As Time Goes by – Integrated Visualization and Analysis of Dynamic Networks

Mathias Pohl
Dept. for Software Engineering University of Trier
pohlm@uni-trier.de

Florian Reitz
Dept. for Software Engineering University of Trier
reit4701@uni-trier.de

Peter Birke
Dept. for Databases and Information Systems University of Trier
birke@uni-trier.de

ABSTRACT

The dynamics of networks have become more and more important in all research fields that depend on network analysis. Standard network visualization and analysis tools usual do not offer a suitable interface to network dynamics. These tools do not incorporate specialized visualization algorithms for dynamic networks but only algorithms for static networks. This results in layouts that bother the user with too many layout changes which makes it very hard to work with them.

To handle dynamic networks the DGD-tool was implemented. It does not only provide several layout algorithms that were designed for dynamic networks but also different instruments for statistical network analysis. Network visualization and statistics are combined in a multiple view interface that allows visual comparison of several network layouts and several network metrics at the same time. Furthermore the time-dependent behaviour of structural changes becomes visible and facilitates the analysis of network dynamics.

Categories and Subject Descriptors

H.5.2 [**Information Systems Applications**]: User Interfaces—*Graphical user interfaces*; I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Interaction techniques*; G.2.2 [**Discrete Mathematics**]: Graph Theory

General Terms

Dynamics of Networks, Human-Centered Visual Analytics

Keywords

Multiple and Integrated Views, Dynamic Network Visualization

1. INTRODUCTION

1.1 Motivation

In the recent years network theory has moved into the focus of social scientists. Relations between people, institutions, or other types of actors within a specified environment are considered to be more than just results of behaviors. These structures may also influence the actors' descisions in the future. To facilitate the analysis of such social networks researchers implemented visualization and analysis tools such as Pajek [1], UCInet (together with NetDraw) [2] or Visone [3]. However, these tools do not provide a suitable interface to cope with network dynamics. As this dynamics is another interesting and expressive dimension of information there was a need for a specifically designed visualization and analysis tool. This tool should be suitable to present a visual interface to time-varying networks on standard hardware (especially on standard displays). To achieve that goal we implemented the DGD system – a visualization and analysis tool for dynamic networks. It makes use of the multiple view paradigm: Several views are semantically combined by brushing and linking effects, e.g. selecting an object in one view highlights it in a different view. However, the goal of DGD to use it on standard size displays requires a less space-consuming seperation into multiple views (one could refer to the term "enhanced view" [4]).

1.2 Technical Preliminaries

When thinking of dynamic networks as a sequence of static networks their visualization is done by specifically crafted graph layout algorithms. These algorithms try to preserve the user's *mental map* of the network [5] – a fact that is cruicial for the understanding of structural changes. More technically such algorithms try to reduce the so-called *mental distance* between two consecutive layouts of a sequence [6]. One algorithm for this task is *Foresighted Graph Layout (*FLT*)* [7]. This layout strategy has already been examined regarding to its usefulness for dynamic graphs [8]. Furthermore the generic design of FLT allows the use of different layout paradigms [9].

However, the analysis of dynamic networks does not only require a sophisticated visualization technique. In many cases it is useful to have more information about the considered network. Such information can be obtained by computation of centrality measures for each actor as well as network properties. Obviously these information is not a single value but a sequence of different values. While dynamic network analysis tools like SIENA [10] compute detail statistical analyses about the evolution of a network DGD provides such results in its main window. This way the user is not only able to read exact values but also can visually compare values of different actors at different positions in the sequence.

2. COMPONENTS OF DGD

The main window of DGD consists of several views. After startup it presents the user an empty network visualization component that can be filled either by loading a file from disk or by modelling a net-

AVI'08 28-30 May, 2008, Napoli, Italy

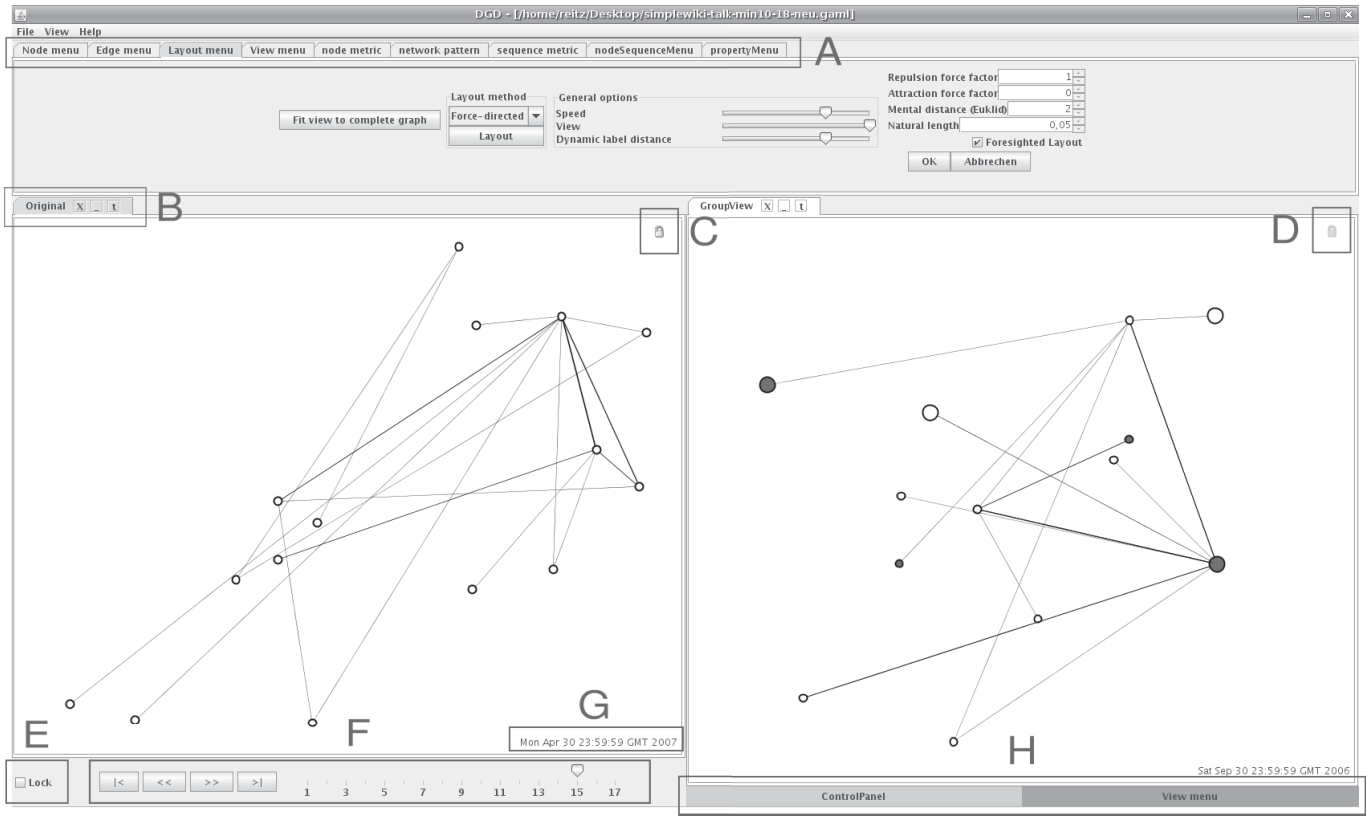

Figure 1: DGD at work: The menu panel (A) is visible at the top, the visualization components are arranged beneath the menu. Every visualization component has a title (B) and contains the sequence navigation controls (F). These controls become visible after moving the mouse on the "Control Panel"-area in the dynamic control section (H). After the mouse pointer exits the visualization component the controls disappear unless they are explicitly locked (E). The currently visible network is labelled at the bottom (G) and a manipulation lock (C) indicates whether the user is allowed to alter the network's structure.

work using the mouse. This component also provides the controls to navigate through the network sequence. In order to have several views onto the network DGD can display more than just one of this visualization components (see Fig. 1). This way it is possible to compare networks with different layouts or at different positions in the sequence.

Besides the network panel there is a menu panel that contains elements to modify the currently selected network component. Properties of nodes and edges can be changed as well as the used layout algorithm and its parameters. Furthermore this panel provides access to the network metrics.

The traditional menu bar of the main window is used for data import and export. DGD comes along with a XML-based file format that has been derived from GRAPHML [11] that provides no explicit support for dynamic graphs.

3. LAYOUT ALGORITHMS

The most important part in creating a view on a dynamic network, is the layout algorithm. In this context a dynamic network is considered to be an ordered sequence of static networks. It has to strike a balance between a high aesthetic quality and a good layout stability (this is usually referred to as *preservation of the mental map*). As mentioned before, DGD uses an implementation of the *Foresighted Graph Layout with Tolerance (*FLT*)* [7, 9]. The main idea behind FLT is to generate a global layout template for the sequence from the layout of the so-called *backbone*. Starting from this template FLT recommends to optimize the static layouts within tolerance limits by adjusting the induced layout.

The backbone contains only representatives of the semantically important nodes in the network sequence. More formally, for a given network sequence $g_1 = (V_1, E_1), \ldots, g_n = (V_n, E_n)$, a mapping $importance : V \rightarrow \mathbb{N}$ and a threshold δ the backbone contains the nodes $v \in \bigcup_{i=1}^{n} V_i$ with $importance(v) \geq \delta$ and the induced edges. The mapping importance defines a semantic on the set of nodes. By using the backbone DGD makes an individual definition of importance possible. By doing so users can modify the focus of layout stability.

The second phase of FLT concerns the iterative adjustment of the global layout template in order to obtain the final positioning in the layout of the static networks of the sequence. FLT bonds the degree of freedom for changes during this phase to a given threshold. FLT therefore needs *mental distances*, i.e. metrics for the preservation of the mental map [5] that are provided by DGD. Alltogether DGD currently contains implementations of three different layout paradigms. Each of them was redesigned to fit into the FLT context and hence fulfills the requirements for dynamic network visualization.

The **Force Directed Layouter** in DGD uses an implementation of the force-directed placement method by Fruchterman and Reingold [12]. The implementation uses a spring embedder with grav-

ity and simulated annealing. That way the algorithm distributes the nodes of a static network by computing pulling forces between connected nodes and pushing forces between every pair of nodes. The control of the different forces via the option's menu enables a visualization which highlights the clusters of the dynamic network. Thus DGD facilitates the identification of strongly connected components by a user.

The **Layer Based Layouter** extends the approach of Sugiyama [13]. A partition of the nodes will be evenly adjusted on different horizontal layers. The adjustment minimizes the number of backward edges, i.e. of edges $e = (v, w)$ with v on a higher and w a lower layer. Therefore, this layouter performs well for graphs with inherent hierarchic structure. Nevertheless this approach also can be used to layout arbitrary graphs so that users possibly detect an unknown hierarchic structure in the dynamic network. Since Sugiyama's approach contains two consecutive stages (layer assignment and layer ordering) that do not work iteratively FLT has been applied to these two stages. This results in the fact that the layer based layouter has controls for two mental distances. Depending on the user's interest in increased stability in layer assignment or in layer ordering he can select different values.

The **Orthogonal Layouter** is an extension of the approach of Brandes et al. [14] which is based on an orthogonal placement method by Fößmeier and Kaufmann [15], a so-called network-based approach. The implementation computes an orthogonal layout, i.e. edges are drawn as a sequence of hierarchical and vertical line segments. The discrete character of an orthogonal layout gives a good global overview of the layout of the static graphs. Evaluations show, that this approach allows a good layout for planar networks and allows users a fast detection of paths between connected nodes. Moreover, the recognition of clusters and shortest path will unfortunately be complicated.

4. VIEWS AND INTERACTION

In order to analyze networks it usually takes more than nice drawings. The possibility of filtering and manipulation and visible node centrality indicators are required. Furthermore characteristics of a network should be computable and visible within an analysis tool.

4.1 Available Views

To work with a dynamic network DGD provides the possibility to have several views to the same network. This can be useful when comparing two different layouts of the same snapshot or when comparing two different snapshots of the dynamic network. Besides the possibility of filtering data in each view seperately (see below) these views also support statistical network analysis by integrated pie diagrams (see Fig. 2).

This type of annotation has the advantage to present exact values to the user which can be crucial in some applications. However, this approach increases visual clutter and is not suitable for the visual exploration of a network. Thus, DGD also supports an integrated view of these metrics. The small pie diagrams show the node's normalized value concerning the selected metric. This approach makes it easier to find outliers in the network. This is an important feature especially in large networks where structures may appear similar although they are quite different.

4.2 Comparative Function Plot

While the integrated pie diagram allows for visual comparison of metrics for different nodes it is also desirable to have an impression of the values' time-dependent behavior. Therefore DGD provides a function plot view that shows the computed values for different

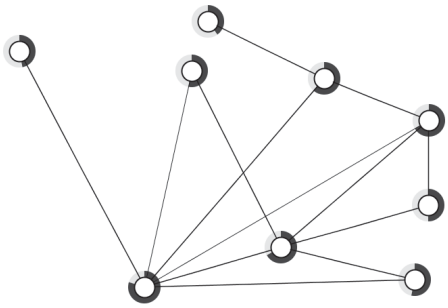

Figure 2: Integrated display of metric values using pie-charts.

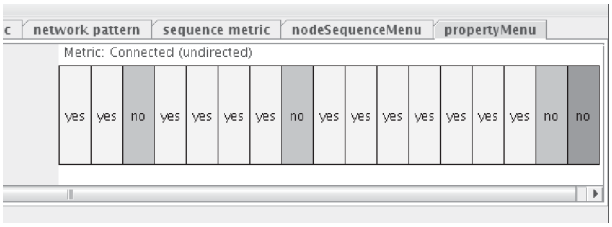

Figure 3: The view for binary properties in DGD. For every network the metric returns either true or false and the result is depicted in two different colors. The currently visible network is indicated by a darker color.

nodes over time (see Fig. 4). This view shows how selected nodes changed their kind and strength of relational integration. Furthermore nodes can be visually compared easily with each other and thus facilitate the identfication of dynamic roles in the network.

4.3 Binary Property View

The metrics and views presented so far work for any scalar value. However there are network properties that can either be *true* or *false* as whether the network is connected or is bipartite. For this binary metrics we implemented an additional view that contains boxes for each network and indicates negative answers with a red box and positive answers with a green box.

4.4 Filtering and Manipulation

DGD provides several mechanisms to work with the inspected data. Using the **GroupView** interface it is possible to alter visual objects. This can be done for each object seperately or by selecting groups of objects. Besides changing the color or the shape of an object it can also be hidden. Through the **ModeChange** interface a network can be semantically transformed. For instance it allows the conversion of a so-called two-mode network into a one-mode network.

5. RELATED WORK

The design of the DGD system was partially inspired by VISONE developped by Brandes and Wagner [3]. It offers many methods for network analysis and makes heavy use of an integrated visualization where computed values can be mapped on any visual property of nodes and edges. Due to its broad variety of available controls it tends to overload users. Other tools like PAJEK [1] and UCINET/NETDRAW [2] are primarily designed for computation of statistics.

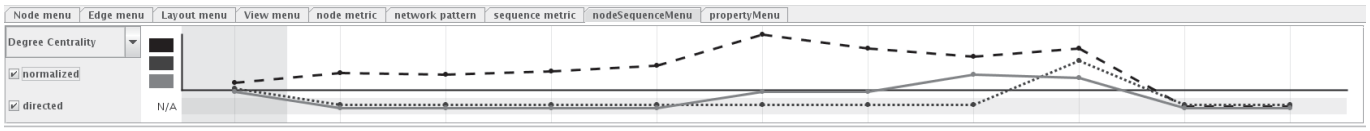

Figure 4: The (normalized) degree centrality values for the three selected nodes as a view in the menu panel. The plots are connected to the network view by using the same color. Authors 1 and 2 are not participating in the first months of the period. Their degree centrality is hence not available and indicated by values in the marked "N/A" area.

6. CONCLUSION

The dynamics of networks is another dimension of information that can be accessed using the DGD system. It incorporates several views onto the examined network as well as to the derived values from the network's structure according to specific metrics. The network visualization is done using a layout algorithm that is specifically crafted for dynamic networks as it pays attention to the user's mental map. The centrality measures can be integrated pie diagrams to facilitate the identification of interesting nodes or they can be displayed in a function plot view to enable visual comparison. The components of DGD are arranged such that it can be used on a usual-size display. This way DGD is a system for network analysis of dynamic networks that is easy to use and that works with standard hardware.

Acknowledgements

Thomas Söhngen implemented centrality measures. Mathias Pohl is partially supported by the Deutsche Forschungsgemeinschaft, grant no. DI 728/6-2.

7. REFERENCES

[1] Batagelj, V., Mrvar, A.: PAJEK – Program for Large Network Analysis. Connections **21** (1998) 47–57

[2] Borgatti, S., Everett, M.G., Freeman, L.C.: UCINet: Software for Social Network Analysis. Harvard MA: Analytic Technologies (2002)

[3] Brandes, U., Wagner, D.: Visone – Analysis and Visualization of Social Networks. In Jünger, M., Mutzel, P., eds.: Graph Drawing Software. Springer-Verlag (2003) 321–340

[4] Görg, C., Pohl, M., Qeli, E., Xu, K.: Visual Representations. In Kerren, A., Ebert, A., Meyer, J., eds.: Human-Centered Visualization Environments. Volume 4417 of Lecture Notes in Computer Science., Springer (2007) 163–230

[5] Misue, K., Eades, P., Lai, W., Sugiyama, K.: Layout Adjustment and the Mental Map. Journal of Visual Languages & Computing **6**(2) (1995) 183–210

[6] Bridgeman, S.S., Tamassia, R.: Difference Metrics for Interactive Orthogonal Graph Drawing Algorithms. In: Proc. of 6th Int. Symp. on Graph Drawing, GD. Volume 1547 of LNCS., Springer (1998) 57–71

[7] Diehl, S., Görg, C.: Graphs, They Are Changing. In: Proc. of 10th Int. Symp. on Graphdrawing, GD. Volume 2528 of LNCS., Springer (2002) 23–30

[8] Purchase, H.C., Hoggan, E., Görg, C.: How Important is the Mental Map. In: Proc. of 14th Int. Symp. on Graph Drawing, GD. Volume 4372 of LNCS., Springer (2006)

[9] Görg, C., Pohl, M., Birke, P., Diehl, S.: Dynamic Graph Drawing of Sequences of Orthogonal and Hierarchical Graphs. In: Proc. of 12th Int. Symp. on Graphdrawing, GD. Volume 3383 of LNCS., Springer (2004) 228–238

[10] Steglich, C., Snijders, T.A.B., West, P.: Applying SIENA. Methodology **2**(1) (2006) 48–56

[11] The GraphML File Format. `http://graphml.graphdrawing.org`, last visited Dec 20, 2007

[12] Fruchterman, T.M.J., Reingold, E.M.: Graph Drawing by Force-directed Placement. Softw., Pract. Exper. **21**(11) (1991) 1129–1164

[13] Sugiyama, K., Tagawa, S., Toda, M.: Methods for Visual Understanding of Hierarchical Systems. IEEE Transactions on System, Man and Cybernetics, SMC **11**(2) (1981) 109–125

[14] Brandes, U., Eiglsperger, M., Kaufmann, M., Wagner, D.: Sketch-Driven Orthogonal Graph Drawing. In: 10th Int. Symp. on Graph Drawing. Volume 2528 of Lecture Notes in Computer Science., Springer (2002) 1–11

[15] Fößmeier, U., Kaufmann, M.: Drawing high degree graphs with low bend numbers. In: 3rd Int. Symp. on Graph Drawing. Volume 1027 of Lecture Notes in Computer Science., Springer (1996) 254–266