing the Primates dataset second, P3 found answers significantly quicker in utilizing the slider movement right from the start, describing the interaction as "enjoyable" to summarize and zoom in to details of the story.

Chapter 6: Discussion

We find that our proof-of-concept prototype for rendering dynamic graphs has a series of benefits, as well as a number of limitations. In this section, we outline some of these advantages as well as drawbacks.

Our analysis with usage scenarios and user evaluations has indicated a potentially engaging and visually communicative way to represent time-series graph networks. While other visual representations such as line charts may provide statistical insights in comparison, the node-link diagrams in comic-style are able to provide several insights at a first glance with its capabilities, which we outline below.

- **Visual Cues:** For each generated panel, updates to the network are clearly visible in terms of node-link additions; a clean and effective communication of these changes between snapshots is able to be achieved with our interface. The ability to compare multiple snapshots at once presents an advantage as compared to other visual representations, and takes a step further towards simulating a real world network for detailed visual analysis. Such information also proves helpful in tracking a specific member or relationship as is validated by our expert review.

- **Summarization:** The frames with minimal or less meaningful changes are combined into adjacent snapshots using hierarchical aggregation, therefore allowing a user to view the

most significant events first. With the help of this aggregation mechanism, the desired number of frames is achieved with minimal loss of information.

- **Frame Selection:** The user is able to select a custom number of frames with the help of an interactive slider. On moving the slider, the user can increase or decrease the number of frames loaded on the strip. Such selection of frames is achieved using the tree formed as a result of the hierarchical aggregation algorithm mentioned above. Our participants found the slider interaction to be engaging and useful, in addition to providing validation for their own judgment on frames that should be combined.

- **Captions and Editable Text:** Our prototype generates captions that are insightful in describing the changes between each frame. Each caption is an editable line of text describing the number of nodes and links added or removed, compared to the frame preceding it. This caption can also be edited if the user wishes to do so. Additionally, the node labels in each graph panel can also be edited.

Given the above benefits of our work, there are also a number of limitations with our prototype worth mentioning. While participants found that green as a color was a strong visual indicator for network additions, 2 out of 3 participants mentioned that the grey color wasn't as suitable a choice for displaying incumbent or deleted elements of the network. We believe that adding a color picker with accommodations for color vision deficiency (CVD) can help address this issue in the interface. Provisions for CVD would also hep in improving the addressable audience for such a tool.

Our implementation of hierarchical aggregation – though simple and effective – can occasionally be hard to fully comprehend, as we observe from our user evaluation. While hierarchical aggregation

is a method that has been widely implemented as a method for summarization [22, 23], we acknowledge that steps can be taken to provide a better means of comprehension in our interface.

In our qualitative analysis, we found that P3 was the only participant that relied on the captions for statistical analysis; however, P2 felt that the captions were certainly important even if they weren't used for their own study in the limited time available. In the design of our application, we certainly believe that the captions are not only of descriptive benefit, but aid in providing an engaging way of storytelling with complex data. We believe that results from a more comprehensive user study would validate the efficiency of captions as a storytelling aid.

In addition to the advantages and drawbacks listed above, there are some more major steps we can take to address our work, some of which we discuss below.

- **Advanced Visual Operations:** Our prototype is currently unable to perform operations such as filtering, ranking and cluster grouping which are especially useful in visual analytics. The grouping mechanism could be especially beneficial in our prototype given that certain members of the network may exhibit a common behavior at a given moment. Our prototype also lacks the ability to change the shape or properties of nodes and links. However, our system is generalizable to include these operations, given the object oriented approach in the design of our system.

- **Distance Metrics:** We have implemented a single distance metric that measures the magnitude of changes between snapshots. However, we would like to be able to provide users with the ability to choose a distance metric more suited to their needs.

- **Further Validation**: While we have been able to assess this interface against real world usage scenarios, and evaluate it against an expert audience, we acknowledge that our work

can benefit greatly from further evaluation with an increased number of users.

- **Comic-style Storytelling**: Our work can certainly take more inspiration from the communicative language of comics, apart from basic panels with captions. There are a number of elements such as motion lines, highlighting and panel layouts (eg. pan and zoom) to make our prototype more engaging through visual narration. We certainly believe it should prove exciting to explore and investigate the effectiveness of comic-style elements as established by several authors in recent years [11, 13].

Chapter 7: Conclusion & Future Work

We have presented a new approach with our proof of concept prototype to visualize a dynamic graph comic, from dense temporal edge-vertex data. We implemented a browser-based application that uses hierarchical aggregation to cluster multiple snapshots in time, giving users the choice to customize the number of frames. We finally evaluated the application with real world usage scenarios and an expert audience to suggest its usefulness as a tool.

Our next immediate step will be to prepare and make changes to this interface for further study among users. Future work will also include the ability to provide users with more advanced visual manipulations, distance metrics, and increased comic-style engagement. In addition, being proponents of accessibility at the University of Maryland, we would like to explore making this application more accessible to people using assistive devices, and thereby widen its addressable audience.

Bibliography

[1] Andre Suslik Spritzer, Jeremy Boy, Pierre Dragicevic, Jean-Daniel Fekete, and Carla Maria Dal Sasso Freitas. Towards a smooth design process for static communicative node-link diagrams. 34(3):461–470, Jun 2015.

[2] Zhenpeng Zhao, Rachael Marr, and Niklas Elmqvist. Data comics: Sequential art for data-driven storytelling. Technical Report. Human Computer Interaction Lab, University of Maryland. http://www.cs. umd.edu/hcil/trs/2015-15/2015-15.pdf, 2015.

[3] Valeria Gelardi, Jeanne Godard, Dany Paleressompoulle, Nicolas Claidiere, and Alain Barrat. Measuring social networks in primates: wearable sensors versus direct observations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2236):20190737, Apr 2020.

[4] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. A taxonomy and survey of dynamic graph visualization. *Comput. Graph. Forum*, 36(1):133–159, Jan 2017.

[5] Benjamin Bach, Natalie Kerracher, Kyle Wm. Hall, Sheelagh Carpendale, Jessie Kennedy, and Nathalie Henry Riche. Telling stories about dynamic networks with graph comics. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 3670–3682, New York, NY, USA, 2016. Association for Computing Machinery.

[6] Colin Ware. *Information visualization: perception for design*. Morgan Kaufmann, San Francisco, CA, 2004.

[7] Edward Segel and Jeffrey Heer. Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1139–1148, 2010.

[8] Scott McCloud. *Understanding Comics*. Kitchen Sink Press, 1993.

[9] Scott McCloud. *Making Comics*. Harper, 2006.

[10] Nahum Gershon and Ward Page. What storytelling can do for information visualization. *Commun. ACM*, 44(8):31–37, Aug 2001.

[11] Zhenpeng Zhao, Rachael Marr, Jason Shaffer, and Niklas Elmqvist. Understanding partitioning and sequence in data-driven storytelling: The case for comic strip narration. In *Proceedings of the iConference*, volume 11420, pages 327–338. Springer, 2019.

[12] Benjamin Bach, Nathalie Henry Riche, Sheelagh Carpendale, and Hanspeter Pfister. The emerging genre of data comics. *IEEE Computer Graphics and Applications*, 37(3):6–13, 2017.

[13] Zezhong Wang, Hugo Romat, Fanny Chevalier, Nathalie Henry Riche, Dave Murray-Rust, and Benjamin Bach. Interactive data comics. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):944–954, 2022.

[14] Zezhong Wang, Shunming Wang, Matteo Farinella, Dave Murray-Rust, Nathalie Henry Riche, and Benjamin Bach. Comparing effectiveness and engagement of data comics and infographics. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA, 2019. Association for Computing Machinery.

[15] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, Jun 1998.

[16] Bret Victor. Scientific communication as sequential art, 2011.

[17] Nam Wook Kim, Eston Schweickart, Zhicheng Liu, Mira Dontcheva, Wilmot Li, Jovan Popovic, and Hanspeter Pfister. Data-driven guides: Supporting expressive design for information graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):491–500, 2017.

[18] Nam Wook Kim, Nathalie Henry Riche, Benjamin Bach, Guanpeng Xu, Matthew Brehmer, Ken Hinckley, Michel Pahud, Haijun Xia, Michael McGuffin, and Hanspeter Pfister. Datatoon: Drawing dynamic network comics with pen + touch interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19. ACM, 2019.

[19] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: an open source software for exploring and manipulating networks. page 361–362, 2009.

[20] yed. http://www.yworks.com/yed, 2014.

[21] Zhenpeng Zhao, William Benjamin, Niklas Elmqvist, and Karthik Ramani. Sketcholution: Interaction histories for sketching. *International Journal of Human-Computer Studies*, 82, Apr 2015.

[22] Benjamin Bach, Nathalie Henry Riche, Tim Dwyer, Tara Madhayastha, Jean-Daniel Fekete, and Tom Grabowski. Small multipiles: Piling time to explore temporal patterns in dynamic networks. In *In Proceedings of Eurographics Conference on Visualization (EuroVis)*, volume 34, Jan 2015.

[23] Michael Burch and Daniel Weiskopf. A flip-book of edge-splatted small multiples for visualizing dynamic graphs. In *Proceedings of the 7th International Symposium on Visual Information Communication and Interaction*, VINCI '14, page 29–38, New York, NY, USA, 2014. Association for Computing Machinery.

[24] Mathieu Génois and Alain Barrat. Can co-location be used as a proxy for face-to-face contacts? 7:1–18, Dec 2018.

[25] Ciro Cattuto, Wouter Van Den Broeck, Alain Barrat, Vittoria Colizza, Jean-François Pinton, and Alessandro Vespignani. Dynamics of person-to-person interactions from distributed rfid sensor networks. *PLoS ONE*, 5(7):e11596, Jul 2010.

[26] Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Vittoria Colizza, Lorenzo Isella, Corinne Regis, Jean-François Pinton, Nagham Khanafer, Wouter Van den Broeck, and Philippe Vanhems. Simulation of an seir infectious disease model on the dynamic contact network of conference attendees. *BMC Medicine*, 9(87), Jul 2011.