INTERACTIVE VISUALIZATION OF HETEROGENEOUS SOCIAL NETWORKS USING GLYPHS

by

Merve Çaylı

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University

July 2009

INTERACTIVE VISUALIZATION OF HETEROGENEOUS SOCIAL NETWORKS USING GLYPHS

Merve Çaylı
CS, M.Sc. Thesis, 2009
Thesis Supervisor: Assist. Prof. Selim Balcısoy

Keywords: Information Visualization, Social Network Visualization, Heterogeneous Social Networks, Node-Link Diagrams, Glyphs

ABSTRACT

There is a growing need for visualizing heterogeneous social networks as new data sets become available. However, the existing visualization tools do not address the challenge of reading topological information introduced by heterogeneous node and link types. To resolve this issue, we introduce glyphs to node-link diagrams to conveniently represent the multivariate nature of heterogeneous node and link types. This provides the opportunity to visually reorganize topological information of the heterogeneous social networks without losing connectivity information. Moreover, a set of interaction techniques are provided to the analyst to give total control over the reorganization process. Finally, a case study is presented to using InfoVis 2008 data set to show the exploration process.

HETEROJEN SOSYAL AĞLARIN GLİFLER ARACILIĞIYLA ETKİLEŞİMLİ GÖRSELLEŞTİRİLMESİ

Merve Çaylı
CS, Yüksek Lisans Tezi, 2009
Tez Danışmanı: Yar. Doç. Selim Balcısoy

Anahtar Kelimeler: Veri Görselleştirme, Sosyal Ağ Görselleştirme, Heterojen Sosyal Ağlar, Düğüm-Bağ Diyagramları, Glifler

ÖZET

Yeni veri setlerine ulaşmanın son zamanlarda daha müsait olması nedeniyle, heterojen sosyal ağlarının görselleştirilmesine yönelik talep de artmaktadır. Mevcut görselleştirme araçları ise heterojen ağların görselleştirilmesinde ortaya çıkan çeşitli düğüm ve bağ tiplerinden oluşan ağın topolojisini okumanın kompleksleşmesi problemi için özel bir yöntem sunmamaktadır. Bu problemi çözebilmek amacıyla, heterojen ağların içerdiği çok değişkenli bilgiyi iletebilmek için, düğüm-bağ diyagramları içinde glifleri kullanarak yeni bir görselleştirme yöntemi sunuyoruz. Bu sayede topolojik bilgiyi bağllantı bilgisini kaybetmeden görsel olarak yeniden organize etme imkanı sağlamayı amaçlıyoruz. Bunun yanısıra, çeşitli etkileşim yöntemleri ile analiste yeniden organizasyon süreci için kontrol sağlıyoruz. Son olarak, InfoVis 2008 veri seti üzerinde yapılan örnek bir durum çalışmasının sonuçlarını sunuyoruz.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Selim Balcısoy for the time he has spent for this work.

I would like to thank CGLab members Tolga Eren, Ceren Kayalar, İsmail Kasarcı, Uraz C. Türker, Selçuk Sümengen, Billur Engin, Serdar Çakıcı, Berkay Kaya, Can Özmen, Çağatay Türkay and Kaan Yanç for all the fun. I would like to thank specially to the "InfoVis Team" for their co-operation and support from the beginning especially when things were not "that much fun".

I would like to thank my friends especially to Ayşen Dinçer, Ayşe Tosun, Nihan Camkıran, Fatma Uyar, Sezen Camkıran, Emrah Kavlak, Ayşe Gül Karatop, Ulvi Kasapoğlu, Nazlı Kaptan and Sedef Antay who have shown great patience and support till the end.

I would like to thank Cem Sina Çetin for his invaluable help and encouragement.

And i would like to thank my family for their love especially to Nurgül Çaylı for being such a perfect sister.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

Images are powerful means of communication. Through visual representations of data such as maps, simple charts, diagrams, it is easy and effective to convey information. However, getting a good grasp of large data sets requires a more systematic way of organizing and visualizing data. With the advancements in the last decades in computer hardware and computer graphics, the possibility of creating dynamic graphics opened a way for visual exploration of large data sets. With the assistance of a wide range of disciplines including pshychology, computer graphics, statistics and data mining, information visualization addresses the problem of revealing patterns, relations, changes, and structure within a large collection of data using visual representations.

Understanding social networks is one of the well studied topics of information visualization. Social network analysis, as a seperate discipline itself, precedes information visualization. It integrates a large body of methods from statistics and graph theory, yet creating visualizations of social networks is crucial for discovering unexpected patterns.

Due to the relative availability of data gathering, more recent social network data is usually heterogeneous i.e. it contains more than one type of *actors* and more than one type of *relations* and attributes associated with actors. Most of the existing visualization methods assume a simple network with one type of actor and one type of relation. Thus, networks with heterogeneous link types and node types cannot be efficiently handled. As pointed out by Chen, “a smooth transformation between topologies on different types is a challenge perceptually, cognitively, and algorithmically” [20]

1.1 Visualization of Heterogeneous Social Networks

Methods for representing network data are node-link diagrams and matrix-based representations. There are various different visualization tools based on these two methods. Although there are few examples with the specific purpose of visualizing heterogeneous network data, none of them propose a method to handle the cognitive load introduced by the task of following connectivity information between different node types.

In this thesis, novel graphical representations based on node-link diagrams are introduced to:

- visually reorganize topological information of the heterogeneous network without losing connectivity information
- help identification of weak ties
- allow investigation of potential relations between attributes and the observed pattern of the network
- visually encode graph theoretic metrics in order to identify the characteristics of the network.

Our idea is based on aggreagating adjacent nodes into super nodes and representing a super node by a *glyph*. The glyph is especially designed to allow organizing and representing the multivariate nature of the heterogoneous node and link types. In order to conveniently convey the aggregated information embedded in the super nodes, the glyph itself is designed as an aggregate object, formed by merging subunits together where each subunit represents an underlying node. (Figure 1) By forming the glyph out of subunits rather than designing it as a unitary object, the connectivity information is not lost and it is possible to represent each possible combination of nodes from different or the same node types. Moreover, the visual parameters of the glyph are used to encode type-specific information.

Figure 1.1. A glyph node representing 3 nodes belonging to 2 different node types.

An example node-link diagram using glyphs is shown in Figure 1.2. In this figure, different combinations of node types are used to construct the glyphs, hence allowing to focus on different node or relation types at each time.The details of the data set that is being visualized in this figure will be given in Section 7.

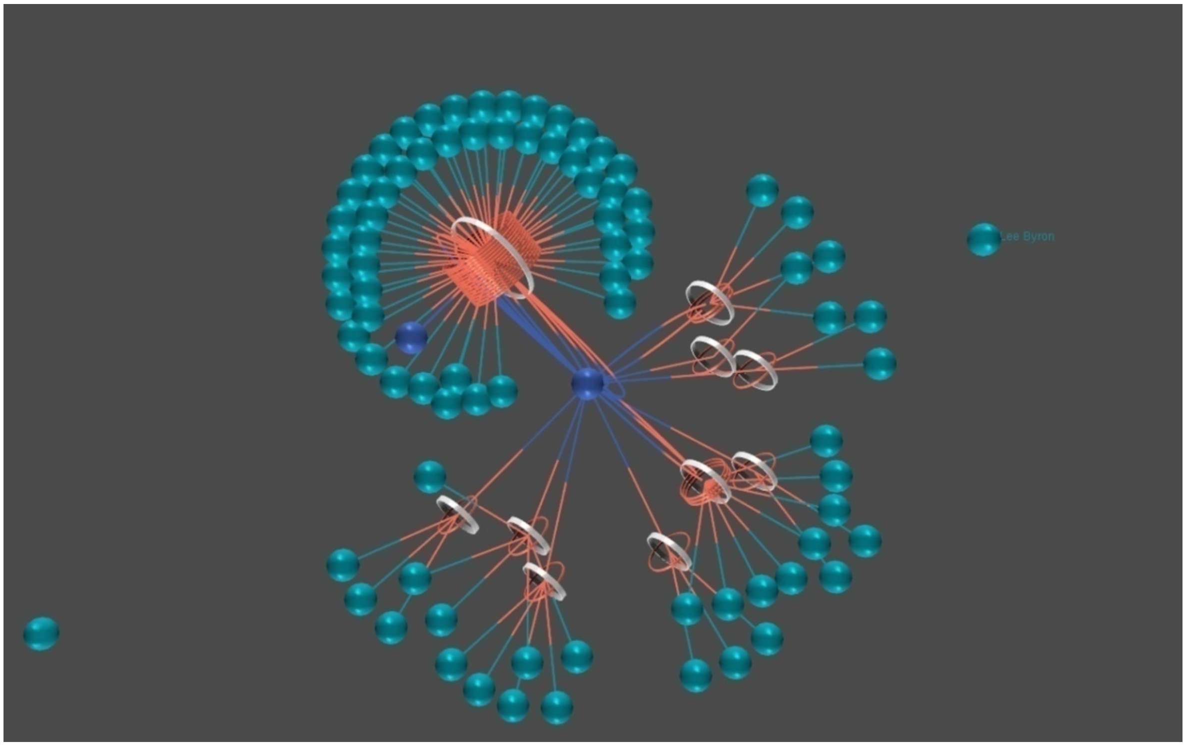


Figure 1.2. Visualization of co-authorship data set of InfoVis 2008.

1.2 Thesis Outline

In this chapter, the goal of information visualization is defined and the complementary role of information visualization for understanding social networks is explained. The need for novel representation methods beyond standard ones for heterogeneous networks is addressed. Finally, the proposed approach for this problem is introduced.

In the second chapter, Background and Related Work, social network analysis tasks are defined, how these tasks are tackled in information visualization is explained. A brief analysis of existing tools is given and whether they present an exploratory analysis and handle heterogeneous networks or not are discussed.Among the methods for social network visualization, graph drawing algorithms are specifically addressed. Force-directed algorithms for graph drawing ,hence for node-link diagrams are summarized.

Finally, in search of new graphical items for node-link diagrams, the use of glyphs in information visualization is introduced. Glyph design details and examples of the use of glyphs in information visualization are given.

The third chapter describes the proposed visualization method in detail.

The fourth chapter presentts the results of a case study.

And the final chapter discusses the successes and the drawbacks of the work presented and possible improvements to the initial model as future work.

2 Background and Related Work

Researchers suggest that progress in various fields is achieved thanks to the use of visual representations. [1]. This statement also holds for social network analysis because visual representations have played a key role in the field since the beginning.

In this chapter, in order to convey information about the contribution of information visualization to social network analysis, the analysis tasks are summarized with a brief introduction of terms and concepts. Then the methods to visualize network data and selected examples of social network visualization are given. The shortcomings of these methods and examples for heterogeneous social networks are investigated. Finally, glyphs are examined as candidates for representing heterogeneous network data with examples of the use of glyphs in information visualization.

2.1 Social Network Terms and Concepts

A social network is a structure made of social entities that are interacting with each other. The social entities are generally referred as *actors*. These actors are usually individuals, but may also be corporate or collective units. Examples of actors are people in a group, departments within a corporation, public service agencies in a city, or nation-states in the world system.[6] The type of an actor is also referred as a mode. If the network consists of actors that belong to the same type, it is called a *one-mode network*.

If two actors are linked to each other in a network, then it is called a *tie*. The type of the tie is referred as a *relation*. There is a wide range of relations that can exist in a social network. Common examples of relation types are evaluation of one person by another(for example expressed friendship, liking, or respect), transfers of material resources (for example business transactions, lending or borrowing things), association or affiliation (for example jointly attending a social event, or belonging to the same social club), behavioral interaction (talking together, sending messages), movement between places or statuses (migration, social or physical mobility), physical connection (a road, river or bridge connecting two points), formal relations (authority), biological relationship (kinship or descent)[6].

A social network data set may have more than one type of actors and more than one type of relations. And there may also be attributes associated to actors and relation types. Such a network is called *multimodal* and *multirelational*. They are also referred as *heterogeneous social networks*. Since, graphs and networks are two closely related concepts, instead of actor or mode, *node type* is also used. In this thesis, these terms will be used interchangeably.

2.2 The Tasks in Social Network Analysis

Wasserman and Faust define the task in social network analysis as: **“**to understand properties of the social (economical or political) structural environment and how these structural properties influence observed characteristics and associations among characteristics. “

The main focus of analysis is on relationships among actors and the patterns of these relationships. This is also referred as the *structure* of the social network. The structural properties can be identified by determining specific characteristics of the social network.

Like many other social network analysis applications, as in [30], the major characteristics to explore the structure of the network are defined as:

- Identifying central actors
- Identifying communities
- Analyzing roles and positions

Social network analysis heavily relies on graph theoretic measures. The usual mathematical description of a graph comprises of vertices and edges which suffices to model one-mode networks. However, heterogeneous networks require a different mathematical abstraction. A *semantic graph* which contains different node types (different types of vertices) and different edge types can be used for this purpose [5]. It is also called *attributed relational graph,* and *relational data graph.* A semantic graph is annotated with an *ontology* also known as a *schema* from relational database theory.

In the following subsections, the explanations of central actors and communities will be given and graph theoretic measures will be introduced to achieve these tasks both using a standard graph for one-mode social networks and using a semantic graph for heterogeneous social networks. Afterwards, analyzing roles and positions will be covered and the models for employing such an analysis are summarized.

2.1.1 Identifying central actors:

In a given social network, some actors are relatively linked to more actors than others, or they may be linking communities. Such actors are said to be “central” in a social network. They can easily be identified by using metrics from graph theory such as node degree centrality and betweenness centrality.

Betweenness centrality for a vertex v is the number of shortest paths that pass through v. Degree centrality is simply the number of links that a node has.

For semantic graphs, and for heterogeneous networks, a given value k of the connectivity of a node type α has no real meaning. For instance, in Figure 3, the topological connectivity in both cases is $k = 4$ but the meaning of it is very different in each case. Another source of confusion is the fact that the number of β-type nodes can be very large thus inducing a bias in the node degree metric. [5]. Therefore, these metrics should be redefined to accommodate different node types and relation types.

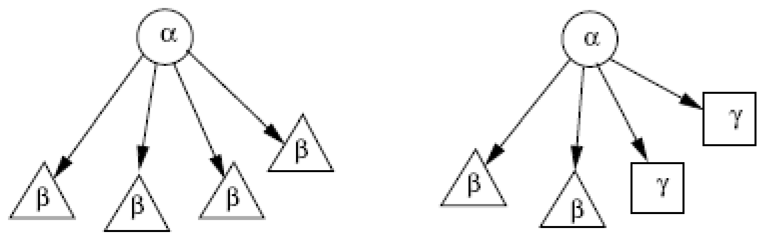


Figure 2.1. Examples of topological connectivity with the same values for type α, but with a different meaning in each case. [5].

In the ontology, each node type α can be connected to a certain number of other types denoted with k_α^0 of other types. The total number of nodes in the semantic graphs is $\sum_\alpha n_\alpha$, and the nodes are denoted by $i = 1, \dots, n.$ The type of a node is defined with the function $t(i)$. And $k_{\alpha\beta}(i)$ is the number of neighbors of type β of a node i of type α. The usual connectivity of the node i (which is of type α) is given by:

$$k_\alpha(i) = \sum_\beta k_{\alpha\beta}(i) \tag{1}$$

Based on this formulation, the average connectivity of type α is the average over all nodes with type α:

$$\overline{k_\alpha} = \frac{1}{n_\alpha} \sum_{i, t(i)=\alpha} k_\alpha(i) \tag{2}$$

In these formulations, it is crucial that some types are connected to many other types while some types are only connected to one type. Hence, rescaling the average connectivity by the different number of neighbor types, we get average number of neighbors per type.

$$m_\alpha = \frac{\overline{k_\alpha}}{k_\alpha^0} \tag{3}$$

2.1.2 Identifying communities:

Social networks are known to have a certain characteristic which is called the small world property. In the general sense, this means a large proportion of highly interconnected nodes and a small proportion of intercommunity edges with respect to the edge distribution throughout the network.[5] Due to this property, communities are formed naturally in the social network. It is possible to identify communities using standard representations such as node-link diagrams.

However, in heterogeneous networks, since there is more information present other than connectivity between same node types, the analyst may want to gain insight about the shared characteristics among the community members. For instance, they may all have the same or very close value of an attribute, or their connection patterns with another node type may be similar. Hence, community analysis in heterogeneous social networks not only involves identifying interconnected nodes, but also describing the characteristics of the members in a community.

2.1.3 Analyzing roles and positions:

Social position and social role are two concepts that translate into analyzing actors' structural similiarities and patterns of relations in multirelational networks. Although such an analysis is employed in any kind of social networks, it is more interesting in multimodal and multirelational networks [5, 6].

Position is a concept that is based on the ties among subsets of actors. On the other hand, role is defined in terms of collections and the associations among relations and refers to the patterns of relations which obtain between actors and between positions [6].

2.1.3.1 Positional analysis:

The major objective of positional analysis is simplification of network data by grouping actors in *positions* based on structural similarity. To assess structural similarity *structural equivalence* is measured, i.e. two actors are structurally equivalent if their ties are identical, in other words, they make ties with the same exact actors. Due to the rareness of observing this case, this condition can be relaxed, so that similar actors in terms of ties are partitioned in the same position. The partitioning part can be handled by clustering algorithms for discrete and by multidimensional scaling for continuous models.

To model positional structure of the network, a method that has been devised is blockmodeling. It was introduced by White, Boorman and Breiger in 1976. For blockmodeling, first actors are partitioned into subsets called positions, then for each pair of positions, a presence or absence of a tie is stated. Presence or absence of a tie among positions is determined using some criterion. The blockmodel can then be interpreted by investigating an indication between the attribute values and the structural form presented in the blockmodel, and possible patterns in the overall model. [6]

2.1.3.2 Role Analysis:

The problem tackled in role analysis is identifying patterns or regularities in relations among positions. Role analysis can be done at different levels such as global role structures, local roles, individual or ego rules. The relations that need to be examined at all levels of analysis are chains of connections among people. Hence compound relations as well as primitive relations are required to be investigated.

To represent positions and roles, density matrices, image matrices and reduced graphs are used. In a reduced graph(Figure 2.2), the nodes represent positons and the ties between nodes represent the ties between positions.

Figure 2.2 Example simplifying a network using structural equivalence. The graph on the left is the original graph and the graph on the right is its corresponding reduced graph. Cf. [6].

In addition to these tasks listed above, there are many other questions the analysts seek to answer while doing social network analysis. For instance, identifying the outliers in the data which are also known as weak ties are also crucial in social network analysis. [14]
The questions asked may vary due to the nature of the social network data.

2.2 Information Visualization Perspective for Social Network Analysis

Visual presentation is known to be an effective way of conveying information. As pointed by Schneiderman, “the bandwidth of information presentation is potentially higher in the visual domain than for media reaching any of other senses.”[22] In other words, as put by Tufte, “graphics reveal data.” Indeed graphics can be more precise and revealing than conventional statistical computations. “[31]

In accordance with these notions, information visualization has proved to be a complementary research field for data mining and statistical methods. [23] The motivation for creating visualizations is to reveal otherwise unexpected patterns, and to develop a well-understood information for needed items in a large collection of data.

The basic principles for visual design are summarized as the Visual Information Seeking Mantra: Overview first, zoom and filter, then details-on-demand. [27]The tasks for creating a succesful visualization are a systematic way of choosing which graphical primitives and methods to use for representation, how to map visual parameters to different data variables, and based on the user demands, how to reorganize the visualization.
Social networks is one of these data domains that have been widely studied in information visualization. The inherent tasks for information visualization all apply well to social network visualization.

There are two representation alternatives for social network visualization. One is node-link diagrams and the other is matrix-based representations. (Figure 2.3) Although matrix based representations outperform node-link diagrams in dense networks, a large proportion of the visualizations are based on node-link digrams.

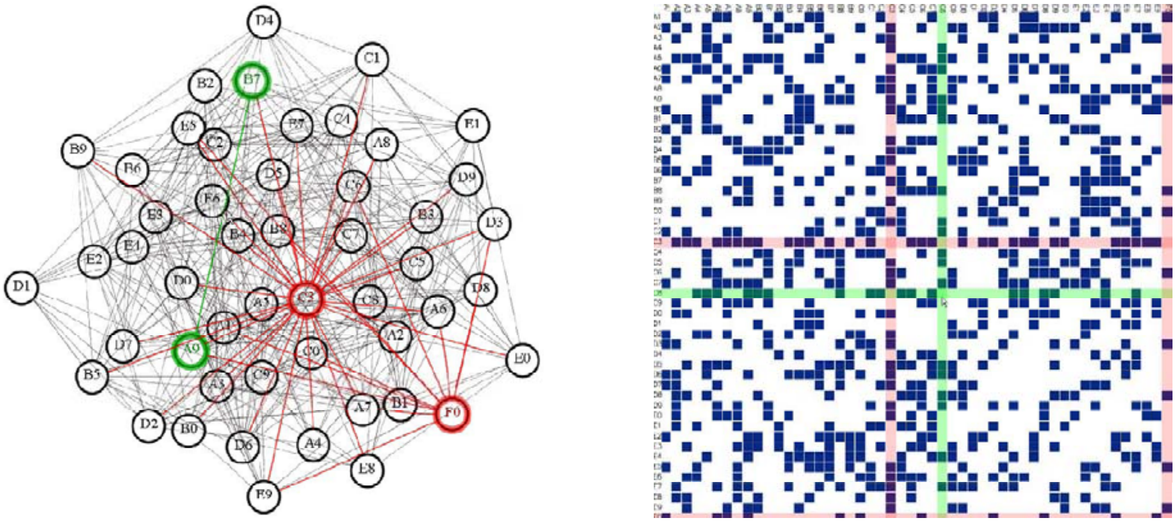


Figure 2.3. Two visualizations of the same undirected graph containing 50 vertices and 400 edges presented in [11].

As noted in the previous sections, the most suitable mathematical model for a network is the graph. Thus, node-link diagrams are good and natural candidates for representing social networks. Therefore, the representation problem in social network visualization is resolved to a large extent by the well-established literature on graph drawing and graph visualization.

2.2.1 Graph drawing in information visualization

Graph drawing algorithms and graph visualization play a fundamental role in information visualization. [20]. Graph drawing is also an active research field beyond information visualization due to the wide range of applicability of graphs. The problem that graph drawing algorithms is dealing with is mainly the graph layout operation. This simply translates into assigning a position to each vertex and a curve to each edge. Hence the goal of graph drawing algorithms is to find a geometrical configuration such that the arrangement is easy to understand, and fits into the viewing area. [32]

The layout of the graph is crucial in the sense that, it may help conveying the key features of a complex data, whereas it mayalso undermine presenting the nature of the underlying structure. [20] For an assessment of the goodness of the layout algorithm, there is an acclaimed list of requirements prepared by Battista[26].

Selected requirements from this list includes:

Readability: As the name suggests, the information contained in the graph should be easy to read. This criteria entails the follwin conditions:

- The drawing should minimize the number of edge crossings
- The drawing should maximize the angle between adjacent or crossing edges
- The drawing should exhibit structural properties of the graph which are of particular interest to the user (for example, symmetries in the graph)
- Vertices should be distributed uniformly in the viewing area with adequate *vertex resolution, defined* as the minimum distance between vertices in the drawing. Pairs of adjacent vertices should not be drawn close to each other.

Conformance: The conformance criteria is the set of the rules defining the guidelines for drawing to conform to the conventions of the application area which have been historically developed.

*Efficiency:*This criteria is a restriction on time for interactive use. This is typically two seconds. [27]

2.2.2 Force - directed layouts

The force-directed layouts are also referred as spring embedder or mass spring models in many contexts. In this section, these terms will be used interchangeably.

Among the wide variety of graph drawing algorithms, the spring embedder model is one of the most popular algorithms for drawing undirected graphs. It has also drawn much attention in information visualization community due to its simplicity and aesthetically pleasing results.

The spring embedder model was originally proposed by Eades [9]. This algorithm satisfies two conditions of aesthetic criteria: uniform edge lengths and symmetry. Later on, this initial algorithm has been modified and extending incorporating other criteria:

- The number of edge crossings should be minimal
- The vertices and edges are distributed uniformly.

As the name of the model suggests, an analogy is drawn between the layout problem and the behavior of some idealized physical system for the sake of easiness of simulating it computationally. The physical system comprises of a mechanical system of frictionless hinges connected by springs. A hinge corresponds to a vertex in the graph and a spring corresponds to an edge between two vertices. These springs, of specified natural or unstressed length, are referred as type I. There are additional springs placed between vertices that are not connected to each other and these are referred as type II.

The goal of the spring model is to find an arrangement of vertices so that the mechanical system is at equilibrium. Such an arrangement is the desired layout of the graph.

At equilibrium, the magnitude of the net force acting on each vertex is equal to zero. This condition can also be characterized in terms of potential energy, i.e. the potential energy stored in the springs is stationary. In general, there are more than one equilibrium configurations, thus, the problem is of search. Numerical methods can be devised to find such configurations.

Initially, the vertices are randomly positioned in space. Then, the system is relaxed through equilibrium incrementally. At every iteration, the net force acting on each vertex is used to calculate the new position of the vertex. This process can be formulated by ordinary differential equations which can be in effect solved numerically.

The form of the force law has an effect on improving the final layout. Instead of the ideal linear spring law which is too strong in the attraction region, Eades[9] has proposed a logarithmic law. Another modification by [39] is a rational polynomial law for reducing computational cost.

$$f(s) = \begin{cases} k_c(L - s) \\ k_c \ln(\frac{L}{s}) \\ k_c(\frac{L^2}{s} - \frac{s^2}{L}) \end{cases} \quad (6)$$

where s is the spring length, L is the unstrained spring length, $f(s)$ is the magnitude of the force developed when the spring has length s, and k_c is a constant.

A repulsive force between vertices that are not directly connected, referred as type II springs in the mechanical model, is formulated with an inverse square law:

$$f(s) = k_u(\frac{L}{s})^2 \tag{7}$$

A form of energy dissipation model for this mechanical system is viscous damping.(Figure 2.4) It is formulated by providing a restraining force proportional to velocity and acting in the opposite direction. Applying Newton's second law to this model, the basic system of second order ODE's completely describing the motion of the vertices is acquired:

$$m\frac{d^2 r_k}{dt^2} = m\ddot{r}_k = F_k \qquad k = 1, \ldots., N. \tag{8}$$

where N is the number of vertices in the graph being modeled, m is the mass of each hinge, $r_k$is the position vector of vertex v_k, t is time, and F_k is the total force acting on vertex v_k. Assuming ideal viscous damping at each vertex, equation(8) becomes:

$$m\ddot{r}_k = F_k^s(r_1, \ldots, r_N) - D\dot{r}_k \quad k = 1, \ldots., N. \tag{9}$$

where $F_k^s(r_1, \ldots, r_N)$ is the total force acting on vertex v_k due to the srings and constraints, $\dot{r}_k$ is the velocity of the hinge corresponding to vertex v_k, and D is a constant defining the strength of viscous damping.

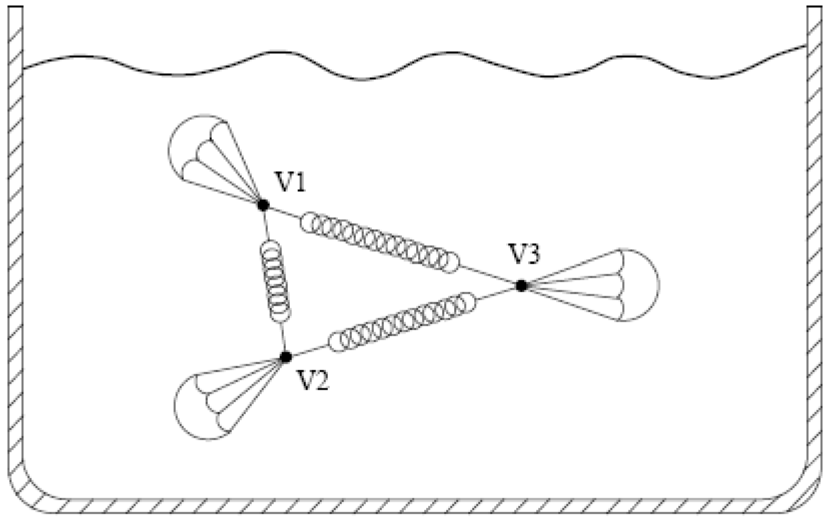


Figure 2.4. Graph layout modeled as a damped spring system.[32]

At an equilibrium, all the vertices are at rest balancing all the forces, so the equilibrium configurations are the roots of the nonlinear system of equations:

$$F_k^s(r_1, \dots, r_N) \qquad\qquad k = 1, \dots, N. \qquad (10)$$

Since it is not feasible the find the roots directly from these equations directly, a numerical solution can be devised to walk the system to equilibrium from initial configuration.

2.3 Social Network Visualization Tools

In this section social network visualization tools and prototypes will be presented under two categories: menu based systems and exploration systems in compliance with the classification done by [30].

Menu-based systems are powerful social network analysis tools containing the most sophisticated statistical and graph theoretic algorithms, however the major drawback of these tools is that the users are required to be experts in the field in order to efficiently make use of these tools. On the contrary, exploration systems put more effort in the visualization and interaction part, making it possible for non-experts to use them hence favoring an exploratory experience.

2.3.1 Menu-Based Systems:

Among the most developed menu-based systems Pajek[2]works with a statistical program and provides a large set of algorithms to partition, permute, hierarchize and layout networks. [13]. Pajek supports bimodal (consisting of two different modes) and multirelational networks. However, both nodes and edges may have only single values (associating multiple attributes is not supported) [33]. Another drawback of Pajek is that it produces static images, which makes it almost impossible for exploration.

Like Pajek, UCINet also supports uni-modal and bi-modal networks and contain a large set of graph mining and network operations. However, it doesn't contain visualization procedures. The results can be visualized only if they are exported to a tool like Pajek. This limits interactive visual exploration. [33, 4]

NetMiner is another tool that accomplishes network analysis operations. And unlike UCINet, it can also present the results visually. Multiple attributes can be associated to each node, but it only supports unimodal and multirelational networks. An example visualization for a one-mode network is given in Figure 2.5.

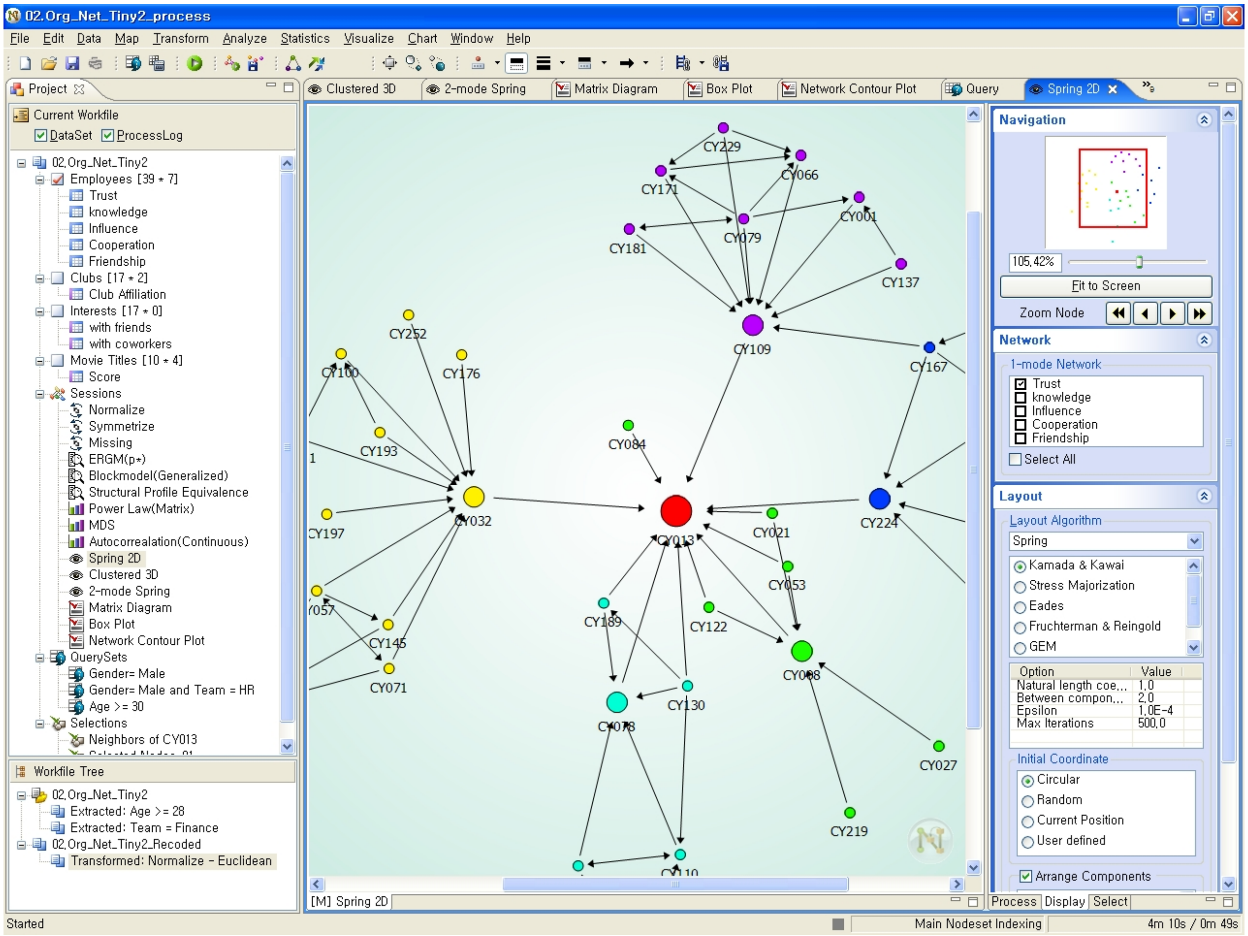


Figure 2.5. An example visualization created with NetMiner, actors are employees in an organization.[40]

Under menu-based systems, there are also some graph toolkits that serve as frameworks for building social network analysis tools. These are JUNG, GUESS and Prefuse. JUNG supports variety of representations of entities and their relations, including directed and undirected graphs and multimodal graphs. It contains algorithms from graph theory, exploratory data analysis, social network analysis and machine learning [34]. These algorithms contain routines for clustering and blockmodeling. For instance, in Figure 2.6, structurally equivalent vertices have been combined defining positions.

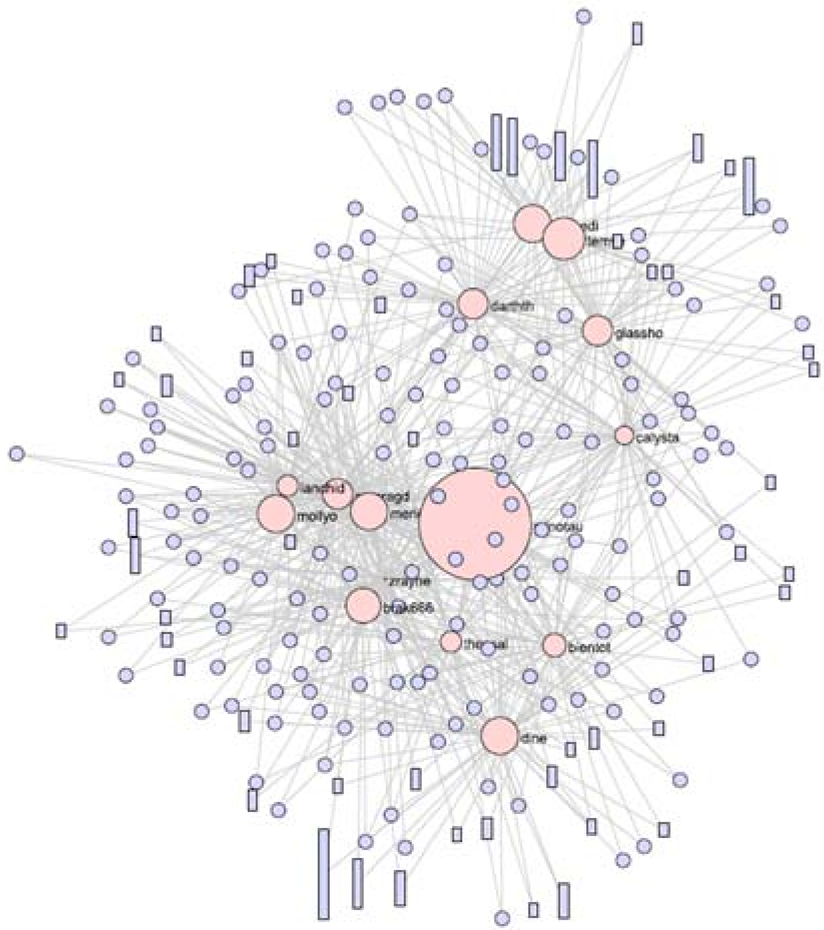


Figure 2.6 A network for which structurally equivalent vertices have been visually comined into vertical blocks, where height represents the number of vertices in a block.[34]

Prefuse is a framework for creating dynamic visualizations. The analysis tools it contains are limited compared to the other tools under this category, but it offers a wide range of interactive visualizations for different data domains. There are numerous visualization tools developed using Prefuse up-to-date [25, 33].

Finally, Guess is a language and interface for graph exploration that combines visualization and analysis into one package. Unlike Prefuse and JUNG, it has a high level language which lets users control visualization and prototype visualizations. The user also has the option of associating attributes with nodes and edges [3].

2.3.2 Exploration systems:

Among the numerous examples in social network visualization tools that can be attributed to this category, the examples are carefully selected according to the network type they are representing. All the selected examples are either visualizations of heterogenous social networks or presenting a method to represent attributes with one-mode network data.

In [17], a method for visualizing multivariate graphs where the nodes have several discrete categorical dimensions is presented. The goal is to allow making cross variable comparisons for analysts who wish to see association between two variables immediately. – e.g. how race affects patterns of communication between genders. The analogy of visualizaiton is based on OLAP and two key reporting operations: roll up and selection. The result of a roll-up operation is desribed in Figuew 2.7:

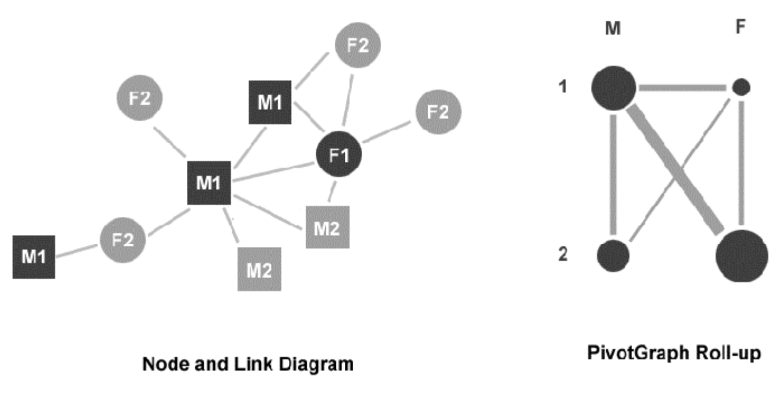


Figure 2.7. Same data visualized with node and link diagram(left) and pivotgraph(right) [17].

The PivotGraph diagram makes it clear that there are connections between all gender/division pairs, with the exception of men and women in division 2. Major drawback is information loss – the graph topology is not preserved under roll-up and selection operations.

In [30], dense subgraphs are simply aggregated and displayed as matrices. Attributes of the underlying nodes and underlying links are combined and propogated up to the aggregated elements. The visual variables of the matrices, i.e. the background color of a matrix can correspond to an aggregated node attribute, while attributes of each underlying node can be shown along the axes (the sides) of the matrix. Another nice feature of nodeTrix is the set of interactions designed for exploration process. The users can edit the network by aggregating a group of nodes into a matrix and vice versa, and by moving the positions of nodes and matrices arbitrarily. In figure 2.8 , a part of the infovis co-authorship network visualization is shown.

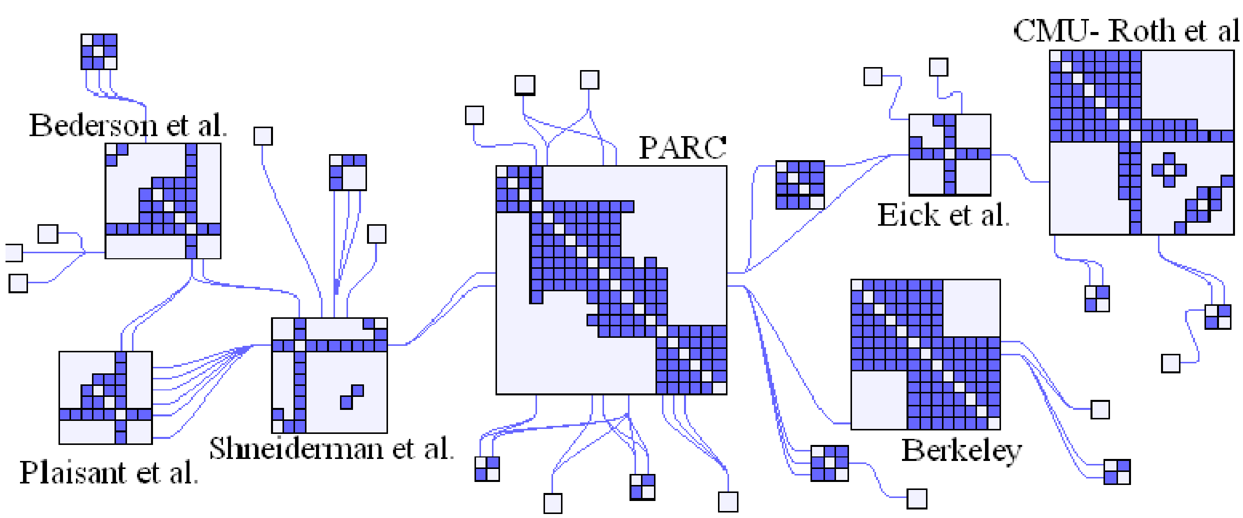


Figure 2.8.. NodeTrix representation of the largest component of the InfoVis Co-authorship Network [30].

OntoVis [21] is a visual analytics tool specifically designed for understanding large heterogenous social networks. The data level abstraction is done by annotating the underlying graph with an ontology and visualization is guided through this ontology. Users can add or remove node types to and from this ontology. Moreover, importance filtering is done using some metrics in order to minimize the visual complexity. The results are presented in a case study using two data sets: terrorism knowledge base and imdb. The figure 2.9 shows an example visualization of the terrorism network.

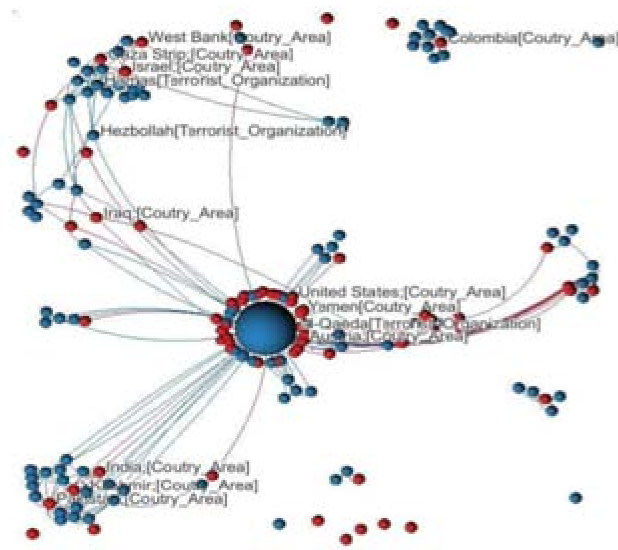


Figure 2.9. Visualization of Terrorist Organizations(in blue) and Locations(in red) created by OntoVis.[21]

Like OntoVis, Invenio [33] is also a visual analytics tool specifically designed for multimodal, multirelational and multiattributed social networks (referred as M*3 social networks). It integrates visualization options provided by Prefuse with graph mining algorithms from JUNG. The major contribution of this work is the support for relational operations which is very useful for positional and role analysis in multirelational networks.

2.4 Glyphs in Information Visualization

The dictionary definition of glyph is "a symbolic figure or character, usually a picture that gives information". [41] The word glyph is referred in many different contexts, with varying graphical representations yet they are all used to represent some kind of information. Usually, there is more than one visual parameter that belongs to a glyph. Each of these parameters can be utilized to encode a certain data dimension, therefore in the field of information visualization, a glyph is a good candidate for multivariate data visualization.

One of the earliest examples of glyphs is Chernoff face developed by Chernoff in 1973 [41]. The features of the human face such as position of the eyes, length of the nose or form of the mouth are all used to encode separate data variables. (Figure 2.10)

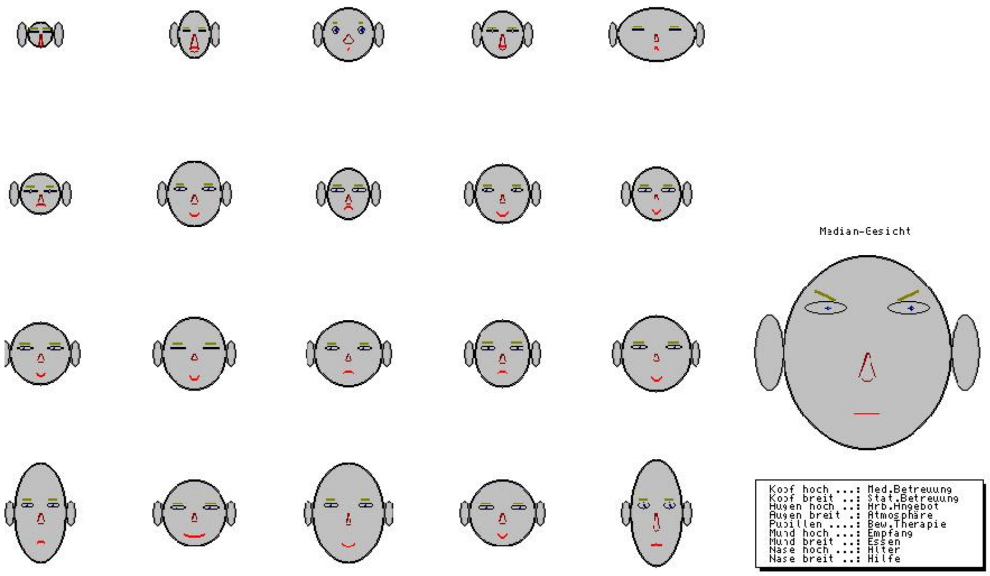


Figure 2.10. Conteneness of patients of a hospital. [44]

Another simple and early example is stick figures. [43]. Lines are used as units of representation and from and orientation of the lines are used to represent data dimensions. Additionally, color, width and height etc. can also be used.

In information visualization literature, where glyphs are used for multivariate data visualization, there are numerous examples of glyph designs. One group of early designs are for multidimensional discrete data. In the whisker plot, each data value is represented by a line segment radiating out from a central point. The length of the line segment denotes the value of the corresponding data attribute. [24]

A variation of the whisker plot is the star plot. The only variation is the connected ends of the lines. Another related plot is Exvis tool. Data values are mapped to visual parameters of the stick figures. [24] These three plots are shown in Figure 2.11 consecutively.

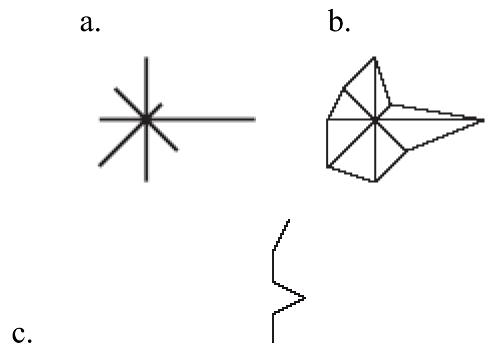


Figure 2.11. Three glyph designs. (a) The whisker or fan plot. (b) A star plot. (c) An Exvis stick icon.[24]

The key issue in glyph design is how the multiple data fields in a data set are mapped to the parameters of the glyph. This issue raises questions about the discernability of the gyphs. The mapping operation is more efficient if it is done in a systematic way based on the theory of integral and seperable dimensions. This concept is related to situations in which one display attribute will be perceived independently from another. Integral display dimensions are two or more attributes of a visual object that are perceived holistically. i.e. A rectangular shape perceived as a holistic combination of the rectangle's width and height. On the other hand, separable display dimensions are perceived separately.

The perceptual studies on glyph design suggest that glyph element orientations should be separated by at least 30 degrees. In [24], Ware suggests that at most three different orientations must be used for "really rapid classification of glyphs." The findings on other dimensions by Ware is presented in Table 1.

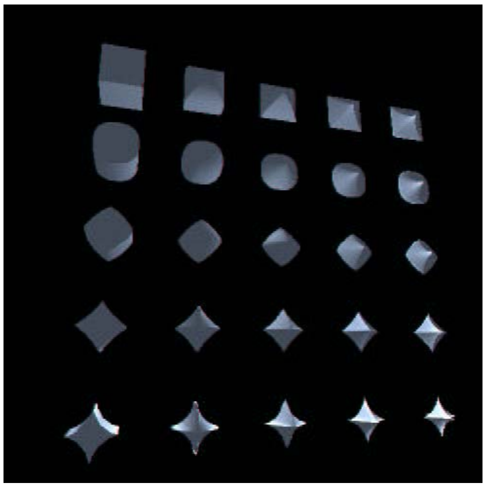
Visual Variable	Dimensionality	Comment
Spatial position of glyph	1 Dimensions: X, Y, Z	
Color of glyph	2 Dimensions:Defined by color opponent theory	Luminance contrast is needed to specify all other graphical attributes.
Shape	2-3? Dimensions unknown	The dimensions of shape that can be rapidly processed are unknown. However evidence suggests that size and degree of elongation are two primary ones.
Orientation	3 Dimensions: for each axes	Orientation is not independent of shape. One object can have rotation symmetry with another.
Surface Texture	3 Dimensions: orientation, size and contrast	Not independent of shape or orientation. Uses up one color dimension.
Motion Coding	2-3? Dimensions: largely unknown, but phase may be useful	
Blink Coding	1 Dimension	Motion and blink coding are highly interdependent.

Table 1. Dimensionality of visual variables.

There are numerous examples of glyphs in information visualization literature . Selected examples will be given under two different categories:

- Works that are focusing on methods to automatically generate glyphs
- Visualization examples using glyphs as the representation method

In [35], procedural techniques for automatically generating glyph design for visualizing quantitative, continuous data are presented. (Figure 2.12) This method helps abstracting the burden of glyph design from the user with procedural techniques, and also allowing a high-level control over the shape design.



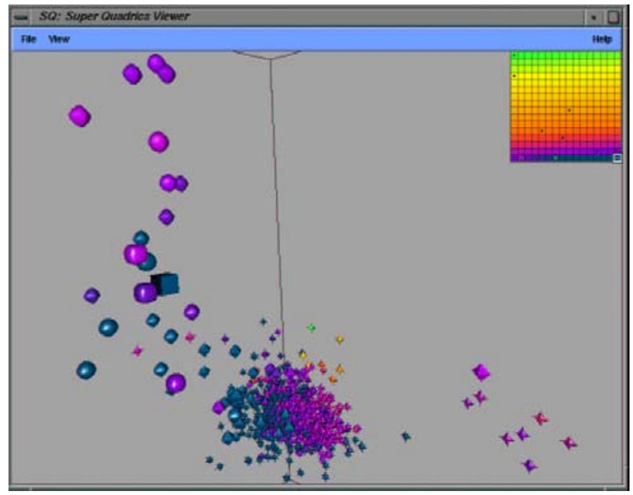


Figure 2.12. Three dimensional visualization of 1833 documents' relationship to gold prices, foreign exchange, the federal reserve, stock prices, and Manuel Noriega presented in [35].

Two similar works are [36] and [37]. In the first example, a system that enables non-programmers to define almost arbitrarily complex glyphs is presented. (Figure 2.13) And in the second example, the number of data dimensions that are simultaneously viewable are increased from 8 to 14 attributes. (Figure 2.14)

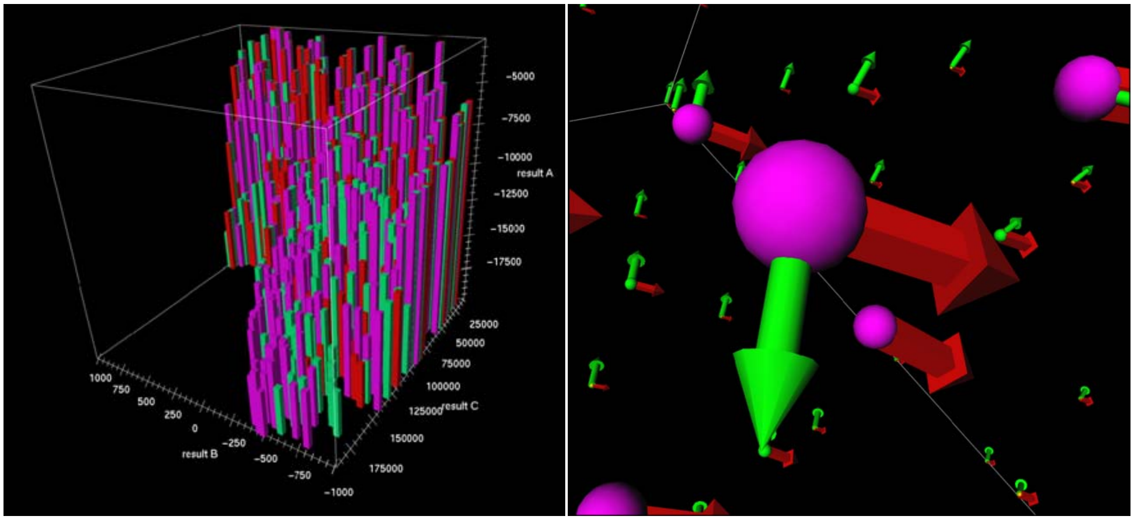


Figure 2.13 A scatterplot using bar glyphs(left) and an example of composite glyphs(right).[36]

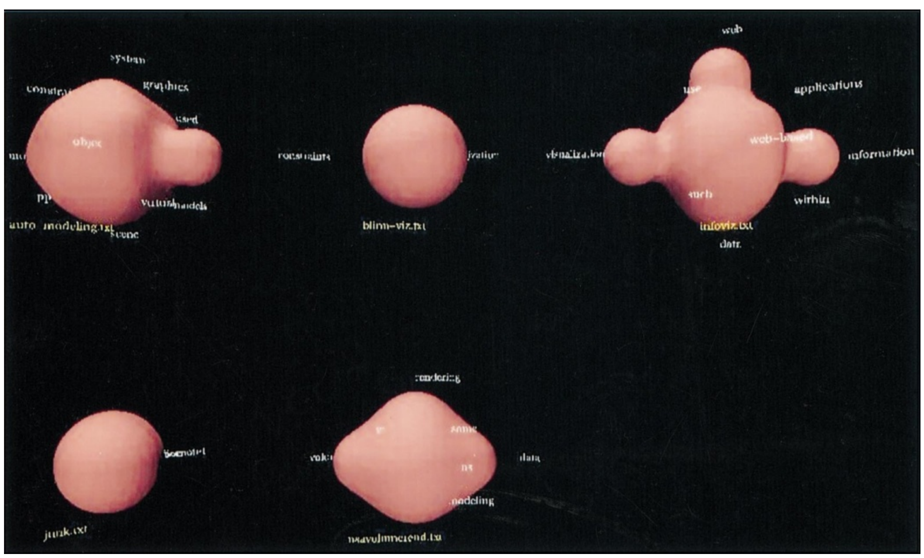


Figure 2.14. Multiple documents term frequency visualized as implicit surface shapes. The document in the upper left and the upper right both have a high frequency of the term "information" (bulge to the right).[37]

In [18], the symbolic objects used for visualizing attribute sets of a data set are called icons. (This term is used as a synonym for glyph in this paper.) A conceptual framework and a process modes of feature extraction and visualization using icons are presented. The generation of icons are achieved with a modeling language using icons that has been used in scientific visualization as templates. One of the template icons is the ellipsoid. (Figure 2.15) It is a good candidate for multiparameter fields, for instance tensor fields.

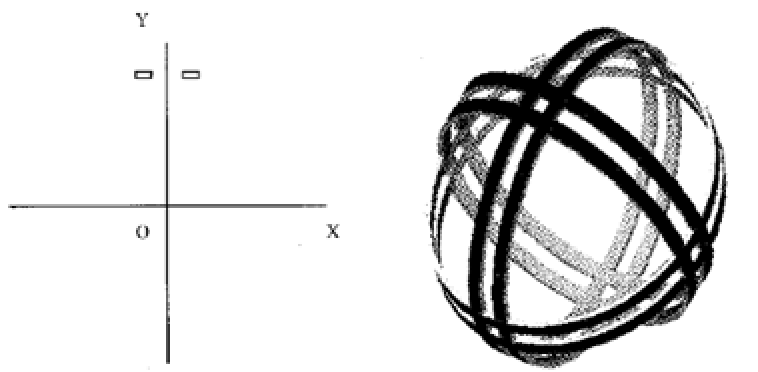


Figure 2.15. Construction of a 3D ellipsoid.[18]

Spatial, geometric and descriptive parameters of these template icons are then modified with the modeling language for visualizing desired features of the data set.

One major drawback of using glyphs for multivariate data is losing the distribution information of attribute values. In this work[38], a distribution glyph is proposed to specifically address how the aggregated data is distributed over the possible range of values. Nested shells are used in glyph design in order to show deviation and variance. In Figure 2.16, a test image with age, education, hours worked and salary are mapped to the X, Y, Z and color. The shell with the vertical wedges show the distribution, the shell in the middle shows the average and the horizontal stripes in a shell show the extent of the variable that is being represented by the shell.

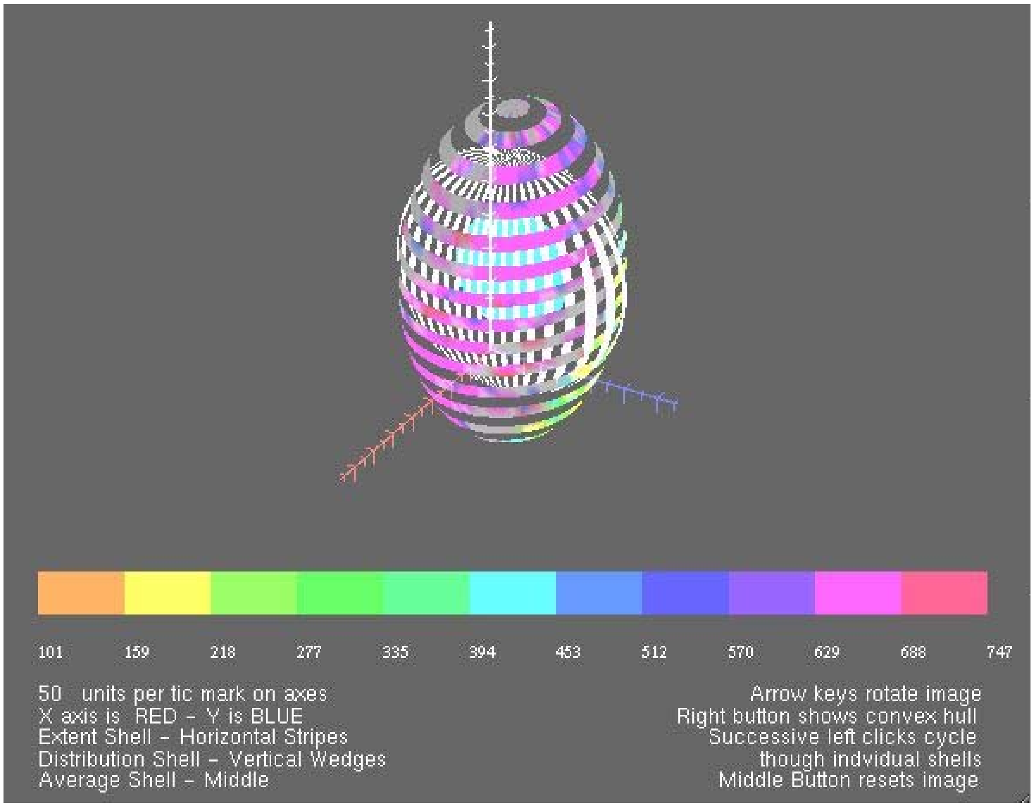


Figure 2.16. Test image with Age, Education, Hours Worked and Salary Mapped to X,Y,Z and Color.[38]

In [39], each item on the wheel is a trend graph depicting change over time. It has N dimensions, each aggregated over time. It abstracts away individual data points. In this example, software project managemt data is visualized over time with glyphs which are referenced in this paper as timewheel glyphs. The conventional way of visualizing temporal data is using a time-series plot. Yet in the timewheel glyph multiple time series are presented with each series rotated around a circle as in Figure 2.17.

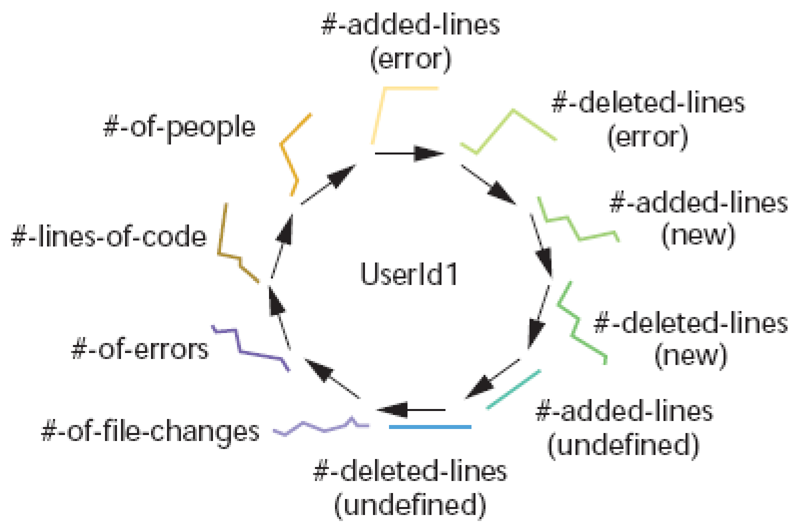


Figure 2.17. Attributes identified by color in a time-wheel glyph.[39]

In this example, the glyphs, coupled with filtering and aggregation help to reveal interesting patterns and anomalies in the data set.

3 Visualization

This thesis proposes a prototype visualization for heterogeneous social networks. Within this prototype, a novel representation method based on node-link diagrams using glyphs is introduced, and a set of interaction methods are employed in order to facilitate the exploration process.

The proposed method specifically addresses the challenge introduced by the cognitive load of following connectivity information between different node and link types by reorganizing the topological information with smooth transformations. The analyst is given control of the reorganization process for an exploratory experience. The representation method is based on node-link diagrams to fulfill conformance criteria and uses glyphs in order to efficiently represent the multivariate nature of heterogeneous social networks.

The glyphs are used to encode connectivity information. Basically, adjacent nodes from the same or different node types are aggregated into a super node within a set of rules that is represented by a glyph which will be referred as a *glyph node*. The connectivity information defined in terms of the classical model of a graph doesn't convey any information in heterogeneous networks. Thus, connectivity is carefully encoded using the semantic graph model.

Using glyphs for connectivity information helps simplifying the visualization to a certain extent by eliminating visual clutter, focusing on a specific relation type without losing information on the other types, and easily identifying weak ties.

The proposed method in this thesis is different from other social network visualization tools because it is specifically designed for heterogeneous node and link types. In the subequent subsections the details of the system about data abstraction, transforming the network, drawing nodes and links, visualizing attributes and visual dimensions of the glyph will be presented.

3.1 Data Abstraction

The mathematical abstraction of an heterogeneous network is a semantic graph and semantic graphs are annotated with an ontology. Ontologies specify relations that can exist in an heterogeneous social network. Although there are methods to detect relations in a given network data set[relationship detection, connection subgraphs], in this thesis a well defined ontology is assumed to be present and the semantic graph comprises of undirected edges.

Throughout this thesis, a semantic graph G is defined with (V, E, vt, et) where V denotes the set of vertices and E denotes the set of edges. The ontology associated with the semantic graph is defined with $O = (T_V, T_E)$. $T_V = \{t_1, t_2, \dots, t_n\}$ is the set of vertex types and $T_E = \{(t_i, t_j) : t_i, t_j \in T_V\}$ is the set of edge t ypes. vt denotes a mapping from V to T_V, and et denotes a mapping from E to T_E.

Figure 3.1 shows an example ontology. It represents the actors, their attributes and relations in a company data set. The data set is synthetic, i.e. it has been randomly generated.The node types are Employee, Department, Project and Activity. There is a "Works In" relation between Employee and Department node types, "Works For" relation between Employee and Project node types, "Planned By" relation between Project and Department node types, "Attends" relation between Employee and Activity node types and finally "Takes Place" relation between Activity and Department node types. The Employee node type is described by Age, Gender and Origin attributes, Department is described by Opening Date attribute and Project and Activiy node types are described by Type attributes.

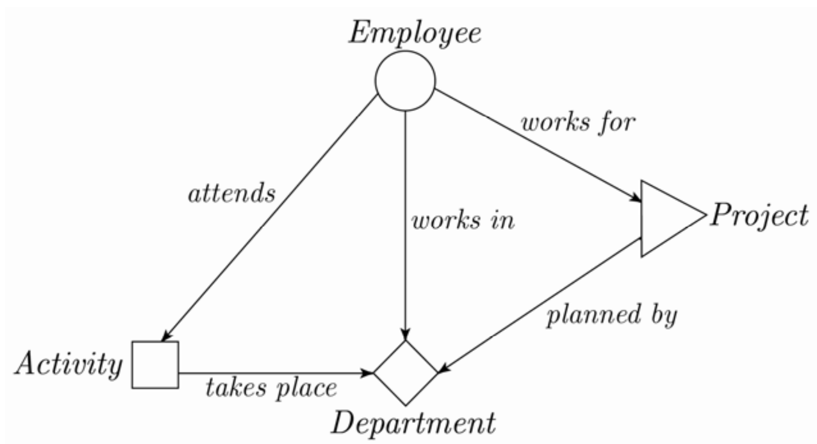


Figure 3.1. Ontology of the "Company" data set.

3.2 Transformations of the Network at Different Levels

Since there is a wealth of information in the network due to the heterogeneous node and link types, one can easily get lost while attempting to understand the structure of the network. Transformations serve as a medium to arbitrarily organize the network by composing aggregate data units out of individual node and relation types at ontology and network level. By forming aggregate data units, the network is simplified to a certain extent allowing to focus on specific node and relation types without losing any information.

Aggregating nodes into super nodes and representing these super nodes as glyphs form the basis of the network transformations. These transformations are done interactively by the user at two different levels: at ontology level , and at network level. At ontology level, different node types are merged into a *class* within a set of rules. Figure 3.2 shows two ontology level transformations on the example data set introduced in 3.1. We will call this ontology a *derived ontology.*

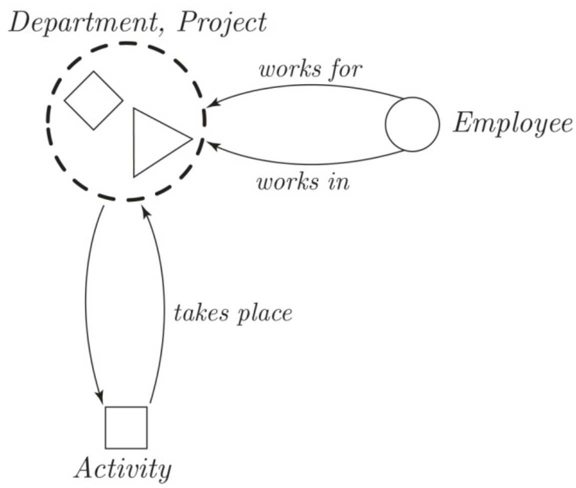


Figure 3.1. Result of transformation 1(left) aggregates Project and Department node types and result of transformation 2(right) aggregates Employee, Department and Project node types.

A derived ontology is denoted by $O_D = (T_{V_A}, T_E)$ where $T_{V_A} = \{C_1, C_2, \ldots, C_n\}$ is a set of classes of types. In a class, there is a relation defined in the ontology for each pair of node types, formally denoted as $C_i = \{ (t_1, t_2, \ldots, t_n) : t_i,\ t_j \in T_V, (t_i, t_j) \in T_E\}$. Another transformation that is possible at the network level is extending the set of node types by converting an attribute belonging to a specific node type and add to T_V. This operation is also controlled by the user. When it is done, a new edge t ype between the converted attribute and the node type owning the attribute type is added to the set T_E automatically.

As mentioned above ontology level changes are operated by the user, then they are automatically propagated to network level (thus modifying the semantic graph.) To formulate the modifications in the semantic graph, a *derived graph* is introduced. It is denoted by $G_D = \{V_A, E_A, v_A t, e_A t\}$. $V_A = \{V_1, V_2, \ldots, V_{n\}}$ is the set of vertices including the aggregated ones, and each V_i denotes the aggretaged vertex. E_A is the set of edges, defined between single or aggregated vertices, $v_A t$ is a mapping from V_A to T_{V_A} and $e_A t$ is a mapping from E_A to T_E. At ontology level, if two node types are merged forming a class, at network level, two vertices that belong to these two types are aggregated into an aggregated vertex (also referred as a super node) if only if they are connected.

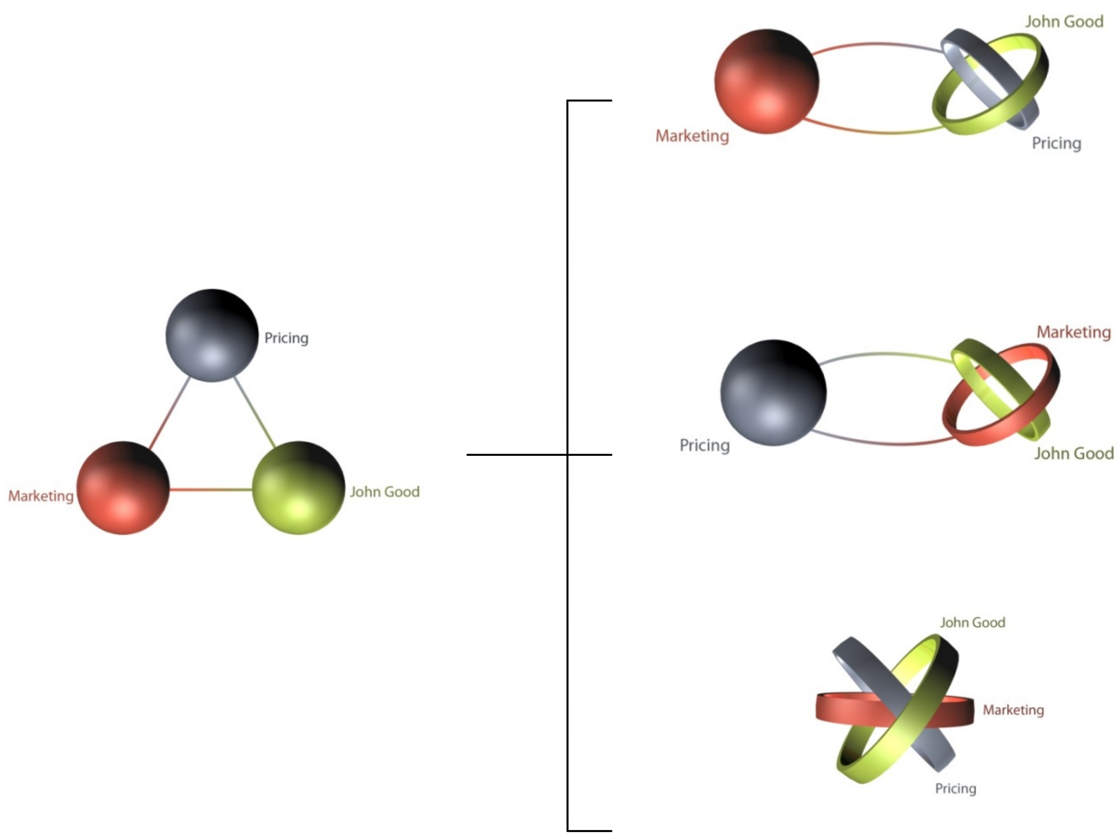


Figure 3.2. Possible transformations(on the right) of the same subgraph (on the left). The subgraph comprises of 3 vertices that are of different types (Grey represents project, orange represents department and green represents employee node types). Since they are strongly connected, any pair can be aggregated as well as all three of the vertices(bottom right).

This condition can be generalized as the following: two classes A and B, containing 1 to n node types can be merged if and only if there is a relation defined for every pair of node types formed out of A and B. This condition is reflected to the network level by aggregating vertices if and only if there is an edge between the underlying vertices(Figure 3.4) unless they are not of the same type. (Figure 3.5) this condition is relaxed if there is a relation defined in the ontology between the same node types.The user is also allowed to directly transform the network, i.e. with the interaction tools, s/he can do local changes to the network by arbitrarily aggregating vertices as long as they follow the connectivity condition described above.

Figure 3.3. An example transformation. Although there is not an edge between the green nodes in the original subgraph(left), they can be aggregetad because they belong to the same node type and there is not a relation defined in the ontology between green node types.

These transformations are believed to be crucial in heterogeneous networks, due to the cognitive challenge of following connectivity information between different node and edge types. And by giving total control to the user for selecting the types for merging, s/he is allowed to focus on speficic types of relations at a time without losing connectivity information on other types. By converting attributes to node types, s/he can investigate the associations between the attributes and the observed pattern. Furthermore, as will be presented in the case study in section 4, these transformations help quickly identify weak ties which is otherwise a quite challenging task due to the complexity introduced by heterogeneous types. Figures 3.6 to 3.8 show example transformations on the network on the same data set.

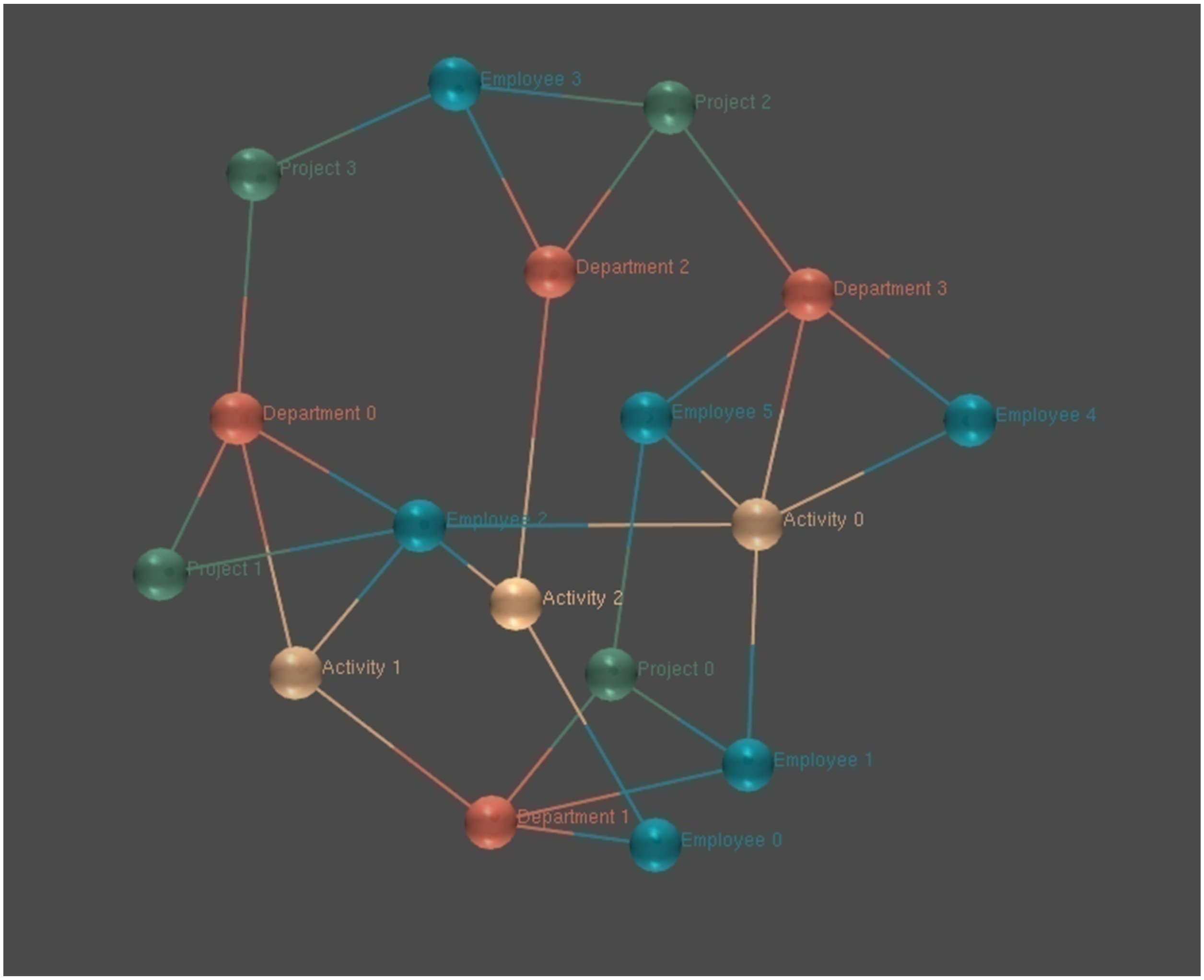


Figure 3.4. The company data set visualization prior to applying any transformations.

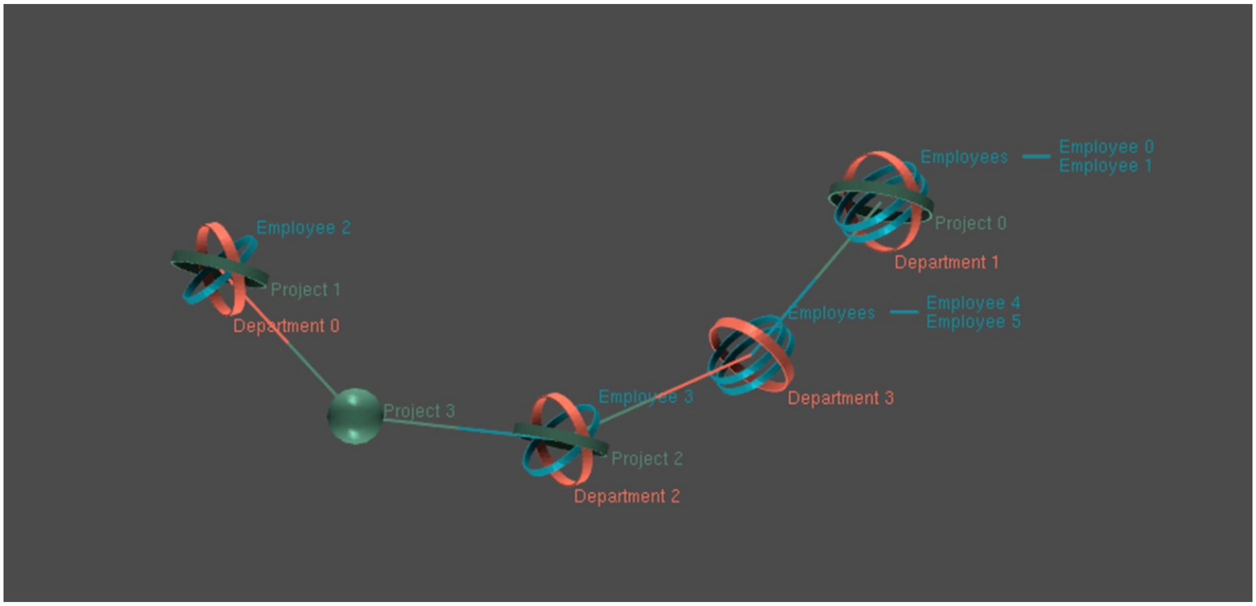


Figure 3.5. Employee-Department-Project node types are merged. Activity node type is excluded from the visualization.

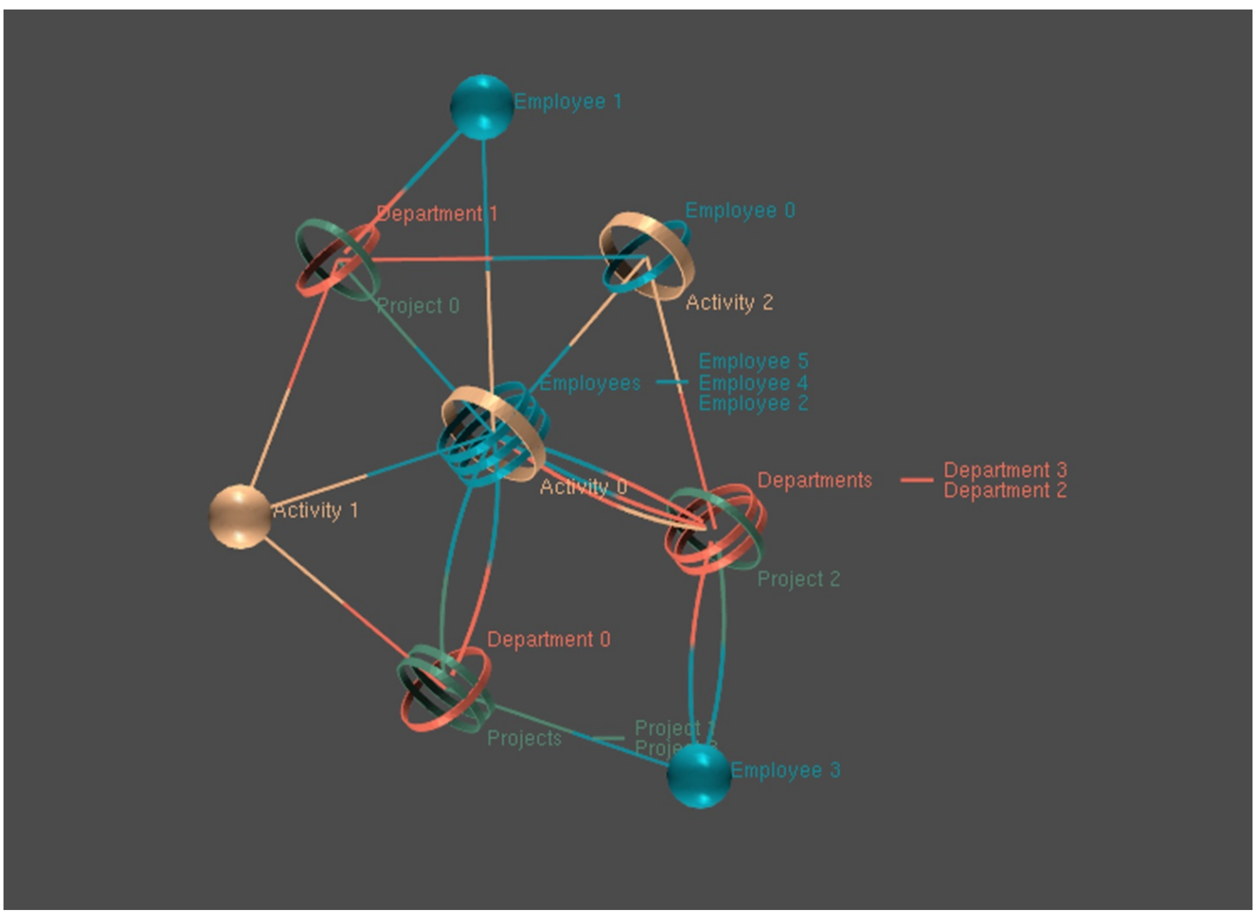


Figure 3.6. Project-Deaprtment and Employee-Activity node types are merged.

3.3 Drawing Nodes

As a result of the transformations described above, the network may contain some aggregated nodes. These nodes are represented by a specially designed three dimensional glyph and single nodes are represented with a sphere. The 3D glyph is designed in such a way that it doesn't diverge from the familiarity of standard node-link diagrams, it has a larger number of visual dimensions than the spherical node, and it is rotationally symmetric so that it conveys the same amount of information from all the angles that is possibly viewed.

The specific glyph represented in this paper is very similar to the ellipsoids presented in [feature extraction and iconic visualization] although functionally different and has a circular form rather than an ellipse. This modification was done for the sake of establishing symmetry. Each underlying node of the super node is represented with a ring. The color of each ring is set to a unique color which belongs to the specific node type that the ring is representing. These rings are positioned at the center of a hypothetical enclosing sphere. The radii of these rings are equal (unless an attribute of a certain type is being visualized with radius), but their orientation and position differ. The number of different node types is used to set the angle. If there is more than one node belonging to the same type, they are positioned at the same orientation, seperated with a preset offset.(Figure 3.9)

Some visual parameters in the glyph are used to encode inherent information that comes from the underlying semantic graph. The radius of the enclosing sphere in a node glyph is mapped to the number of nodes in the aggregated node. This way, central figures for specific node types are visually identified. But, the number of nodes of a specific type may exist in large quantities compared to the other node types in the data set . Therefore, linearly mapping the radius of the sphere to the number of nodes in the glyph may visually introduce a bias. The important task is to visually identify central actors by taking node type quantity into account. Thus, centrality is normalized by assigning a weight ratio to each node in the glyph according to the number of nodes of that node type that exist in the data set.

Figure 3.9. Three example glyph nodes. The nodes that belong to the same node type have the same orientation. The ring widths represent node type affinities.

Another visual parameter that is used for conveying semantic information is the ring width. This parameter is mapped to the node type affinity. For instance, if nodes from type A are usually connected to many nodes from type B, then the affinity of type A for type is high. (also rescaled according to the numbe of nodes in the data set) Under such circumstances, the width of the rings representing nodes from type B are thinner. Examples of glyphs with varying ing widths are represented in Figure 3.9.

Encoding semantic information with these visual parameters is a design choice we have made hence they can not be interactively set or modified by the user as opposed to the other visual parameters that are used for attribute visualization.

3.4 Drawing Links

When nodes are merged in to an aggregated node, their links are updated. So, the number of links that an aggregated node has is the total number of links that the underlying nodes have with the other nodes except the links they make with each other. So, there is a great possibility that the aggregated node has more than one link. For each tie that an underlying node has, a link is drewn from the glyph node to the adjacent node. If there is a single link between two nodes, it is drawn as a straight line, and if there is more than one link, then they are drawn as curves to prevent visual clutter. Since there is more than one node type in an aggregated node, the links that tie the aggregated node with other introduces an ambiguity. This is resolved by linearly interpolating the 2 colors that belong to the two adjacent node types. In Figure 3.10, links that are drawn between a gylph node and a neigbor node is shown.

Figure 3.10 Two links drawn as curves between a glyph node and a neihbor node. The neighbor node is adjacent to both of the underlying nodes in the gyph node.

When there is more than one node that belong to the same node type, then which of these nodes the tie is actually connecting is ambiguous. Our goal is to visualize the patterns of relations between aggregated nodes, we believe that the number of links and the type information are more relevant. Hovewer, the user can see the exact connection pattern by double clicking on the glyph node. The glyph node is coverted back to the underlying subgraph.

3.5 Attribute Visualization

Adding attributes to the visualization might be necessary during analysis to investigate potential effects of the attribute variables on the existing ties. Moreover, the distribution of specific attribute values might need to be analyzed in a specific node type. In the ontology of the graph, an attribute is a feature of the object. As mentioned before, each attribute can be converted to a node type and added to the ontology generating a new relation between the node type and the attribute. Then, these two node types can be merged into a class causing aggregated nodes in the network. This way, it is intuitive to observe the connection patterns of the nodes that have specific attributes. Figure 3.11 and 3.12 show adding attributes to the company data set visualization and merging attribute node types with others respectively.

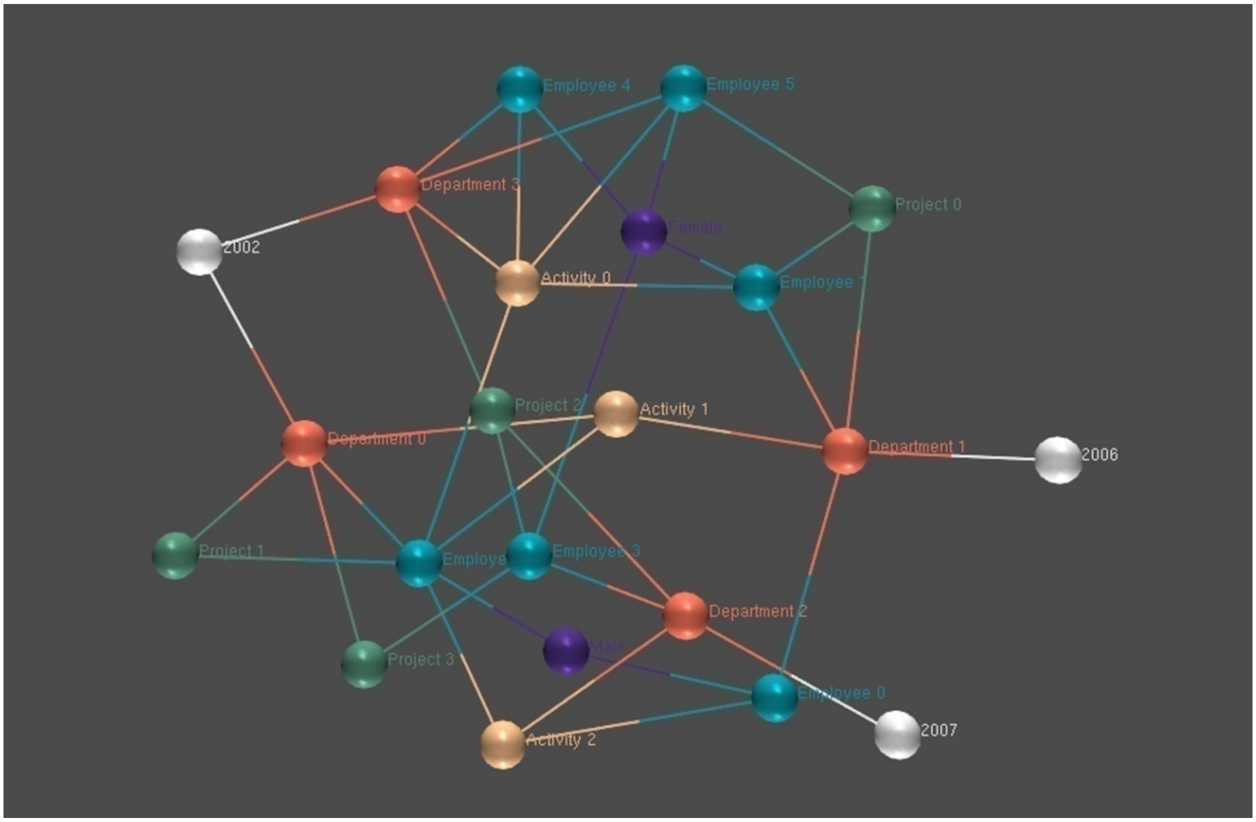


Figure 3.11. The company data set visualization. Opening Date attribute of Department node type and Gender attribute of Employee node type are converted to node types and added to the visualization.

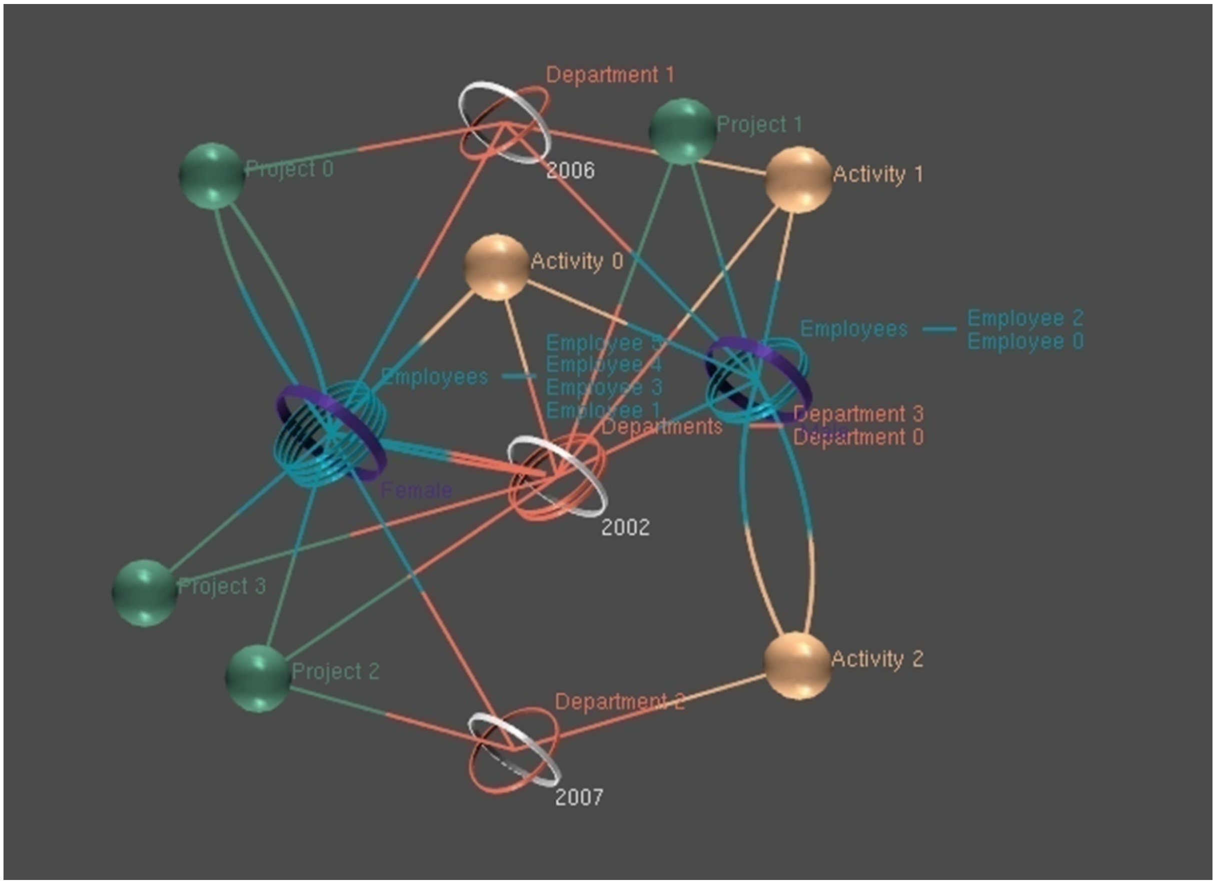


Figure 3.12. Employee-Gender and Company-Opening Date node types are merged.

3.6 Interaction and Animation

The goal of employing interaction techniques in visualization is to allow an exploratory process. The interaction tools presented in this thesis are employed specifically for selecting any subset of the ontology to visualize, make transformations at ontology and network levels, zoom in and zoom out to navigate through the netwotk and zoom in and out to see the overview or focus on certain areas.

Selecting the node and edge types to visualize and ontology level transformations are done by the help of a menu that can be toggled as visible or invisible. The actions in this menu are organized as a tree, so that with each new action , subtrees of them become visible serving as a guide for the currently available actions. Moreover, visual feedback is provided for the actions that become unavailable as a result of one of the previous actions. This way, the user is guided through the exploration process. (Figure 3.13)

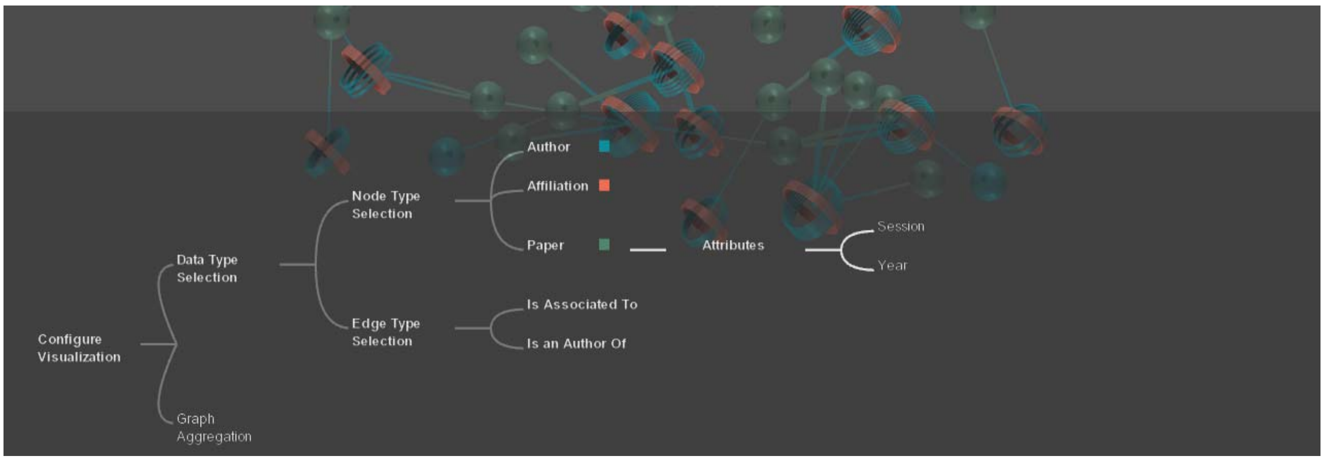


Figure 3.13. Snapshot of the visualization of the Infovis co-authorship data set(introduced in Section 5) while the menu is visible.

By the help of this menu, node/edge types can be activated causing nodes that belong to the activated node/edge type to be added to the visualization and vice versa. Likewise, active node/edge types can be deactiveted, thus being removed from the visualization. Ontology level transformations are also controlled through this menu. To each node type A, a subtree is appended consisting of the node types that can be merged with A. Clicking on a node type from this submenu triggers the transformation. For a smooth transition between different topologies, the merging operation of the nodes are animated. Rather than instantly making the necessary updates to the graph, the nodes to be merged advance their position through each other, and when finally the distance is very small(close to zero) they are replaced by a glyph.

Transformations at the network level, i.e. locally editing the network can be achieved by simple drag and drop, i.e. a single or super node can be dragged and dropped on a single or aggragated node. If this operation doesn't violate any condition for merging, a new glyph is drawn representing the aggregated nodes merged by drag&drop. Otherwise, (if this operation doesn't obey the rules for merging), then the position of the dragged node is automatically set to a certain distance from the target node as a feedback about the failure of the operation.

4 Case Study

4.1 Co-authorship Data of Infovis 2008

In this case study, how the proposed method in this thesis can be used for exploring co-authorship data is presented. The co-authorship data set basically contains information about authors, their papers and affiliations of the Infovis 2008 conference . There are two relation types defined in this data set. The first one is “is an author of” relation which is between authors and papers and the second one is “is associated to” relation which is defined between authors and affiliations. (Figure 4.1) To further describe the properties of node types, there are attributes associated with each node type. These are title, commitee member and gender information for authors, session information for papers, country and industrial/academic information for affiliations.

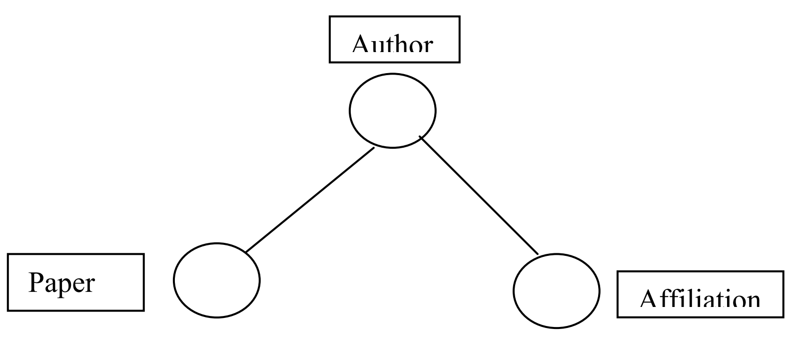


Figure 4.1. Ontology of the co-authorship data set.

there are x number of nodes in the graph. Despite the relative smallness of this network, it poses the challenges of visualization of heterogeneous social networks using standard node-link diagrams. Figure 4.2 show the whole network before applying any transformations.

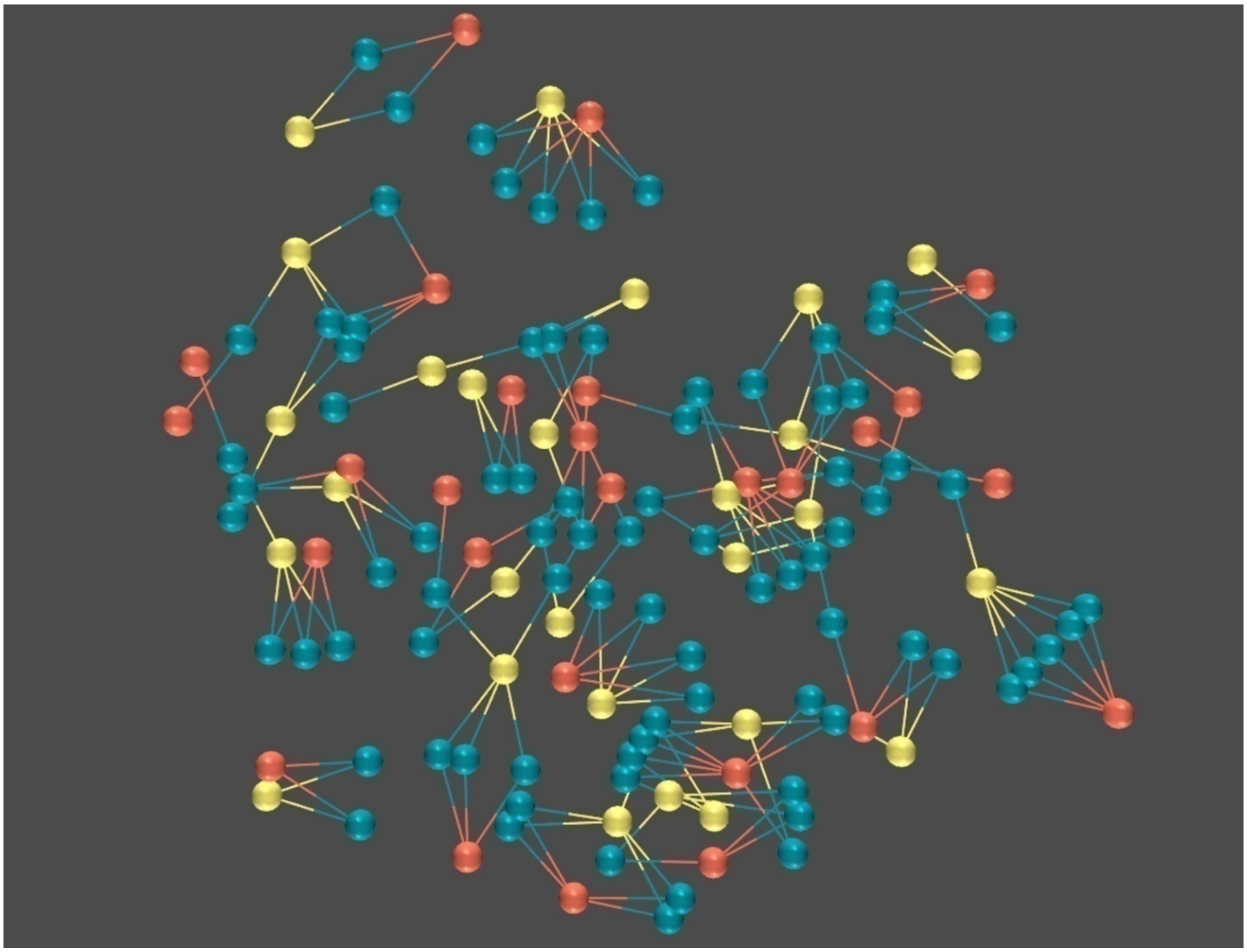


Figure 4.2. Snapshot of the visualization of co-authorship network, all the node and relation types are added to the visualization. Blue nodes represent authors, orange nodes represent affiliations and yellow nodes represent papers.

To understand the structure of this network, ontology and network level transformations are applied. For instance, Figure 4.3 shows the result of merging author and affiliation node types.

This figure immediately reveals the connectivity pattern of this relation type, which in turn helps to identify a number of characteristics of the network. For example, UNCC and Georgia Tech are central affiliations, they both have the larger number of authors associated to them and Heidi Lam, Holy Schmidt and Lee Byron are not associated to any affiliations (weak ties). But we can infer that Holy Schmidt and Lee Byron are related to University of Calgary and IBM respectively due to the collaboration patterns. Another observation that can be made is the relation between the paper and affiliation node types. There is not a relation defined between these node types in the ontology, however they are implicitly related. Hence, this implicit relation can also be used to define central actors among affiliatons in terms of the number of papers it is related to. According to the secondary relation between papers and affiliations, IBM is the most prestigious actor among affiliations.

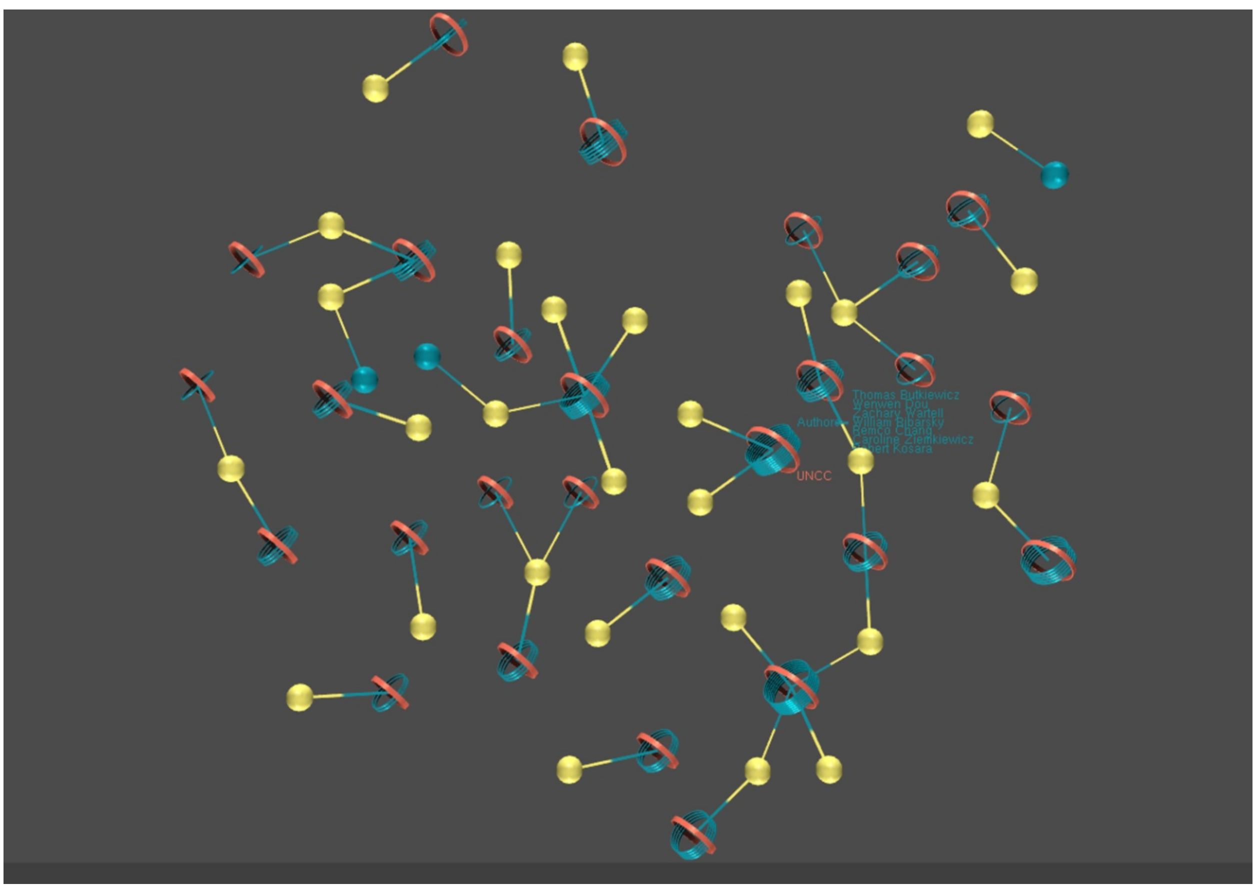


Figure 4.3. Snapshot after Author and Affiliation node types are merged.

To understand if the attribute values have any effects on the observed characteristics, related attributes were added to the visualization by converting to node types. For instance, the title attribute of the authors may be correleated to the connectivity distribution between authors and affiliatons due to the fact that “the weak tie” Lee Byron is connected to the “designer” node (his/her title is a designer) which is not connected to any authors who have an affiliation. (Figure 4.4)

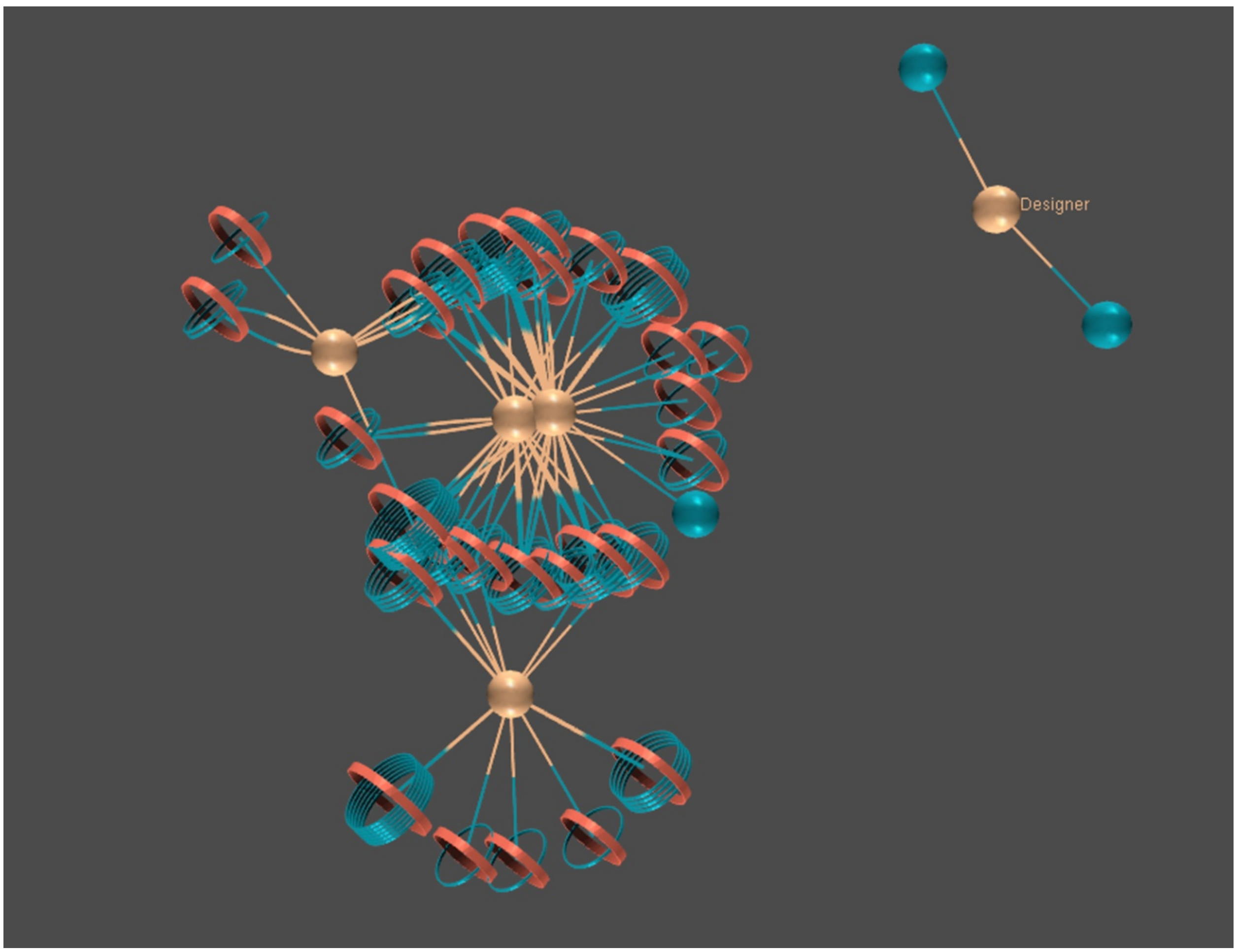


Figure 4.4. Snapshot after Title attribute of Author node type is added to the visualization and also Paper nodes are removed.

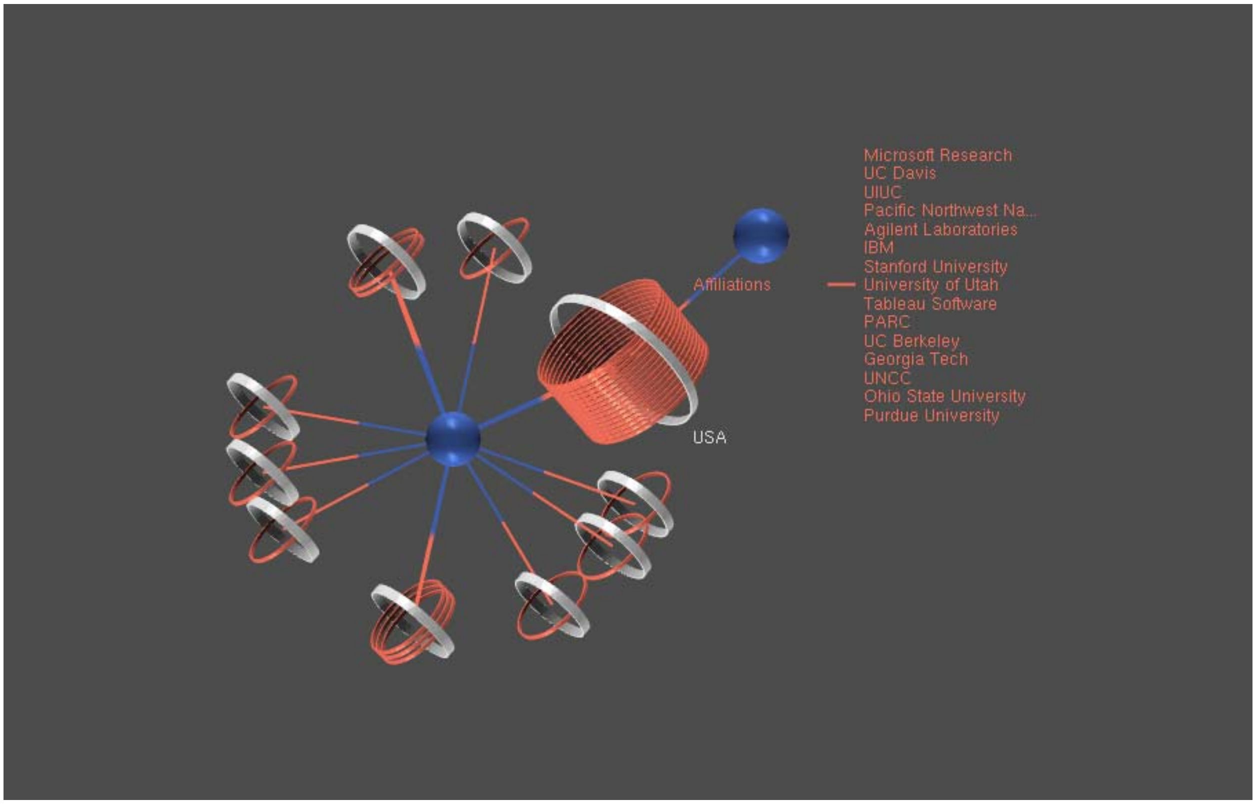


Figure 4.5. Visualization of Affiliation nodes with its attributes Country and Industrial/Academic are converted to node types and added to the visualization.

Moreover, the properties of a node type can be investigated in depth by converting its attributes to node types and adding to the visualization. For instance, by visualizing the affiliation node type in relation to its attributes and applying transformations, we can easily infer that all the industrial affiliations are based in USA. (Figure 4.5)

The interaction methods presented in this thesis were used to ease the exploration process. Besides activating or deactivating node and edge types to visualize the subsets of the network, the ability to locally edit the network allows making local analysis and resolving ambiguities.

5 Conclusions and Future Work

We have introduced a novel representation method for visualizing heterogeneous social networks. This method integrates standard node-link diagrams with glyph based representations of multivariate data. The proposed method efficiently organizes the node and relation types making it possible to explore the heterogeneous social network data.
A set of interaction techniques were presented to transform the network at ontology and network level. Animation was also used for smooth transitions between different topologies. Finally, we have illustrated how the proposed method tackles social network analysis tasks in heterogeneous social networks with a case study.

We plan to extend our study by making use of the visual dimensions of the glyph in a more systematic and efficient way. The visual dimensions of the glyph can be used to encode attribute information allowing comparisons between underlying nodes in a glyph in terms of the attribute values.

References:

[1] L.C. Freeman, “Visualizing Social Networks” *J. Social Structure*, vol. 1, no. 1, 2000.

[2] V. Batagelj and A. Mrvar, “Pajek: Analysis and Visualization of Large Networks” *Graph Drawing Software,* pp. 77-103, Springer, 2003.

[3] E. Adar, “Guess: a language and interface for graph exploration”, In *CHI’06: Proceedings of the SIGCHI conference on Human Factors in computing systems,* pages 791–800, New York, NY, USA, 2006, ACM Press.

[4] S. Borgatti, M. Everett, and L. Freeman, “UCINET V user’s guide”, Analytic Technologies, Natick, MA, 1999.

[5] M. Barthelemy, E. Chow and T. Eliassi-Rad. “Knowledge Representation Issues in Semantic Graphs for Relationship Detection.” *AI Technologies for Homeland Security: Papers from the 2005 AAAI Spring Symposium,* AAAI Press, pp. 91-98, 2005.

[6] S. Wasserman and K. Faust, “Social Network Analysis: Methods and Applications”, *Cambridge University Press*, 1994.

[7] I. Herman, G. Melancon, and M.S. Marshall, “Graph Visualization and Navigation in Information Visualization: A Survey”, *IEEE Trans. Visualization and Computer Graphics,* vol. 6, no. 1, pp. 24-43, 2000.

[8] F. Van Harmelen and C. Fluit, “Ontology-Based Information Visualization”, *Proc. of the Fifth International Conference on Information Visualization,* pp. 546, 2001.

[9] P. Eades, “A Heuristic for Graph Drawing”, *Congressus Numerantium,* vol. 42, pp. 149-160, 1984.

[10] G.G. Robertson, J.D. Mackinlay and S.K. Card, “Cone Trees: Animated 3D Visualizations of Hierarchical Information”, *Proc. Human Factors in Computing Systems,* pp. 189-194, ACM Press, 1991.

[11] M. Ghoniem, J.-D. Fekete, and P. Castagliola, “On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis”, *Information Visualization,* 4(2):114–135, 2005.

[12] J. Heer and D. Boyd. “Vizster: Visualizing Online Social Networks”, In *Proceedings of the 2005 IEEE Symposium on Information Visualization,* pages 32–39, IEEE Computer Society, 2005.

[13] N. Henry and J.-D. Fekete, "MatrixExplorer: a Dual-Representation System to Explore Social Networks", *IEEE Transactions on Visualization and Computer Graphics,* 12(5):677–684, 2006.

[14] M. Granovetter, "The Strength of Weak Ties", *American Journal of Sociology,* vol. 78, no. 6, pp. 1360-1380, 1973.

[15] C. Plaisant, B. Lee, C. S. Parr, J.-D. Fekete, and N. Henry, "Task taxonomy for graph visualization", In *Beyond time and errors: novel evaluation methods for Information Visualization (BELIV'06),* pages 82–86, Venice, Italy, 2006. ACM Press.

[16] B. Shneiderman and A. Aris, "Network visualization by semantic substrates", *IEEE Transactions on Visualization and Computer Graphics,* 12(5), 2006.

[17] M.Wattenberg, "Visual exploration of multivariate graphs", In *Proceedings of the SIGCHI conference on Human Factors in computing systems,* pages 811–819. ACM Press, 2006.

[18] T. Walsum, F. H. Post, D. Silver, F. J. Post, "Feature Extraction and Iconic Visualization", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 2, No. 2. (June 1996), pp. 111-119.

[19] L Singh, M Beard, L Getoor, MB Blake, "Visual Mining of Multi-Modal Social Networks at Different Abstraction Levels", *Information Visualization, 2007. IV '07. 11th International Conference* (2007), pp. 672-679.

[20] C. Chen, "Information Visualization: Beyond The Horizon", S*pringer - Verlag*, London, 2004.

[21] Z. Shen, T. Eliassi-Rad, K. Ma, "Visual Analysis of Large Heterogeneous Social Networks by Semantic and Structural Abstraction", *IEEE Transactions on Visualization and Computer Graphics,* vol: 12, no:6, pp. 1427-1439, 2006.

[22] B. Schneiderman, "The Eyes Have It: A Task By Data Type Taxonomy For Information Visualization", *Visual Languages, 1996. Proceedings., IEEE Symposium on* (1996), pp. 336-343.

[23] B. Schneiderman, "Inventing Discovery Tools: Combining Information Visualization With Data Mining", *Information Visualization,* Vol.1 No. 1, 2002.

[24] C. Ware, "Information Visualization: Perception For Design", *Morgan Kaufmann Publishers*, 2000.

[25] J. Heer, S. K. Card, and J. A. Landay. prefuse: a toolkit for interactive information visualization. In *SIGCHI conferenceon Human factors in computing systems*, pages 421–430, New York, NY, USA, 2005. ACM Press.
[26] G. di Battista, P. Eades, R. Tamassia, and I.G. Tollis, “Algorithms for drawing graphs: an annotated bibliography”, *Computational Geometry: Theory and Applications*, Vol. 4, No..5, pp. 235–282, 1994.
[27] B. Schneiderman, “Designing the user interface: strategies for effective human-computer interaction”, Reading, *Massachusetts: Addison-Wesley*, 1986.
[28] Herman Chernoff, "The use of faces to represent points in k-dimensional space graphically," *J. Am. Stat. Assoc.*, v68, 361-368 (1973).
[29] D. Silver, N. Zabusky, V. Fernandez, M. Gao, and R. Samtaney, “Ellipsoidal Quantification of Evolving Phenomena”, Scientific Visualization of Natural Phenomena, N.M. Patrikalakis, ed., pp. 573-588. *Tokyo: Springer-Verlag*, 1991.
[30] N. Henry, J D Fekete, and Micheal J. McGuffin, “NodeTrix: A Hybrid Visualization of Social Networks”, *Information Visualization*, 2008.
[33] L Singh, M Beard, L Getoor, MB Blake. Visual Mining of Multi-Modal Social Networks at Different Abstraction Levels. *Information Visualization, 2007. IV '07. 11th International Conference* (2007), pp. 672-679.
[34] O'Madadhain, J., Fisher, D., White, S. and Boey, Y.-B, “The JUNG (Java Universal Network/Graph) Framework”, University of California, Irvine, California, 2003.
[35] David Ebert, James Kukla, Christopher Shaw, Amen Zwa, Ian Soboroff, and D. Aaron Roberts. “Automatic shape interpolation for glyph-based information visualization”, *IEEE Visualization*, October 1997.
[36] Martin Kraus and Thomas Ertl, “Interactive data exploration with customized glyphs”, In *Proceedings of WSCG'01*, pages 20.23, 2001.
[37]D Ebert, R Rohrer, C Shaw, P Panda, J Kukla, A Roberts, “Procedural Shape Generation For Multi-Dimensional Data Visualization”, 1999.
[38] E B Chlan, P Rheingans, “Multivariate Glyphs For Multiobjet Clusters”, *IEEE Symposium on Information Visualization* 2005 October 23-25.

[39] Mei C. Chuah and Stephen G. Eick. “Information rich glyphs for software management”, *IEEE Computer Graphics and Applications*, pages 2.7, July.August 1998.
[39] T. Fruchterman and E. Reingold, “Graph drawing by force-directed placement”, Software Practice and Experience, 21(11):1129 1164, 1991, also as Technical Report UIUCDCS-R-90-1609, Dept. of Computer Science, Univ. of Illinois at Urbana-Champaign, 1990.
[40] Netminer ii: Social network mining software. website: http://www.netminer.com/NetMiner/home 01.jsp.
[41] Brooklyn Public Library, Brooklyn Expedition, http://www.brooklynexpedition.org/structures/glossary_latin.html.
[42] H. Chernoff. “The Use of Faces to Represent Points in k-Dimensional Space Graphically”, *Journal of the American Statistical Association*, Vol. 68, No. 342, 361-368, 1973.
[43] R. Picket and G. Grinstein. “Iconographic displays for visualizing multidimensional data”, *IEEE Press*, 514-519, 1988.
[44] VGSPS mbH, Beratungsgesellschaft für statistische Programmsysteme, http://www.ps-explore.de/html/chernoffgesichter_0.html.