# An Overview of Visualizing Dynamic Graphs

Patricia J Sazama

February 9, 2015

## 1 Abstract

Dynamic graphs are graphs that represent networks that have a time-varying component. A great deal of processes in the world can be modeled as such graphs but cannot be modeled as their static counterparts. This paper discusses the approaches that have been proposed in the literature to visualize these dynamic graphs and the applications which they have been used for. This paper intends to provide an overview of the variety of visualization techniques that have been applied and the special considerations for time-varying data that have been identified.

## 2 Background

Many aspects of the world can be modeled as objects with relationships to each other, from individuals in a social network, a biological network of the human body, or pieces of a software system communicating and exchanging data. There has been a great deal of research on how to visualize such relationships as graphs, however the vast majority of the literature focuses on static graphs only. These fail to capture the dynamic relationships in a rarely static world and there has been steadily growing interest in visualizing these relationships as dynamic graphs.

When considering graphs as dynamic rather than static a host of new challenges appear for visualization. Making sense of time-varying data is often more complicated than simply displaying each time step of the graph and there have been many different approaches attempting to identify the best form of visualization. The first major distinction in these visualization approaches is how time is represented. There are two major ways that the time component has been represented so far, through animation or through use of a timeline [4, 5], though there have also been some hybrid combinations of the two. The animation method is a time-to-time mapping of the sequence of graphs to the visualization time steps and the timeline method maps the time dimension to a spatial dimension [6].

Within each of those types of time representations there are further divides in the types of approaches taken. This paper discusses different types of dynamic graph visualizations that have been investigated, the unique challenges for dynamic graphs, and the applications that have used them thus far.

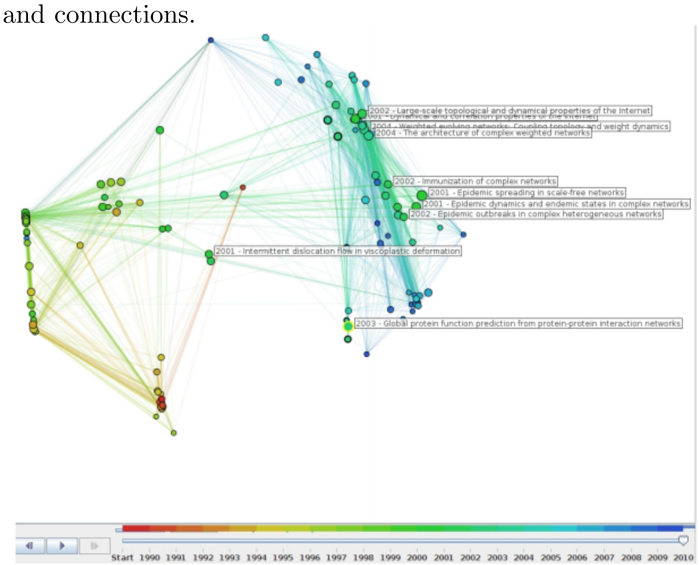
## 3 Animated Dynamic Graphs

So far all approaches that use animation to represent time use a node-link diagram as a basis. These node-link diagrams are the traditional way many static graphs are visualized, but there are several variations in how time is visualized and several new concepts that come into play. One important concept is that of the mental map. This is the structural information a user generates by looking at the layout of a graph. In order to aid in user-understanding when changes are made to the graph the position of the nodes should be kept stable.

In general, the animated approaches have a large visualization of the node-link graph that removes and adds nodes and edges at each time step, and the user is free to navigate to different time points in the animation.

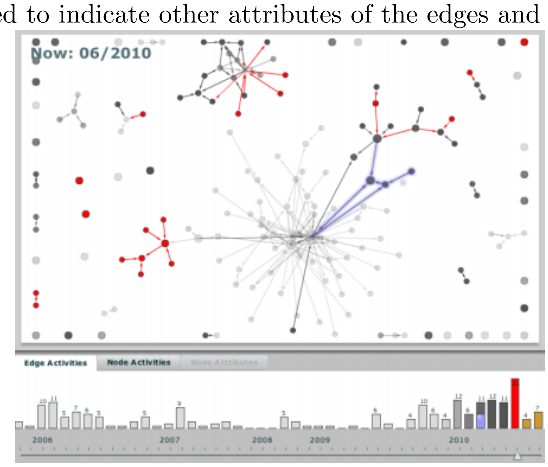
In order to preserve the time at which certain nodes appeared, some visualizations have used a color scale in which nodes and edges that appear earlier on in the graph are at one end of the scale and nodes and edges that appear closer to the end are at the other end. This allows for the graph to grow over time but still preserve a sense of how old different nodes are. This approach was only applied on a subject where nodes and edges were never removed so preserving the notion of a former connection was not needed [3].

Figure 1: The visualization technique proposed by [3] displays a large node-link diagram that updates at each new time step and uses color to indicate the age of nodes and connections.



Some approaches have indicated the age of a node or edge using a fade out effect, where older edges and nodes are painted at lower intensities. In addition, for visual comparisons users can set a color for all edges or nodes from a specific time point and anchor that so that it can be easily compared against multiple points in time. In applications where there are multiple types of edges, these colors can be used to highlight a type of edge and a time point for easy comparison against other types of edges at that same time point or a different time[2].

Figure 2: One of the visualization technique proposed by [2], TempoVis uses a fading effect to indicate the age of nodes and connections and also allows for colors to be used to indicate other attributes of the edges and nodes.




These scale-based ways of indicating the age of a node or edge must take into account the total length of time covered by the data set and so can generally only be applied to offline graphs. An exception can be made in cases where the application allows for some cut-off time at which nodes and edges that have been connected since before the cut-off can disappear or have the same color or intensity on the scale.

Often, in animated dynamic graphs there will be a small timeline which is primarily used as navigation for the user-interface. Since some of these navigational timelines contain some type of overview of the graph at each timestep it could be argued that these approaches are a hybrid visualization method, although the primary visualization is the animated node-link diagram[2].

Within animated dynamic graphs there has been work on both online [30, 9, 26] and offline graphs [16, 25, 22]. The difference between the two is whether the graph is created in real time or if preprocessing can be done on the entire graph evolution. There are a number of unique challenges in the online case, one of the biggest ones is the initial layout of the graph. Since it is widely considered best to have little to no movement of nodes from one time frame to another the initial layout of the graph is generally best determined by considering all future edges and nodes. In the online case all future edges and nodes are not known so there has been special consideration for how to best determine the initial positioning of the graph[26].

One of the biggest problems with node-layout graphs is the same as in their static graph counterparts. In terms of scalability it can very quickly become impossible to identify fine details in the graph as more edges and nodes are added.

# 4 Timelines

The other representation of time that has been explored is the timeline representation, and there has been both node-link diagram approaches and matrix-based approaches within this representation. This has primarily been done with small datasets since scaling to larger datasets often makes a timeline graph unreadable.

Within the node-based representation there have been several approaches to displaying sequential graphs on a timeline including juxtaposition, superimposition, integrating, and hybrid methods. Juxtaposing has been done by juxtaposing entire graphs next to each other or by aligning individual nodes by their previous and future time step so that as the timeline progresses the changes in the node are more apparent [4, 14].

One approach for timelines is to use a circular structure. Juxtaposition has been used in circular timelines by representing the graph as a series of concentric circles, where each circle contains all the nodes of the graphs, and there are connections between pairs of concentric circles to represent the edges between the nodes at a particular time step[14].

Figure 3: Timeline representation styles using juxtaposition from [14], from displaying node-based graphs in sequence, to a timeline representation that aligns nodes along a single line to represent each time step, to a circular timeline layout.

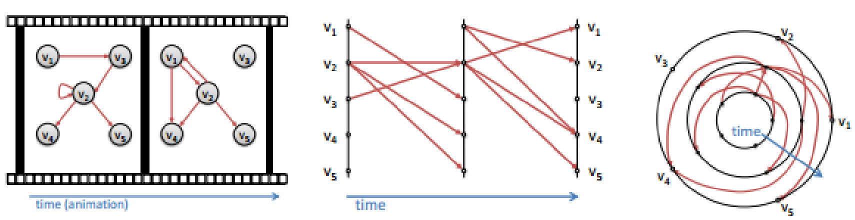
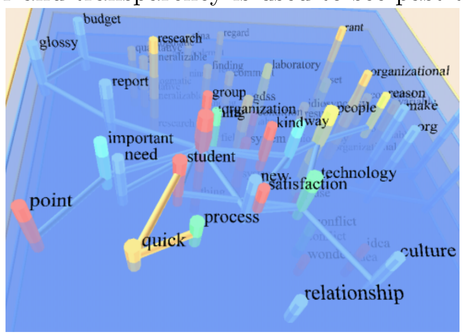


Figure 4: A superimposition timeline representation style from [20], where each time frame is a layer and transparency is used to see past time frames.




In superimposition graphs from different time steps can be stacked on top of each other[20, 7, 1]. In superimposition there have been several approaches to describe the temporal nature of the nodes and edges in one aggregate image. Some approaches choose specific colors to indicate that a node or edge has persisted across multiple time steps and uses colors to indicate which time steps it came from in a similar approach to the color scales mentioned in the animated approaches[20].

In integration approaches the timeline is often combined with the node-link diagram so that the diagrams for the individual time steps are no longer separable[33, 37]. The integration approaches tend to divide up the original network and combined pieces of it to highlight certain aspects of the network, and can offer different visualizations based on what aspect of the graph is being examined. These often use glyphs to represent the underlying data to avoid clutter and focus on the aspect of the data that the user is interested in.

Some strategies have taken a hybrid approach between juxtaposition, superimposition, and integration. Some layer the graphs on top of each other so that, when viewed from the side, the graph at one time step can be directly compared with the graph at the next, or they can be viewed from above where the entire stack of graphs can be seen merged together by adding some transparency to the layered graphs[7, 27]. A similar approach to this added connection between the same nodes in different graph time steps, creating long tube-like structures. The graph design also allowed for two additional temporal attributes, symbolized by a color gradient along each tube and increases or decreases of the radius of the tube at various time points[1]. Other hybrid methods include multiple views where users can choose to view the superimposed graph, juxtaposed graphs, or a two-and-a-half dimension representation that is a combination of the two and allows for easier tracking of changes in a node across the juxtaposed time steps[21].

One hybrid technique used a node-based layout with a timeline in the third dimension. This converted the nodes into long cylindrical objects in the zdirection that grew and shrunk to show changes in that node. In a variant the cylinders were converted into ”worms” where the cylinders would bend towards and away from other nodes to indicated hos closely the two nodes were related[17].

Figure 5: The hybrid model proposed by [1] in which tubes connect each superimposed graph and variations in width and color indicate various attributes.

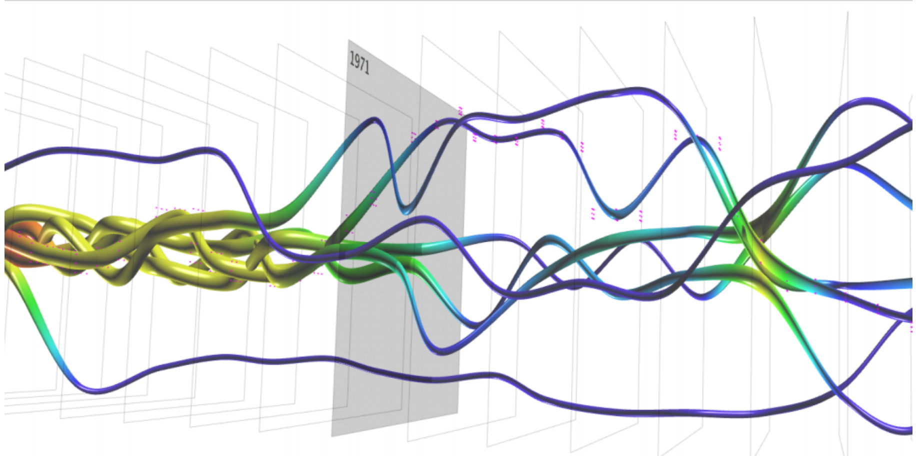
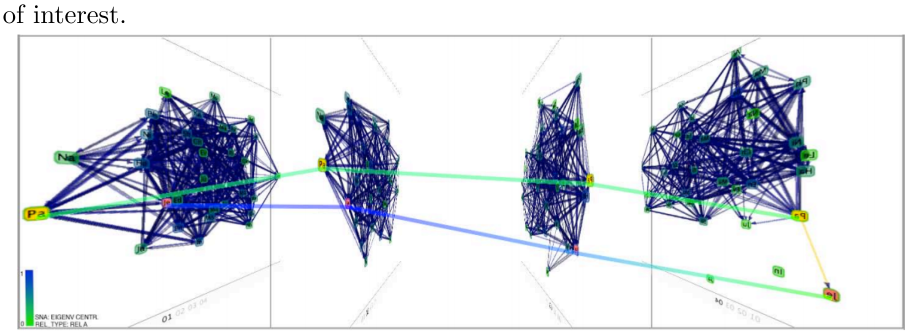



Figure 6: A hybrid model proposed by [21], where a two-and-a-half dimension representation allows the user to see multiple full graphs while tracking a point of interest.




Some of these strategies have user options that allow for controlling how simplified the edges and nodes of the graph are. These can range from very simplified, where edges and nodes are combined and visualizing the big picture is the key goal, to no simplification at all so a user can inspect the fine details of the graph on a single node level.

Within the adjacency matrix-based representations the visualizations include intra-cell timelines [12, 40] and layered matrices [11, 38], the primary difference between the two being whether the adjacency matrix has the timelines within each cell or whether the adjacency matrices are juxtaposed.

Figure 7: An intra-cell timeline visualization approach proposed by [12] where each cell has a timeline representing the events that happened to that particular cell.

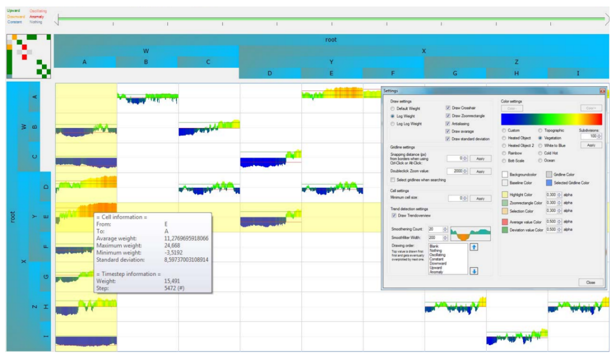
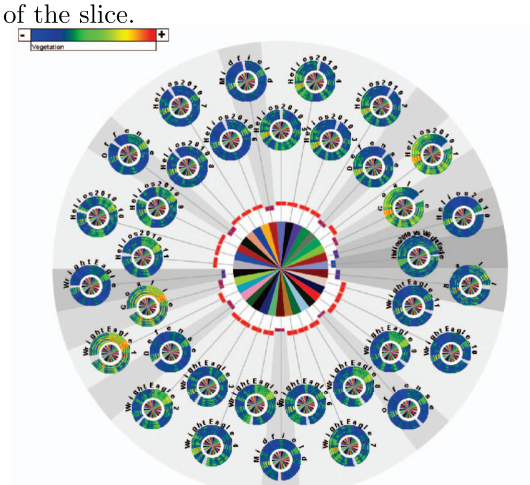


Figure 8: A layered matrix approach used by [11], where each cell is displayed as a slice of a circle and the time element is indicated by the number of steps from the center of the slice.



Matrix-based approaches attempt to overcome the scalability issue that is often associated with node-based approaches. They can represent large and dense networks although they may not preserve spatial relationships intrinsic to node-based designs[40]. In some approaches, cells of these matrices can be expanded to show the changes over time of the cell. Some can show the changes over multiple cells, given a user selection. These types of matrix-based explorations have been applied in both traditional square, grid-like matrices as well as in circular matrices[11, 40, 38]. In circular representations steps in the timeline have have been shown using concentric circles, similar to circular approaches using juxtaposition in the node-link timeline approaches.

# 5 Hybrids

Most dynamic graph visualizations can be classified as either timeline-based or animation-based, however there are a few that use hybrid combinations of animations and timelines. Such approaches include embedding small animated graphs into a larger timeline and allowing the user to navigate to these animations[36, 35]. Some of the previous approaches described use a timeline as the navigation tool to explore the animated node-link graph, but the visualization itself is still primarily the node-link animation.

# 6 Additional Considerations

When considering the illustration of graphs using the time dimension there are often many additional considerations that do not arise in their static counterparts or that are not as considerable issues. One such effect was the contagion effect studied in [39]. In graph visualization often nodes can be collapsed to remove clutter. When one node collapses its neighbors are collapsed as well and this can lead to a chain-reaction in the visualization. The effects of the chain-reaction are usually clear on static graphs, but in time-varying graphs it is difficult to identify all the across-time effects of the node collapse.

When considering a dynamic graph compared to a static graph, several new concepts emerge. One of these is the concept of alignment. When considering a node at one time point compared to another, in order for the user to understand it more quickly the locations of the nodes should remain consistent. This ties in with the concept of the mental map. This is the user's mental interpretation of the graph and, in practice, it is easiest for a user to understand a graph if their mental map is preserved from one time step to another. Changing a node's position can cause discrepancies between the user's mental map and the dynamic graph. Often this does force the use of some foresight when choosing where to place nodes on the graph and this can be a problem in online graphs. There have been multiple approaches to try and solve the node-layout problem in such a way that decreases clutter and preserves the mental map as much as possible[4, 9].

## 7 Applications

Dynamic graph visualization has been used so far primarily in a few topics. It has been used for the exploration of social data in social networking and other online communities [31, 8, 32, 2], visualizing changes and relationships between documents [19, 3, 23], and visualizing changes in software and program executions [18, 24, 15]. While the previously mentioned areas are the major applications for visualizing dynamic graphs there has been some work starting in biological, psychological, and business areas [34, 28, 10, 13, 29, 39, 17].

In business areas, specifically around the topic of financial networks, there has been work to address some of the specific issues that arise from time-varying graph visualizations[39, 17]. In academia there has been exploration in using dynamic graphs to visualize the changing contents of document collections, specifically the published academic papers in a certain field. These can show trends, emerging sub-topics, and can provide a broad summary from multiple documents at once[3]. In other related work, visualization of academic literature has been used to add authors into the dynamic graph, showing collaborations and interactions between different research specialties[19].

For biological purposes, dynamic graphs have been used to visualize a number of areas. They have been used to aid in understanding areas of interest for a user in eye-tracking studies[13] and for gaining insight in models of chemical reaction networks and metabolic pathways[28, 34]. Dynamic graphs have also been frequently used to map changes to software during development. The graphs in these cases can show which parts of the program are unstable over long periods of time, which programmers are responsible for which parts over what time periods, and to understand the overall evolution of a large piece of software[15]. Beyond the development of software, dynamic graphs have also been used to visualize the activities of a running piece of software. Understanding the way the software works can be easier when there are visualizations for system activities such as when certain parts of a system start up, begin to communicate, or stop communicating[18]. These types of visualizations have also been extended to detect anomalies in large network systems[29].

## 8 Conclusion

Dynamic graphs are a crucial tool in understanding time-varying networks. They will be able to provide intuitive understanding of complex processes and allow the users that interact with them to make natural inferences about those complex processes. Interest in the area has been growing and many useful visualization approaches have come from it. As research continues there will be more techniques to visualize and gain insight from datasets and our ability to work with and understand complex processes will drastically increase.

## References


[1] Interactively visualizing dynamic social networks with dyson. In *Workshop on Visual Interfaces to the Social and the Semantic Web (VISSW2009)*, February 2009.

[2] Jae-wook Ahn, Meirav Taieb-Maimon, Awalin Sopan, Catherine Plaisant, and Ben Shneiderman. Temporal visualization of social network dynamics: Prototypes for nation of neighbors. In John Salerno, Shanchieh Jay Yang, Dana Nau, and Sun-Ki Chai, editors, *Social Computing, Behavioral-Cultural Modeling and Prediction*, volume 6589 of *Lecture Notes in Computer Science*, pages 309–316. Springer Berlin Heidelberg, 2011.

[3] Aretha B. Alencar, Katy Börner, Fernando V. Paulovich, and Maria Cristina F. de Oliveira. Time-aware visualization of document collections. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 997–1004, New York, NY, USA, 2012. ACM.

[4] F. Beck, M. Burch, and S. Diehl. Towards an aesthetic dimensions framework for dynamic graph visualisations. In *Information Visualisation, 2009 13th International Conference*, pages 592–597, July 2009.

[5] F. Beck, M. Burch, and S. Diehl. Matching application requirements with dynamic graph visualization profiles. In *Information Visualisation (IV), 2013 17th International Conference*, pages 11–18, July 2013.

[6] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. The state of the art in visualizing dynamic graphs. In *Proceedings of the Eurographics Conference on Visualization (EuroVis 14)–State of The Art Reports*, 2014.

[7] Ulrik Brandes and Steven R. Corman. Visual unrolling of network evolution and the analysis of dynamic discourse. *Information Visualization*, 2(1):40–50, 2003.

[8] Ulrik Brandes, Natalie Indlekofer, and Martin Mader. Visualization methods for longitudinal social networks and stochastic actor-oriented modeling. *Social Networks*, 34(3):291 – 308, 2012. Dynamics of Social Networks (2).

[9] Ulrik Brandes and Dorothea Wagner. A bayesian paradigm for dynamic graph layout. In Giuseppe DiBattista, editor, *Graph Drawing*, volume 1353 of *Lecture Notes in Computer Science*, pages 236–247. Springer Berlin Heidelberg, 1997.

[10] Steffen Brasch, Georg Fuellen, and Lars Linsen. Venlo: Interactive visual exploration of aligned biological networks and their evolution. In Lars Linsen, Hans Hagen, Bernd Hamann, and Hans-Christian Hege, editors, *Visualization in Medicine and Life Sciences II*, Mathematics and Visualization, pages 229–247. Springer Berlin Heidelberg, 2012.

[11] M. Burch, M. Hoferlin, and D. Weiskopf. Layered timeradartrees. In *Information Visualisation (IV), 2011 15th International Conference on*, pages 18–25, July 2011.

[12] M. Burch, B. Schmidt, and D. Weiskopf. A matrix-based visualization for exploring dynamic compound digraphs. In *Information Visualisation (IV), 2013 17th International Conference*, pages 66–73, July 2013.

[13] Michael Burch, Fabian Beck, Michael Raschke, Tanja Blascheck, and Daniel Weiskopf. A dynamic graph visualization perspective on eye movement data. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '14, pages 151–158, New York, NY, USA, 2014. ACM.

[14] Michael Burch, Fabian Beck, and Daniel Weiskopf. Radial edge splatting for visualizing dynamic directed graphs. In *GRAPP/IVAPP*, pages 603–612, 2012.

[15] Christian Collberg, Stephen Kobourov, Jasvir Nagra, Jacob Pitts, and Kevin Wampler. A system for graph-based visualization of the evolution of software. In *Proceedings of the 2003 ACM Symposium on Software Visualization*, SoftVis '03, pages 77–ff, New York, NY, USA, 2003. ACM.

[16] Stephan Diehl, Carsten Grg, and Andreas Kerren. Preserving the mental map using foresighted layout. In DavidS. Ebert, JeanM. Favre, and Ronald Peikert, editors, *Data Visualization 2001*, Eurographics, pages 175–184. Springer Vienna, 2001.

[17] T. Dwyer and P. Eades. Visualising a fund manager flow graph with columns and worms. In *Information Visualisation, 2002. Proceedings. Sixth International Conference on*, pages 147–152, 2002.

[18] N. Elmqvist and P. Tsigas. Causality visualization using animated growing polygons. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pages 189–196, Oct 2003.

[19] Cesim Erten, Philip J. Harding, Stephen G. Kobourov, Kevin Wampler, and Gary Yee. Exploring the computing literature using temporal graph visualization, 2004.

[20] Cesim Erten, Stephen G. Kobourov, Vu Le, and Armand Navabi. Simultaneous graph drawing: Layout algorithms and visualization schemes. In Giuseppe Liotta, editor, *Graph Drawing*, volume 2912 of *Lecture Notes in Computer Science*, pages 437–449. Springer Berlin Heidelberg, 2004.

[21] Paolo Federico, Wolfgang Aigner, Silvia Miksch, Florian Windhager, and Lukas Zenk. A visual analytics approach to dynamic social networks. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*, i-KNOW '11, pages 47:1–47:8, New York, NY, USA, 2011. ACM.

[22] Kun-Chuan Feng, Chaoli Wang, Han-Wei Shen, and Tong-Yee Lee. Coherent time-varying graph drawing with multifocus+context interaction. *Visualization and Computer Graphics, IEEE Transactions on*, 18(8):1330–1342, Aug 2012.

[23] Emden R. Gansner, Yifan Hu, and Stephen C. North. Interactive visualization of streaming text data with dynamic maps. *Journal of Graph Algorithms and Applications*, 17(4):515–540, 2013.

[24] Orla Greevy, Michele Lanza, and Christoph Wysseier. Visualizing live software systems in 3d. In *Proceedings of the 2006 ACM Symposium on Software Visualization*, SoftVis '06, pages 47–56, New York, NY, USA, 2006. ACM.

[25] Carsten Grg, Peter Birke, Mathias Pohl, and Stephan Diehl. Dynamic graph drawing of sequences of orthogonal and hierarchical graphs. In Jnos Pach, editor, *Graph Drawing*, volume 3383 of *Lecture Notes in Computer Science*, pages 228–238. Springer Berlin Heidelberg, 2005.

[26] A Hayashi, T. Matsubayashi, T. Hoshide, and T. Uchiyama. Initial positioning method for online and real-time dynamic graph drawing of time varying data. In *Information Visualisation (IV), 2013 17th International Conference*, pages 435–444, July 2013.

[27] M. Itoh, M. Toyoda, and M. Kitsuregawa. An interactive visualization framework for time-series of web graphs in a 3d environment. In *Information Visualisation (IV), 2010 14th International Conference*, pages 54–60, July 2010.

[28] Mathias John, Hans-Jrg Schulz, Heidrun Schumann, AdelindeM. Uhrmacher, and Andrea Unger. Constructing and visualizing chemical reaction networks from pi-calculus models. *Formal Aspects of Computing*, 25(5):723–742, 2013.

[29] Q. Liao, L. Shi, and C. Wang. Visual analysis of large-scale network anomalies. *IBM Journal of Research and Development*, 57(3/4):13:1–13:12, May 2013.

[30] Kazuo Misue, Peter Eades, Wei Lai, and Kozo Sugiyama. Layout adjustment and the mental map. *Journal of Visual Languages & Computing*, 6(2):183 – 210, 1995.

[31] James Moody, Daniel McFarland, and Skye Bender-deMoll. Dynamic network visualization1. *American Journal of Sociology*, 110(4):1206–1241, 2005.

[32] S. Pupyrev and A Tikhonov. Analyzing conversations with dynamic graph visualization. In *Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on*, pages 748–753, Nov 2010.

[33] Florian Reitz. A framework for an ego-centered and time-aware visualization of relations in arbitrary data repositories. *CoRR*, abs/1009.5183, 2010.

[34] Markus Rohrschneider, Alexander Ullrich, Andreas Kerren, PeterF. Stadler, and Gerik Scheuermann. Visual network analysis of dynamic metabolic pathways. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Ronald Chung, Riad Hammoud, Muhammad Hussain, Tan Kar-Han, Roger Crawfis, Daniel Thalmann, David Kao, and Lisa Avila, editors, *Advances in Visual Computing*, volume 6453 of *Lecture Notes in Computer Science*, pages 316–327. Springer Berlin Heidelberg, 2010.

[35] 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, Dec 2013.

[36] Arnaud Sallaberry, Chris Muelder, and Kwan-Liu Ma. Clustering, visualizing, and navigating for large dynamic graphs. In Walter Didimo and Maurizio Patrignani, editors, *Graph Drawing*, volume 7704 of *Lecture Notes in Computer Science*, pages 487–498. Springer Berlin Heidelberg, 2013.

[37] S. Van Den Elzen, D. Holten, J. Blaas, and J.J. van Wijk. Dynamic network visualization with extended massive sequence views. *Visualization and Computer Graphics, IEEE Transactions on*, 20(8):1087–1099, Aug 2014.

[38] C. Vehlow, M. Burch, H. Schmauder, and D. Weiskopf. Radial layered matrix visualization of dynamic graphs. In *Information Visualisation (IV), 2013 17th International Conference*, pages 51–58, July 2013.

[39] Tatiana von Landesberger, Simon Diel, Sebastian Bremm, and Dieter W Fellner. Visual analysis of contagion in networks. *Information Visualization*, 2013.

[40] Ji Soo Yi, Niklas Elmqvist, and Seungyoon Lee. Timematrix: Analyzing temporal social networks using interactive matrix-based visualizations. *International Journal of Human-Computer Interaction*, 26(11-12):1031–1051, 2010.