12-2012

A Comparison of Display Techniques for Large Graphs

Ana Rocio Pena
University of Texas-Pan American

A COMPARISON OF DISPLAY

TECHNIQUES FOR LARGE

GRAPHS

A Thesis

by

ANA ROCIO PEÑA

Submitted to the Graduate School of the
University of Texas-Pan American
In partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2012

Major Subject: Computer Science

A COMPARISON OF DISPLAY

TECHNIQUES FOR LARGE

GRAPHS

A Thesis
by
ANA ROCIO PEÑA

COMMITTEE MEMBERS

Dr. Richard H. Fowler
Chair of Committee

Dr. Wendy A. Lawrence-Fowler
Committee Member

Dr. Pearl W. Brazier
Committee Member

December 2012

ABSTRACT

Peña, Ana Rocio, A Comparison of Display Techniques for Large Graphs. Master of Science (MS), December, 2012, 51 pp., 23 figures, references, 9 titles.

Visualizing information focuses on the display of data in order to provide the user a representation that provides understanding of the data. Information visualization systems typically couple interaction mechanisms for providing overviews of the data with more detailed information through a zooming interface. This thesis compares three different techniques for displaying graphs provided by the *prefuse* visualization system: force-directed node positioning, radial node positioning, and a tree view of graphs. Using a large, real world data set from the South Texas College's Distance Education department, the three visualization techniques are compared for a set of tasks that users routinely need to perform using standard data access techniques. Though the tree view visualization is the most limited in generality of the three techniques, it is found to best provide support for the tasks, in part because of its ability to provide the abstractions that best match the tasks.

TABLE OF CONTENTS

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

Today, the use of computers, mobile devices, and electronics is pervasive. They are constantly used in schools, home, work and many other places to provide knowledge and information. Every computing system provides an interface through which the user communicates and interacts with the system, and the study of Human Computer Interaction examines the phenomena surrounding this interaction. The interaction may utilize icons, task bars, and windows on a personal computer or widgets and keypads on a mobile device. Those tools provide the mechanisms for exchange of information between user and system. Human Computer Interaction investigates both human and machine. The development of Human Computer Interaction was influenced by information science, an older discipline that came into existence before modern technology and is concerned with the organization, management, and manipulation of information (Dix, Finlay, Abowd and Beale, 2003).

The large volumes of the information today has led to new and innovative methods for presenting data to the user, often making the interaction between user and system more intuitive by presenting data in a visually informative way. Visually represented data "catches the eye of the human" and provides a mechanism to explore and comprehend large amounts of information at once. Visualization can be thought of as a method of communicating information by creatingimages, videos, diagrams or animation (Chen, 2004). Visualization provides the ability to comprehend large amounts of data. Representing large data sets in a manner to facilitate comprehension often uses a visual metaphor. A visual metaphor can be defined as the mechanism used to represent information graphically that transforms data that may not be inherently visual to a form that can be represented visually (Dix, Finlay, Abowd and Beale, 2005). Visualization can also provide a mechanism for the user to perceive emergent properties of the data, often not anticipated. The perception of a pattern can often be the basis of a new insight. Visualization facilitates understanding of both large scale and small scale features of the data. It can be especially valuable in allowing the perception of patterns linking local features with overall structure, often referred to as detail and context. In the context of the large scale overview it is easier to comprehend the small scale features of the data.

For the user, effective communication of data is necessary to utilize an interface which accommodates the information in the most informative manner and provides effective methods of interaction. In order for researchers to take a path towards enhancement of computing systems, various interface designs have been developed and interface designs are becoming more complicated as required by the development of new technology (Dix, Finlay, Abowd and Beale, 2003).

Research in interface design is a branch of computer science, and it provides guidelines and principles for the proper design of interfaces elements, such as menus, icons, forms, data display and entry screens. For example, for an Internet user, the browser is an essential tool that incorporates interface design principles in interface elements such as buttons and menus to make it easy for the user to access the web. The biggest challenge many people deal with using computers is often the poor design of the human computer interface, which makes navigation difficult. Usability is an issue for effective design. Unexpected problems occur when poorly designed interfaces make it interaction difficult (Norman, 1988; Shneiderman, 1998).

The larger information displays of today and the future can provide better user interfaces with more sophisticated and resourceful designs that turn a troublesome mass of information into a productive river of knowledge. Human perceptual skills are significant and mainly underutilized in current information and computing systems. Techniques such as dynamic queries, star field displays, tree browsers, tree maps, zoom-able user interface and a variety of widgets can be used to present, search, browse, filter and compare rich information spaces. The FilmFinder (Ahlberg and Shneiderman, 1994) (shown in Figure 1.1) has an extensive scope of dynamic queries by providing a display in which two variables from a database relation are plotted against each other and used as an interactive visualization.

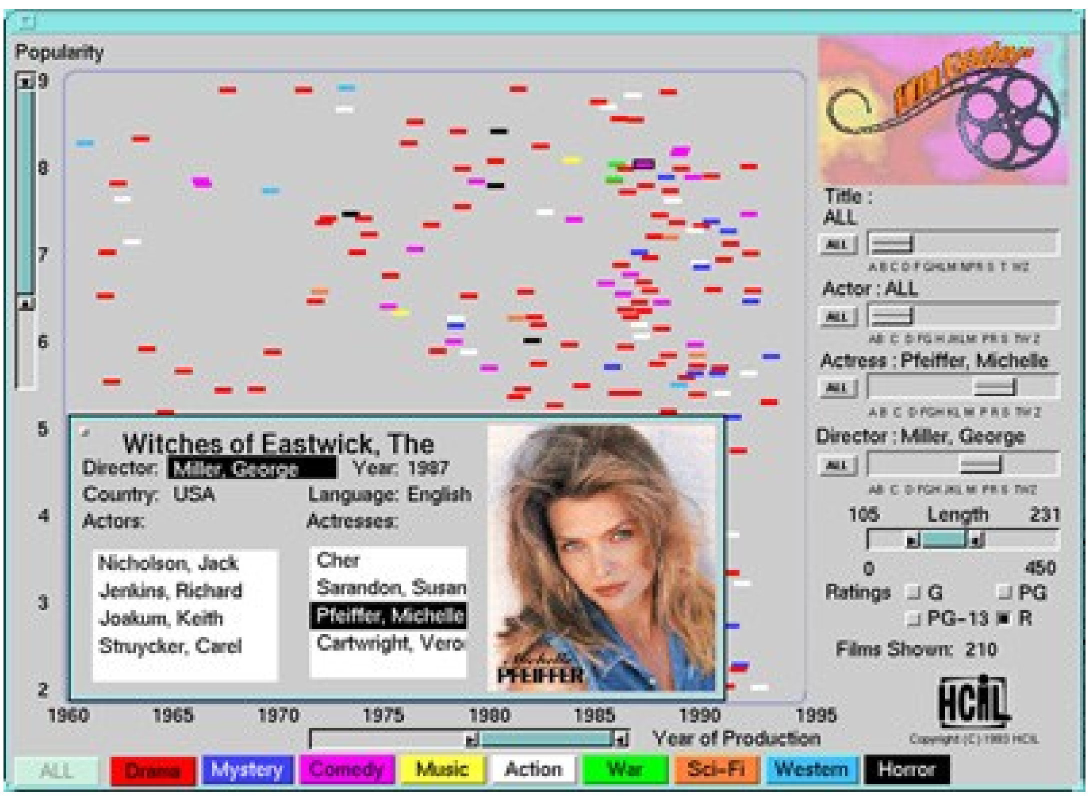


Figure 1.1 FilmFinder application to show dynamic queries

A basic principle for browsing and search is summarized as the visual information and seeking Mantra: Overview first, zoom and filter, then details on demand (Shneiderman, 1996). A user will be able to responsibly and confidently take on more tasks when the system can have better visual displays, direct manipulation interfaces and dynamic queries. The following screenshots in Figure 1.2 show an example of a treemap (Norman, 1988). which incorporates the above visualization mantra. It displays the hierarchical structure of different items such as people, things, qualities, knowledge, places and event.

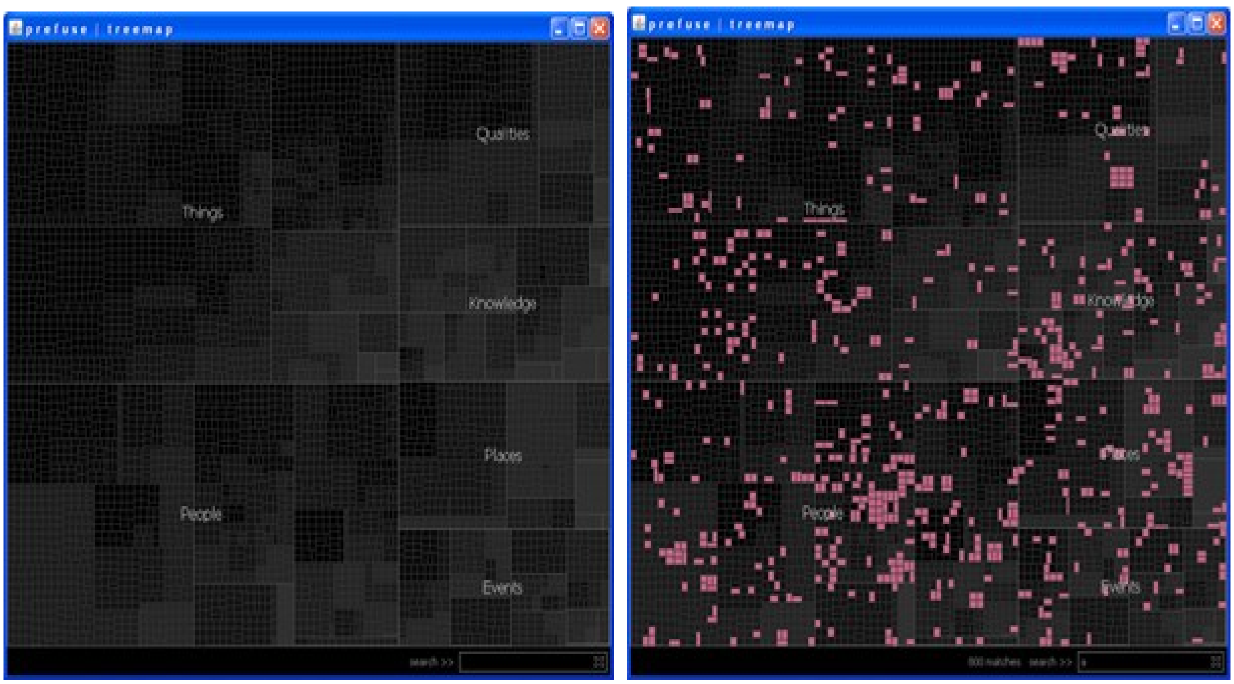


Figure 1.2 Treemap examples.

One area of Human Computer Interaction is information visualization. Information within a computing system can be expressed visually and has received attention by researchers. Information visualization researchers have tried to show that it is important for an application to be responsive to users' actions and also be informative. An example is Microsoft Office with its reasonably intuitive response by the application which can be helpful to the user. Microsoft Office has enhanced the way various styles can be applied to a given text in the word document, e.g., by showing a preview of each style for the target text as the user hovers above the available styles. The application clearly shows interactivity through the user feedback mechanism as the user performs an action. It performs its functions to facilitate the usability of the application.

Human Computer Interaction principles were important in the development of the information visualization application called Vizster (Figure 1.3), which was created by human computer interaction scientists at Stanford University (Heer, Card and Landay, 2005). It provides visualizations of social networks. The application uses profile information from the Friendster network as data input, and then builds a radial network for information visualization that shows the current user at the center. Every friend of the central user appears around the user's radial network using the main image from his or her profile.

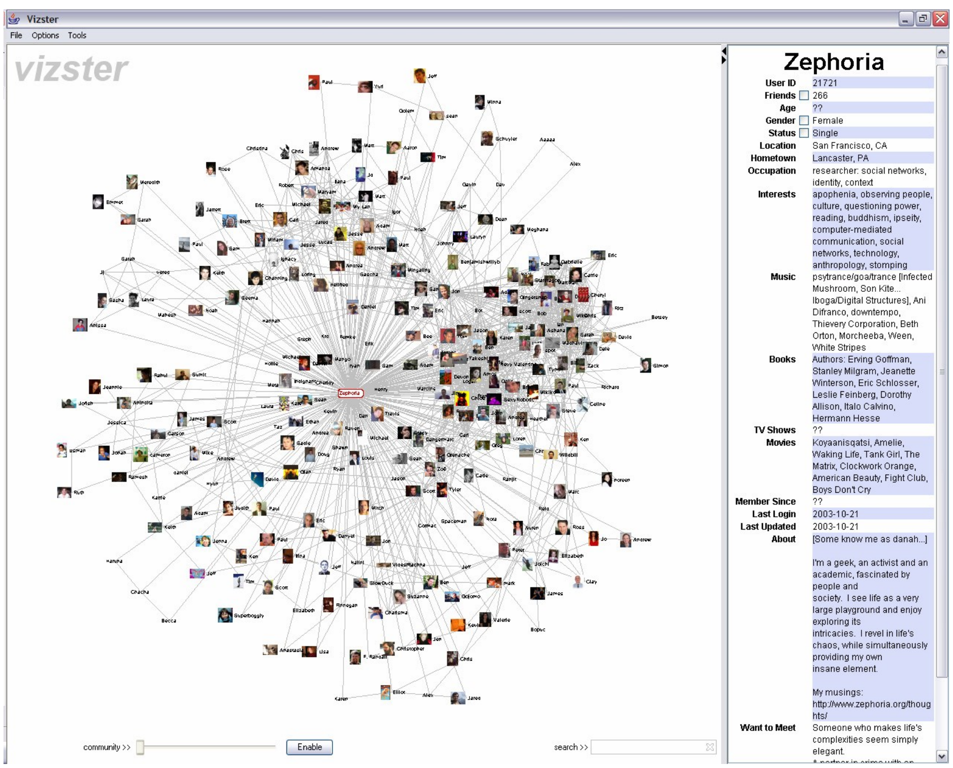


Figure 1.3 Vizster Application for visualization of social networks

Vizster treats every user as a node in a network graph. By clicking on a user's image, all of the data in that user's profile is displayed to the right of the network on the side pane, as shown on Figure 1.3. This provides a visual feel to the social networking experience and makes it more interesting and useful to the users (Heer, Card and Landay, 2005).

One of the main features of this application is the network visualizion. The approach is original and extensible to other networking environments by designing other network applications. One example would be a network representation using Vizster for visualizing classrooms in a school (Heer, Card and Landay, 2005). Every class would contain a professor with numerous students working under that professor, and each student would have a record that needs to be maintained by the professor in order to track the performance in the class.

In this thesis, techniques for the visualization of information using large graphs are explored by using user interfaces incorporating Human Computer Interaction design factors and techniques. The *prefuse* application (Heer, Card and Landay, 2005) is used to create the visual representations and interfaces. This research examines the *prefuse* application and its display and interaction techniques to understand the concepts and implementation of visualization. The domain application is drawn from data for distance education utilization at South Texas College and tracks online classes. Keeping track of student's online classes for the college can be a tedious task for both students and professors. The current work is creates a visual representation of the networks of classes, where every class is linked to a particular section of the class and instructor in order to provide a visual representation through which interaction with all courses and related activities can take place.

1.1 Organization of the Thesis

This thesis deals with the visualization of data using the *prefuse* display program. It begins by introducing the concepts and needs for improved interactivity between humans and computers. Chapter II's literature review provides a roadmap of Human Computer Interaction and its principles. Chapter III's methodology and findings sections discuss the need for visualization of data and a discussion of information visualization, focusing on network visualization and graph drawing using *prefuse*'s tree view, graph view and radial graph view. The chapter also discusses the *prefuse* display program, which is designed with principles of human computer interaction in mind, and supplies the tools to create visual representations in an application domain. Finally, Chapter IV concludes by discussing the importance of visualization, in particular graph visualization.

CHAPTER II

REVIEW OF LITERATURE

People interact with computers in many different, and the interface is crucial in facilitating interaction. There are different ways a user can communicate with the system, such as batch input, direct manipulation, and virtual reality. Most handheld computers, desktop applications, Internet browsers and computer kiosks make use of the well established graphic user interface. Human Computer Interaction is based on both art and science in design of the software system's functionality and its interface (Ware Purchase, Colpoys and McGill, 2002). A well designed interface reveals system functionality to support the user's task, but, if the interface is not well designed, then the functionality is hidden, and the user may have problems completing the task. This chapter discusses the history of Human Computer Interaction, its design principles, and research areas.

2.1 The Beginning of Human Computer Interaction

The desire to build better weapons during World War II created an interest in the study of interaction between humans and machines, a new and significant challenge for researchers. The Ergonomics Research Society, founded in 1949, was concerned about the physical characteristics of machines, systems and their effects on user performance (Dix, Finlay, Abowd and Beale, 2003; North, 2005). The field of ergonomics is concerned with the user performance in the mechanical, computer or manual system. As the use of computers expanded, researchers started to study the interaction between people and computers. In doing this, researchers began to concentrate on each of the physical, psychological and theoretical parts of the interaction. In the early years, information technology professionals were the only people that interacted with computers. Much later, personal computing changed both software and hardware, as direct use of computers became possible for a much wider range of users. Personal computer software includes productivity applications, such as spreadsheets, as well as interactive computer games. With personal computers everyone in the developed became a potential computer user, and this highlighted the insufficiencies of computers with respect to usability for the new class of users. Human computer interaction was one of the first examples of cognitive engineering that was designed for information technology systems that support cognitive tasks and artifacts. Computer graphics rapidly came to recognize that interactive systems were the main point of progress in computer development. This lead to the conclusion that a way forward for computing involved understanding and better empowering users. Human factors engineering had developed many techniques for experimental analysis of human system interactions in control domains (North, 2005). Human Computer Interaction became a valuable area for system improvement operators using the experimental techniques of human factors.

Usability has been the primary focus of Human Computer Interaction, and refers to the ease of use and ability to learn to use computer systems. According to Shneiderman (Shneiderman, 1996), an interface is designed through the use of principles that are derived from experience and which are applied in most interactive systems after being properly refined, extended and interpreted. One principle of interface design is to strive for consistency. Consistent sequences of actions should be used in similar situations, such as the using the same terminology in prompts, menus, and help screens, as well as consistent commands employed throughout. Usability is very important for user interface design because the interface needs to be accessible, comprehensible and intelligible. According to Shneiderman and Plaisant (2010), the US Military Standard for Human Engineering Design Criteria (1999) states that the purpose to achieve required performance by operator, control and maintenance personnel. The goals for requirements analysis focus on the user's needs, which will determine what tasks and subtasks must be carried out. Common tasks are easy to identify, and the functionality must match the need or the users will reject or underutilize the product. It ensures reliability by the actions functioning as specified, e.g., a database data display must reflect the actual database. To address the user's potential sense of mistrust, the system should be available as often as possible, and the system must not introduce errors. The system should also ensure the user's privacy and data security by protecting against unwarranted access, destruction of data and malicious tampering (Shneiderman and Plaisant, 2010). Promoting standardization, integration, consistency and portability are also requirements of usability. Consistency is important, as well, as there should be compatibility across different product versions and compatibility with related paper and other non-computer based systems.

Human Computer Interaction has developed to be expansive and much more diverse than computer science because it has expanded from individual and generic user behavior to include social and organizational computing for all people. (Shneiderman and Plaisant, 2010). It went from desktop office applications to include games, e-learning, and process control.

2.2 Human Computer Interaction Design Principles and Techniques

Human computer interaction focuses on effective interfaces. Every software application has the potential to be made easier to use and allow the users to more quickly learn its use.

Complexity in interface utilization increases difficulty and frustration for the user. Every time the number of application functions increase, the complexity also increases, and the design can easily become uninformed and irregular (Ware Purchase, Colpoys and McGill, 2002). There are no set rules when developing an interactive application. The functionality and environment are the main factors in which the application's design must be based. Making the application user-centric and visually attractive leads to creative design thinking. Scenarios, storyboarding, mockups and prototypes are techniques that provide quick realizations of the design.

Good designs rely on the ability to understand and evaluate the interface during the development process (Norman, 1988). When dealing with design, there needs to be creativity, mental, problem solving process and the representation is more of a physical process of capturing or recording the design. An effective representation technique is very important for any new interface development method focusing on refinement and cooperating roles for producing the interface (Keahey, 1998). Designer, implementer, evaluator, documenter, marketing, and customer are some of the roles included in the interface design. Each of the roles has its own needs for communicating, recording, delivering, reading and understanding in interface design (Geisler, 1998).

Interaction design is based on the artifact produced and how it is going to affect the way people work. According to Norman (1988), the design must let the user know what to do and let the user know what is going on. It is critical for an application to communicate to the user about the current state and provide feedback about the application's performance of a particular action. Norman (1988) lists seven techniques to facilitate transforming difficult and complex tasks into easier ones: use both knowledge in the world and in the head, simplify the structure of tasks, make things noticeable, get the mappings correct, exploit the power of natural and artificial constraints, design for error, and, last but not least, standardize when everything fails.

In the application, an operation is learned more readily and the problems are tracked more accurately and easily when the user has a good conceptual model. The designer develops the conceptual model, and it must be appropriate for the user by making the important parts of the operation of the device and is understandable (Card, Mackinlay, Shneiderman, 1999). The design model, the system model and the user model are part of any mental model. The design model is the abstract view of an application that the designer has in mind, The user model is the one that users develop to explain the operation of the application. The system model decides the application's physical appearance, operation and the way it reacts. The system model communicates between the user model and the design model (Dix, Finlay, Abowd and Russell, 2005). The design of the application can be simplified by using technology because it can provide help in providing an alternative option and help with implications and expose outcomes in interpretable manner. Having a good design for the application involves having certain aspects of the application automated. It will also help in keeping things visible within the application and provide responsive feedback. The design should let users recover from any errors and must assume that a user can end up with any sort of error. Error handling techniques must be included to make the user aware of the mistake and how to recover (Norman, 1988).

Shneiderman's (1996) "Principles of Human Computer Interface Design" states that the application must support diversity and cater to the varying experience of the user, ranging from novice to intermediate to experienced levels. A challenge has always been to accommodate the full range of expertise, helping novice users with common errors, as well as expert users wanting to perform the task as quickly as possible. One element of design is to strive for consistency;

consistent sequences of actions for a user should be required in similar situations (Shneiderman, 1996), e.g., identical terminology with prompts, menus and help screens. Consistency in color, layout, and capitalization and fonts should be utilized throughout the interface. Usability is also enhanced by allowing frequent users to use shortcuts in order to increase the pace of interaction though abbreviations, special keys, hidden commands and macros (Shneiderman, 1996). Providing informative feedback for the design application is always a good thing because for every user action, the system should respond in some way. For example, in web design it can be accomplished by DHTML when a button will make a clicking sound or change color when the user has clicked to show the user something has happened when it was clicked. Offering error prevention and simple error handling is important for good interaction application. For example, designing forms so that the users can’t make a serious error, and, if the user does make an error, adding simple instructions for recovery (Shneiderman, 1996; Norman 1988). A good application also allows easy reversal of actions. Reducing short term memory load is useful for interaction as well and can be accomplished by providing options that are clearly visible or using pull down menus and icons (Shneiderman, 1996; Norman 1988).

Activity centered design does not focus on the goals and preference of the user, but how users behave when performing certain tasks (Geisler, 1998). Activities can be like the actions and decisions that are done in order to accomplish a task. The activity will stop when the user terminates the application or something from the outside force terminates it. Contextual Design on the other hand is a user centered design process that incorporates ethnographic methods for getting data that is relevant to the product, such as field studies and rationalizing workflows (Shneiderman, 1996; Norman 1988). Iterative design is a methodology based on a repeated process of prototyping, testing, analyzing, and refining a product or process. Based on the results of the testing, changes and refinements are created to improve the quality and functionality of a design (Shneiderman, 1996; Norman 1988). User-centered design is a design philosophy and a process in which the needs, wants and limitations of end users are given great attention at each stage of the design process. This allows the design to be categorized as a multi-stage problem solving process that allows designers to analyze and predict how users are likely to use a product (Geisler, 1998).

2.3 Information Visualization: Background

The development of the web browser made digital information of the Internet more manageable by having graphics and sound with text. The growth of the World Wide Web demonstrates information can be more accessible by adding visualization techniques (Geisler, 1998). Visualization is just one aspect of a larger range of methods of interacting with digital information. Information visualization focuses on a graphical representation of data or concepts that either have internal construct of the mind or an external item supporting decision making. Visualization's main task is to record information, process and present it in informative ways (Geisler, 1998). Gathering data to create a visualization to have a mental picture of the data and it is important because it makes a clear picture of the data set. Seeing the data gathered in a visual representation allows the mind to better accomplish some goal as it is possible to see a representation of it. Focusing on any given goal will allow the user to address it far sooner than if the user didn't focus on it at all. Exploiting visual processing can help people explore or explain the data which is the standard argument for visualization. This is an active field of study because the design challenges are major and not fully understood (Geisler, 1998). Information visualization is based on finding a spatial mapping of data that is not inherently spatial while scientific visualization uses a spatial layout that is implicit in the data. Also, scientific visualization is often used as an expansion of the human sensory system by showing things that are on timescales too fast or slow for the eye to perceive. Visualization deals with many data abstraction techniques that can handle large amounts of data. Software-based information visualization allows interaction and representation on various kinds of abstract data, and the visual design is treated as important by embracing users' strengths as a necessary aid to effective communication (Geisler, 1998). For a person approaching a data presentation problem, he or she will often find it difficult to know where to begin, such as what tools to use or books to read. This makes it difficult to evaluate the representation used because it will be lacking the necessary background.

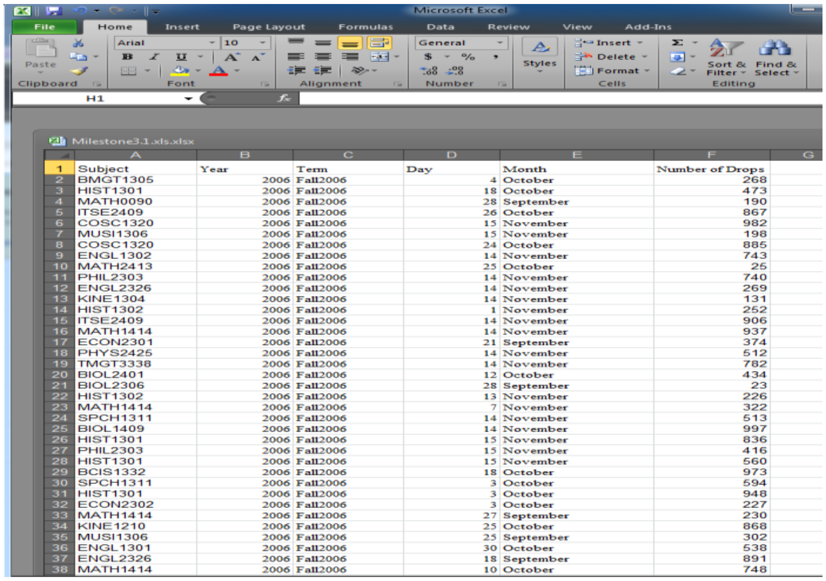
2.4 Demands for Information Visualization

The amount of information in society today is greater than ever before which makes information overload a problem because of the growth of information. Many sorts of data have been stored on computers for decades such as newspapers, magazines, catalogs, and forms which are made available to computer users (Borner, Chen, and Boyack, 2003). Society today is beginning to use higher capacity storage devices to store more information so that they can have more data at their fingertips. Therefore information filtering and sharing services are needed.

Information visualization allows people to deal with all of this information by taking advantage of the visual perception capabilities that visual representation provides. It presents information graphically allowing the human brain to understand the concept by using its perceptual system in initially processing information, rather than immediately relying entirely on the cognitive system (Card, Mackinlay, and Shneiderman, 1999). Visualization of data has the facilitates users' finding information they need in a more efficient and automatic way. The general idea of visual data investigation is to present the data in some visual form which allows the user to get insight into the data, draw conclusions and directly interact with the data provided. Visualization of data targets the integration of the user in the data investigation process directly, and then apply perceptual abilities to the large data sets available in computer systems. The main advantage of visual data investigation over automatic data techniques from statistics or machine learning is that the visual data investigation can easily deal with highly heterogeneous data and visual data investigation is intuitive (Chen, 2004).

The software system *prefuse* produces interactive visual representations of abstract data to reinforce human cognition thus enabling the user to gain knowledge about the structure of the data and causal relationships in the data (Heer, Card and Landay, 2005). This consists of the communication of abstract data through the use of interactive visual interfaces, and it involves a diverse set of disciplines ranging from computer science, graphic design, and psychology and new perspectives emerging from cognitive science. A user might change perspectives on how he or she perceives information and start to view information as being much more dynamic and reactive to the nature of the tasks, activities and relationships with others (Herman, Melancon and Marshall, 2000).

Consider a program that needs to find the number of student drops per month. The data set is huge and contains the subject, year, term, day, month and number of drops. Visualization cam make a large impact on the users. Figure 2.1 and 2.2 show how visualization of large data set can improve and retrieve results faster. The traditional way of representing the list of data is through an Excel spreadsheet as shown below.



Subject	Year	Term	Day	Month	Number of Drops
BMGT1305	2006	Fall2006	4	October	268
HIST1301	2006	Fall2006	18	October	473
MATH0090	2006	Fall2006	28	September	190
ITSE2409	2006	Fall2006	26	October	867
COSC1320	2006	Fall2006	15	November	982
MUSI1306	2006	Fall2006	15	November	198
COSC1320	2006	Fall2006	24	October	885
ENGL1302	2006	Fall2006	14	November	743
MATH2413	2006	Fall2006	25	October	25
PHIL2303	2006	Fall2006	14	November	740
ENGL2326	2006	Fall2006	14	November	269
KINE1304	2006	Fall2006	14	November	131
HIST1302	2006	Fall2006	1	November	252
ITSE2409	2006	Fall2006	14	November	906
MATH1414	2006	Fall2006	14	November	937
ECON2301	2006	Fall2006	21	September	374
PHYS2425	2006	Fall2006	14	November	512
TMGT3338	2006	Fall2006	14	November	782
BIOL2401	2006	Fall2006	12	October	434
BIOL2306	2006	Fall2006	28	September	23
HIST1302	2006	Fall2006	13	November	226
MATH1414	2006	Fall2006	7	November	322
SPCH1311	2006	Fall2006	14	November	513
BIOL1409	2006	Fall2006	14	November	997
HIST1301	2006	Fall2006	15	November	836
PHIL2303	2006	Fall2006	15	November	416
HIST1301	2006	Fall2006	15	November	560
BCIS1332	2006	Fall2006	18	October	973
SPCH1311	2006	Fall2006	3	October	594
HIST1301	2006	Fall2006	3	October	948
ECON2302	2006	Fall2006	3	October	227
MATH1414	2006	Fall2006	27	September	230
KINE1210	2006	Fall2006	25	October	868
MUSI1306	2006	Fall2006	25	September	302
ENGL1301	2006	Fall2006	30	October	538
ENGL2326	2006	Fall2006	18	September	891
MATH1414	2006	Fall2006	10	October	748

Figure 2.1 Traditional way of presenting data set

Having a large data set in a spreadsheet makes finding how many students drop per month, day, and term very tedious and frustrating for the user. Therefore, if the same data is presented in a graphical environment as show below, it helps the user visually find the information.

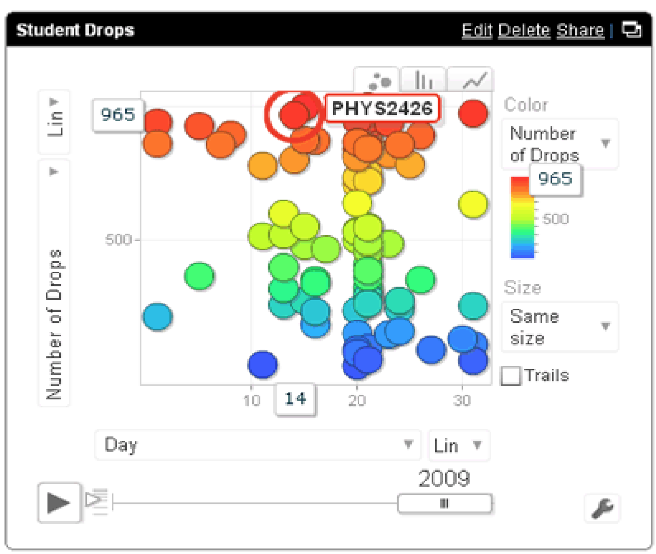


Figure 2.2 Visualizing data set improves understanding dramatically

Visualization is based on ideas from several intellectual traditions including computer science, psychology, graphic design, cartography and art, and the main influence on the field are computer graphics and human computer interaction (Borner, Chen and Boyack, 2003). How humans perceive visual information is based on cognitive and perceptual psychology. The long history of creating visual representations is based on the cartographers, and finally the artists have refined methods to make the visualizations meaningful in their disciplines.

2.5 The Steps for Data Visualization

Data visualization goes through three step process which is the overview first, zoom and filter and then details on demand which is called Information Seeking Mantra (Shneiderman, 1996). The first step is that the user needs to get an overview of the data set. Through the overview, the user identifies patterns, and the user can focus on one or more of them. Analyzing the patterns is next, the user needs to drill down to access details of the data set. Visualization techniques can be used for all the steps since it is useful for showing an overview of all the data set, allowing the user to identify interesting subsets. It is important for the user to keep the overview visualization while he or she focuses on the subset using other visualization techniques. In order to explore the interesting subsets, the user needs to get all the details of the data set to fully understand those elements (Shneiderman, 1996).

2.6 A Main Component of Visualization – Interaction

Interactivity is a main advantage of a computer based system for information versus traditional paper visualization. Interaction is used differently in many different contexts which sometimes mix with animation (Card and Mackinlay, 1997). It involves how users can interact with and control the data set representation. Principles of interaction are important when designing visualization because the basic layout rules might range form simple connections between elements in a diagram to more complicated rules that use tagged data (Card and Mackinlay, 1997). Selecting items makes is a fundamental operation when interacting with large data sets with a visual indication through which the selected set is highlighted, such as by color coding as a common. Techniques for interaction can be described in terms of what aspect of the display is changed through navigation.

2.7 User Interaction

Viewing and movement controls how the user interacts with the data. Having the representation being dynamic, makes it possible to modify the data or call attention to important features, for example, by using viewing controls to get the best camera angle or distorted view of the space from the representation data set, such as with a fisheye lens display of a large data set in a compact space. To allow data to be sampled at specific location, the user might use sampling controls which are interactive, for example, sliders can be used to set proportions or ranges for updating the dynamic data set view (Card and Mackinlay, 1997; Card, Mackinlay and Shneiderman, 1999).

Restraint of options in software is a problem since it is very easy to provide more options to a user. The most important task is to determine how to limit the number of choices that are relevant to the tasks executed. When analyzing a complex data set, it is useful to provide a default view to the user because this will be a clear functionality of the system that is running properly. An initial state makes it easy to move laterally allowing the user to modify parameters of the basic state instead of having the complicated set of knobs and sliders as the first thing that the user interacts with (Card and Mackinlay, 1997; Card, Mackinlay and Shneiderman, 1999).

The design must make an educated guess about what the user wants to do and then make the options for continuing in much more detail if needed. Scale and zoom allow choosing the relevant levels of zoom useful means for between viewpoints while maintaining context because of the animation of moving between the different levels (Card and Mackinlay, 1997; Card, Mackinlay and Shneiderman, 1999).

2.8 Visualization's Benefits

Applications containing information visualization rely on basic features whereby the human perceptual system assimilates quickly such attributes as color, size, shape, and motion (Schulz and Schumann, 2006). A good visualization enables the user to not only perceive information more easily, but also to perceive more information at one time. The user is able to see patterns in the data which indicate trends, recognizing gaps in the data, discovering outliers or errors in the data, pinpointing minimum/maximum values and identifing any clusters. Visualization applications allow the users to better understand complex systems, which allows the user to make better decisions and discover information that might be unknown (Herman, Melancon and Marshall, 2000).

Information visualization helps users with different backgrounds and knowledge investigate large amounts of data faster and gain significant insights (Healey, Booth and Enns, 1993). *Prefuse* techniques and applications are designed to enable a wide variety of computer users to more easily navigate information spaces providing better display information and to improve their understanding of information. (Heer, Card and Landay 2005). Vision is how people primarily perceive information, and the display of digital information is likely to continue to be designed for visual perception.

Human cognitive capabilities for visualization of information rely on six basic elements (Chi, 2000): increasing cognitive resources, which expands human working memory; reducing search which gets a large amount of data into a small place; enhancing the recognition of patterns which information is organized in space by its time relationships; supporting the easy perceptual implication of relationships that are more difficult to make; enabling perceptual monitoring of huge number of potential events; provide a controlling average that enables the exploration of a space of parameter values.

The emerging field of information visualization has created many interesting examples of how information can be made more accessible through visual representation (Eick, 1996). Many information visualization tools have appeared in the research community and have been produced in the marketplace because information visualization is applied in many areas where understating of the basic structure in data is fundamental (Herman, Melancon and Marshall, 2000). Some are: document-oriented interfaces such as text editing, document formatting, illustrators and spreadsheets; communications oriented interfaces such as email, computer conferencing, telephone and voice messaging systems; design environments such as programming environments; online tutorial systems; multimedia information kiosks and embedded systems such as copier controls, elevator controls and home appliance controllers.

Visualization is used by businesses have made a series of contribution for many tools and projects (North, 2005). It enhances the visual look of data, tools and projects, such as those listed below:

- ManyEyes –a pioneer in the field of online visualization software that facilitates data visualization without the need for programming.

- Tableau Software – infinite possibilities for researchers, corporations to interact, explore and manipulate the data. It has interactive visualization online as opposed to static images.
- Verifiable – online website that allows online visualization of data, also visualizations are through bar charts, scatter plots and line charts.
- Wordle – a tool that creates a cloud of words and highlights the ones which have a large repeat value. The higher the repetition of the word, the bigger the word appears in the section.
- Swivel – similar to Verifiable where the user can upload the data and create online interactive visualizations.
- Google API – just like ManyEyes is a website that online visualization software that facilitates data visualization without the need for programming.
- MarketMap – tool on the website SmartMoney that tracks the rise and fall of stocks over a period of time using treemaps.

2.9 The Approach to Visualizing Graphs

An important area in computer science is graph drawing. Graphs can provide insights for information presented to users because it can show objects and relations among them (Ware, Purchase, Colpoys and McGill, 2002). There are semi-automatic graph drawing tools that provide drawings with the output this kind of tool can be improved manually. Graphs usually have application specific semantics known to the users working with the graph, but not known to the graph drawing algorithms.

The visual representations of graphs and networks are external representations that augment human visual processing to diminish the cognitive load of many tasks that require understanding of global/local structure (Eick, 1996). Computational manipulation of large graphs is common, though graph visualization is usually limited to datasets of few thousands of nodes. A graph can be described as G = (V, E), where a set, V, of vertices and a set, E, of edges describe edges as a pair of vertices. Usually a graph is described with its vertices as points in a plane and the edges as line or curve segments connecting those points (Eick, 1996).

Network visualization and graph drawing has focused on improving the visual organization of nodes and links according to principles of graph drawing. Sophisticated algorithms have addressed issues that deal with graph positioning and layout, and which depend on scaling on the data set provided and navigation within the graph layout that provides interactivity between the graph elements and the user (Munzner, 1997). Graphs to meet aesthetic criteria typically have uniformly distributed vertices in the display. Also, minimizing edge crossings, making edge lengths uniform and reflecting natural symmetry assists graphs in meeting the criteria.

Multidimensional data can be visualized by networks with nodes pertaining to data elements and links representing relationships between them. The conventional node and link presentations are beneficial for visualization of small sparse networks, but for larger networks visualization encounters three main problems. The first issue is clutter of the displays, positioning of nodes to permit a meaningful interpretation and encoding additional information are problems to be addressed in effective network visualizations (Eick, 1996). Another issue is that limited viewing area of display is why the clutter problem occurs in visualization of multidimensional large sets. The last issue is the conflict between showing low details and high level context information simultaneously which users need to access both (Munzner, 1997). Graphs can consider both overview and detail in display through different views. There might be a zoomable view which shows the objects in many different scales, but a lot of global information may be lost (Gazner, Koren and North, 2004). Another view would be the focus and context which gets local detail and global context in integrated views, for example fisheyes views. The image below is an example of fisheye view of a network (Furnas, 1986).

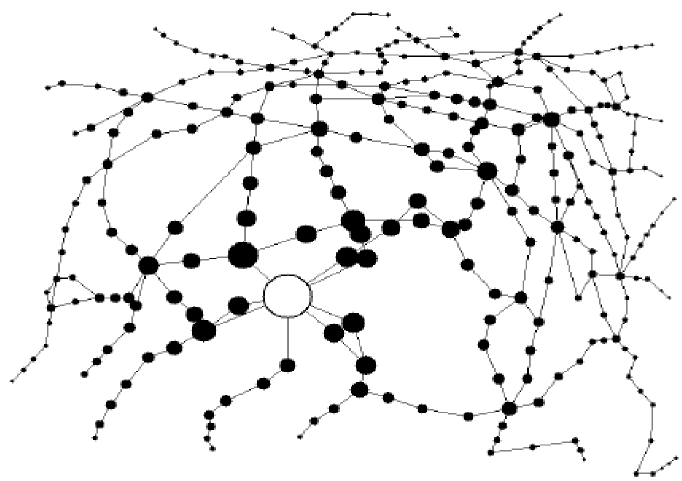


Figure 2.3 Fisheye example

A system that that visualizes a clustered graph using hierarchical grouping and summarization is HiMap (Shi, Cao, Shixia, Qian, Tan, Wang, Sun and Lin, 2009). HiMap is able to use data analysis techniques to accurately control the visual density of each graph view. It avoids visual clutter by having an optimized layout algorithm and allows the user to easily navigate the network with smooth and clear view transitions. The effectiveness of HiMap algorithms is displayed through graph-traversal based evaluations.

The Internet has been perhaps the most exciting application to use social networking. A large variety of applications has been integrated into the online social network. To be able to present a hierarchical view of the social network is useful because it allows the online user to comprehend clearly his exact role and build mental impression about the entire network (Shi, Cao, Shixia, Qian, Tan, Wang, Sun and Lin, 2009). Then, it provides rich material for the social network operators to investigate the network further. Other systems such as Vizster, MatrixExplorer and NodeTrix are adapted to visualizations that tile nodes that represent the social network users in one view. With HiMap the software assumes that the entire social network may be millions of nodes. It must define each graph view of the network which should be adaptively visualized in a readable manner that is easy to be comprehended independent with its scale, topology and the screen size to display. Also, a suite of navigation methods should be provided so that it is capable to visualize and diagnose every detail of the network. The system should also provide smooth animations that will be presented when any view changes. Finally the system should run fast enough and keep lightweight because it could catch up with the animation speed and load social network data incrementally and on-demand. HiMap uses clustered node-link graph, the layout algorithm avoids cluster overlapping and in order to adapt each visualization view with changing screen size with novel adaptive data loading method. The system HiMap employs the adaptive data loading method to preserve the readability of the graph view. Interactions with the system include; selecting, dragging, zooming, panning and two classes of edge bundling methods. The HiMap overview design includes the visualization pipeline, which the data manipulates methods and the various visual metaphors tailored for presenting social networks. The offline data manipulation, which is collecting, cleaning and hierarchical clustering, prepares the graph data required for further on line visualization of the target social networks. On the other hand, the adaptive visualization part involves the data loading, graph layout, projection and rendering stages.

CodeSaw is a social visualization software development system for the distributed software community (Gilbert and Karahalios, 2007). The software visualizes two different perspectives, code repositories and project communication. Open source software development differs from traditional development by lacking the face to face interactions, not having any plan, and having minimum schedule. CodeSaw differs from other project visualization software. SeeSys (Gilbert and Karahalios, 2007) uses treemap visualization of the source code file system, but it is limited to large projects with relatively stable contents. Auger uses line-oriented style presenting one user. CodeSaw is different from both by abstracting the information of one year of a project, showing all people involved and having a spatial messaging (Gilbert and Karahalios, 2007). The software is based on the result of the initial interviews is which the developers of the open source community, who feel disconnected from the rest and use project mailing list to maintain awareness. CodeSaw shows only up to eight developers in comparison to the others. It doesn't consider other people who help the development by sending feedbacks and patches, reporting bugs or solving special problems with specific part of the code with limited one time writing a vital code. I like the fact that the research focuses on the share of the developers in the development and the project itself. The factor for measuring the participation if the raw number of lines of written code, which in my opinion this doesn't reflect the skill of the developer and the real share of development (Gilbert and Karahalios, 2007). For example, more skillful programmers use smaller but more complex algorithms to solve the same problems and this way their overall share of their contribution would not be reported correctly. The amount of contribution strongly depends on the part they are developing. Someone might solve an unsolved problem in development by spending much time but the result could be developed in only a few lines of code. I feel that more advanced research could be done on finding the best measurement factor of the share of each user in overall development than just using the number of code lines. It's very interesting to see how the system is capable of extracting social information from the visualization results. For example, the fact that one contributor only writes code in the summer can be used to guess that he is a student with more free time in the summer. This information could be used in findings who are the main developers of the open source software projects and how can the project coordinators help them by knowing the factors. It is considerate of the authors to use the frequency of appearance of words in the emails to visualize the communications; I feel this is the best way to minimize the large amount of communication data into specific key words (Gilbert and Karahalios, 2007).

One of the main things about the research is having small data sets as the solution for visualizing, but this would limit the application of the system into limited time range and groups with small users. Further studies should be taking into consideration for expanding the system into more crowded groups and over large time period. One solution is using the same technique but replying on grouping of users rather than focusing on each individual. For example, in a large project with multiple participants, instead of focusing on each person, people could be grouped by the area they are working. The research was successful in achieving good balance between revealing private information to the world and leaving enough of it ambiguous so that it does not invade privacy (Gilbert and Karahalios, 2007). Adding spatial messaging which allow the users to leave comments on the visualization itself was very considerate for the authors. Generating automatic mail to the mailing lists regarding the spatial data was also very strong solution for not introducing a parallel communication tool. In summary, CodeSaw is an innovative tool to encourage developers to participate more in open source software development by showing their participation share in the overall project.

CHAPTER III

METHODOLOGY AND FINDINGS

Applications that utilize information visualization depend in part on simple features that the human perceptual system naturally extracts instantly, like color, size, shape, nearness and movement (Norman, 1988). Good visualizations allow the users to perceive information easily, as well as more information at one time. A user can quickly see patterns in data that might consist of a trend, gaps in the data, errors in the data, maximum and minimum values, and clusters. Information visualization can allow users to better understand complex system and make better decisions (Norman, 1988). In particular, visualization has been successfully used for network based objects. It is common practice to use visualization to data sets that are represented as graphs or networks which show the different relationships. Social networks are an example where visualization can be developed to make information presentations that facilitate understanding. Every user can be represented as a node in the graph, and the edges between the nodes show the immediate connection that each user has with the others. In this thesis, we create useful tools for better managing and monitoring the South Texas College Distance Education department as a whole. This chapter discusses the concepts for visualizing the Distance Education online courses and the features to enhance performance tracking.

The application is developed using the visualization framework for three graph representations: tree view, graph view and radial graph view supplied by the software system *prefuse* (Heer, Card and Landay, 2005). The *prefuse* visualization process starts with abstract data to visualize. *Prefuse* provides extensible interfaces for input and output of this data and includes support for incremental loading and caching from a database. Renderers are components that use the visual attributes of an item to determine screen appearance. Renderers have a simple API consisting of three methods: one to draw an item, one to return a bounding box for an item and one to indicate if a given point is contained within a item, as well as renderers for drawing basic shapes, straight and curved edges, text and images. I/O libraries are provided to read and write data seat. Filtering is a process of applying rules to the abstract data to convert it to representation appropriate to be visualized. Once the filtering is completed for the data set, a visual item is created for every item of information in the data set. Rendering the visual data and is composed of interactive display that unites the elements. The user interacts with the display unit using the basic user interface controls.

3.1 Design

Tree View

The application was mainly to be used by professors to maintain and track their online courses and to track the enrollment of a specific section. The tool produces a graph which contains Distance Education as a central node and all the online courses with all their sections together with list of certified online instructors connected to this central node. *Prefuse* provides a tree view graph layout. By clicking on each node, the node expands as a sparse tree for easier data interaction. Every course node is associated with enrollment and section number which provides the professor easy access to view the courses by subject, section number and number of enrollments. A professor might also want to view the faculty members that are certified to fully teach online, and this view can be provided. *Prefuse* also provides searching facilities based on title of courses or faculty names associated with Distance Education. The search engine looks up titles and names by letter being typed and matches these with any available data, highlighting with color for better visibility. If a match is found, then the node that is being looked up is highlighted.

Graph View

Prefuse's graph view is mainly used by users to keep track of faculty depending on the subject they teach, whether they are teaching more than one subject and whether they have an office. The tool produces a graph which contains a sparse tree with an undirected layout. This setup gives a graph view layout for faculty members that can teach online. By clicking on each name on the graph, *prefuse* makes the relationship the name might have which is either by the same subject or if they are teaching more than one subject and if they have an office. Every node is associated with the subject, subjects taught and office number. Graph view uses force directed layout to position nodes and provides users animated tools.

Radial Graph View

Prefuse's radial graph is similar to graph view and is mainly used to keep track of faculty ordered by the subject they teach, whether they are teaching more than one subject and whether they have an office. The tool produces a graph which a single node is displayed at the center of the display and all the other nodes are positioned around it. The whole graph is like a tree rooted at the center node. This center node is referred to as the focus node, and all the other nodes are arranged around it. The nodes are like a ring corresponding to the shortest network distance from the focus, which is the middle. Any nodes that are joined by an edge in the graph as known as its neighbors, and the immediate neighbors in the middle fall on the smallest inner ring, their neighbors fall on the second smallest ring, and so forth.

3.2 Layout and Basic Interaction

Tree View

The Tree View layout shown on Figure 3.1. The algorithm used to draw the graph is from Ka-Ping Yee, Danyel Fisher, Rachna Dhamija and Marti Hearst and computes a tree layout that factors in possible variation in sizes and maintains both orientation and ordering constraints for easier and more understandable transitions between changes in display.

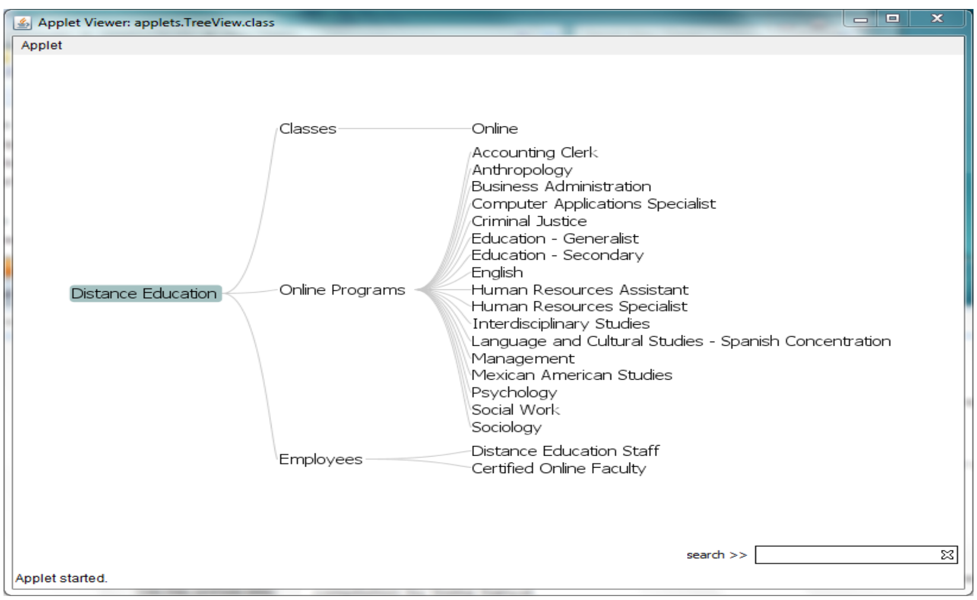


Figure 3.1 Tree View layout – visualization of Distance Education

Prefuse has basic interaction with simple mouse operations. Clicking a node causes the corresponding data to expand for viewing of the information in the section. The zoom feature is particularly effective for visualization in this display type. Using the scroll button on the mouse will either zoom in or out of the tree. This will make the text larger as the user is zooming in, and with zooming out the text will get smaller for viewing the entire tree. Also, when the user right clicks on the application, it will center the tree view which will make everything on the tree fit to the screen. Figure 3.2 shows the zooming in and out of the application.

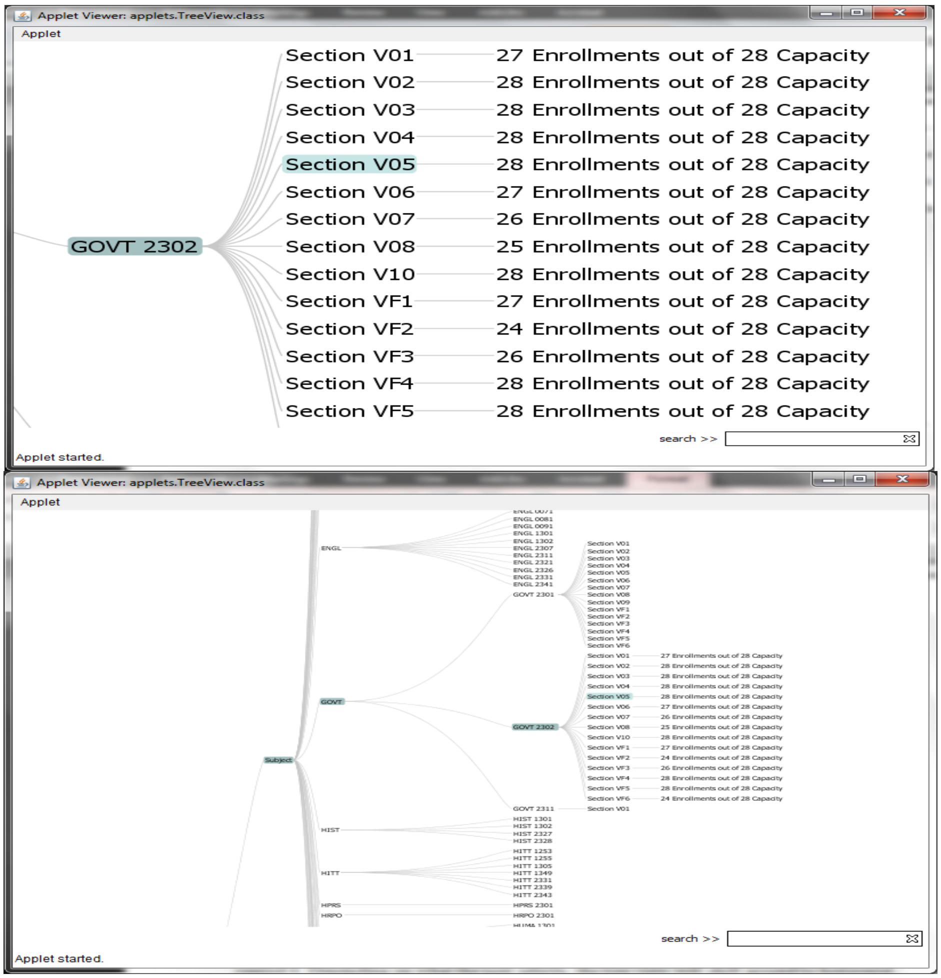


Figure 3.2 Zoom in and out

Other views can be obtained by using control 1, control 2, control 3 and control 4. Depending on what the user selects, the tree view will shift around to horizontal, vertical, and backwards. The tree view display also supports keyword search of the visualized network, with color coding for targets. The *prefuse* application is a desktop application that can be used by the professors to keep track of the enrollment in each online course. It is an attempt to visualize the enrollments of each online course and build sparse trees depending on information needed such as certified online instructors that can teach online courses.

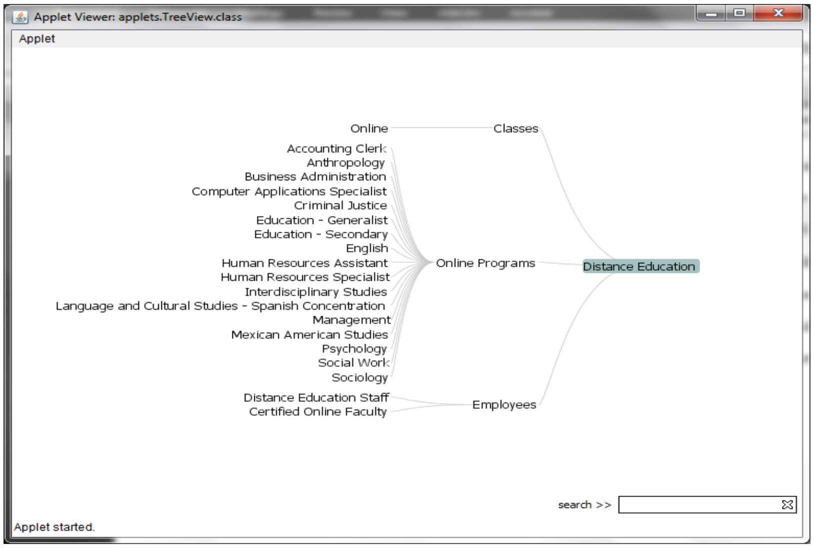


Figure 3.3 Tree view: different control view

Graph View

Prefuse's graph view is shown in Figure 3.4. Again, basic interaction is provided through mouse operations. Clicking a node causes shows the relationship of the node by either subject, teaching more than one subject or if they have an office.

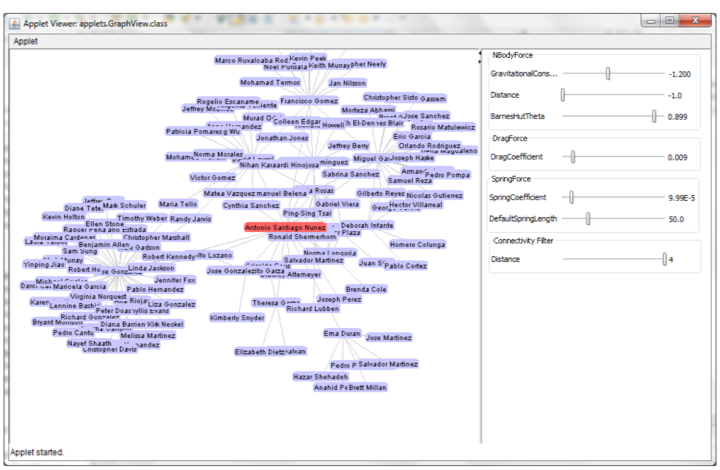


Figure 3.4 Graph view layout

Graph view displays thousands of nodes with edges between them. It can provide a visualization of large number of nodes in a single view and provides interactive exploration for the user through the graph. The nodes are positioned by force directed layout, manipulable by the controls on the right of the display. It provides effective way to explore large graphs by focusing on the selected node. Figure 3.5, shows the drag function. A user clicks on the professor's name and drags it to be able to show the professors that teach the same subject as the node selected or whether other professors teach more than one subject.

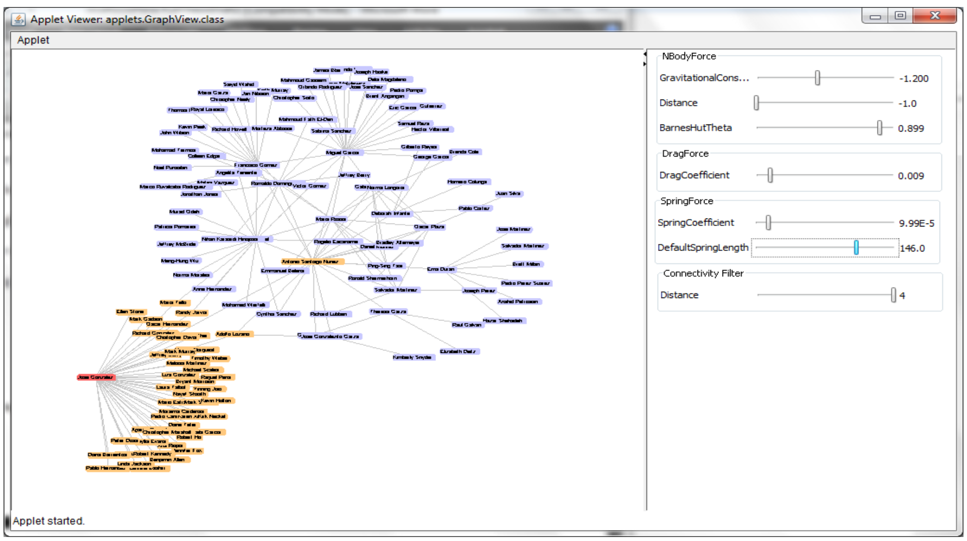


Figure 3.5 Graph view dragging a node for interaction

Graph view also allows zooming in and out to the user. The scroll button on the mouse will either zoom in or out of the graph view. It will make the text bigger as the user is zooming in the application and with the zoom out the text will get smaller. If the user right clicks on the screen it will make the whole graph fit to the window. Figure 3.6 shows the zooming in and out of the graph view application.

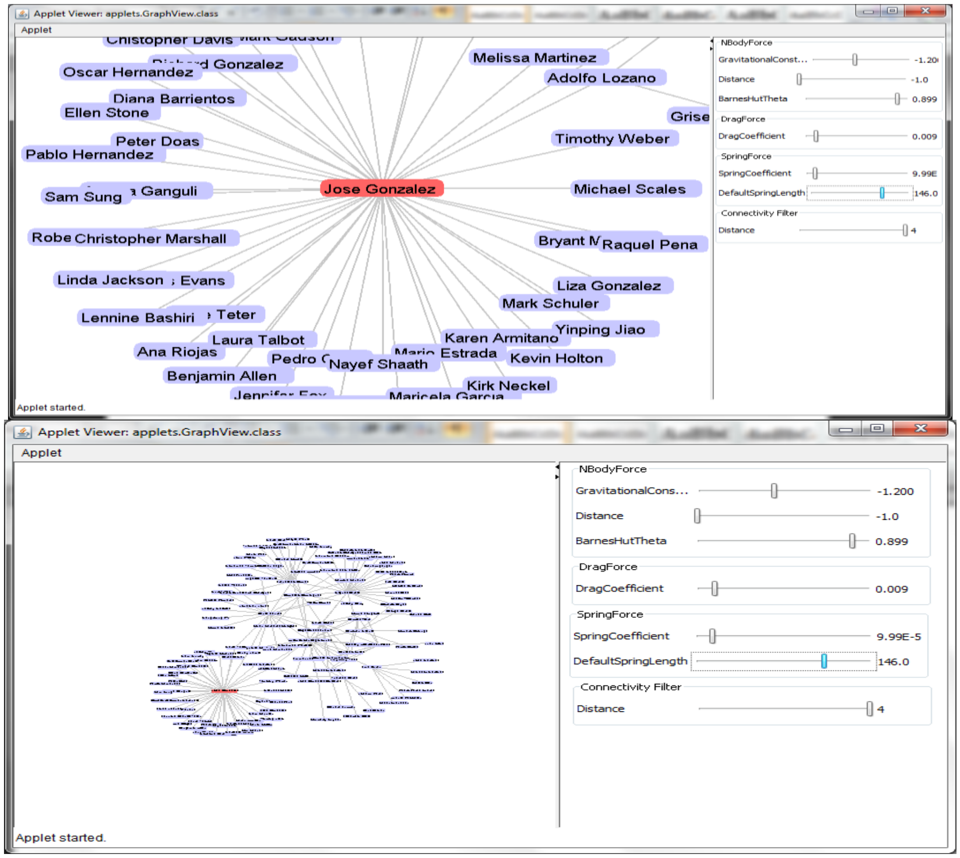


Figure 3.6 Graph view zooms in and out

The disadvantage of the graph view is that it is clustered. The professor's name includes first name and last name which makes the graph overlap. It makes it hard for the user to clearly view the information because of the overlapping names. The graph view also contains extra features that allow the user to expand the length between nodes as shown in figure 3.7.

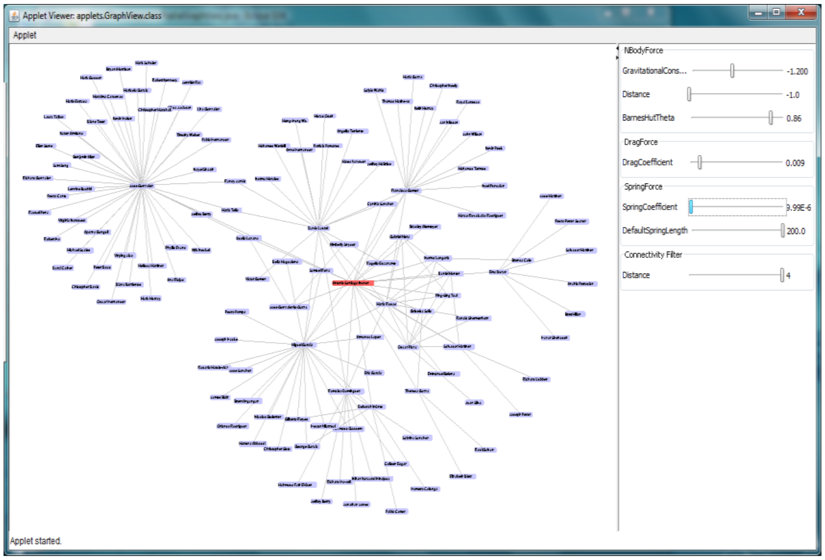


Figure 3.7 Graph view controlling the length between nodes

Radial Graph View

Prefuse's radial graph view is shown on Figure 3.8. As with the other display types, licking on the faculty's name, highlights the name and shows the relationship between the faculty instructors making the selected name the central node. As different professor's names are selected the graph keeps moving in a circle formation. The scroll button on the mouse will either zoom in or out of the radial graph view. IT will make the text bigger as the user is zooming in the application and with the zoom out the text will get smaller for viewing the entire graph.

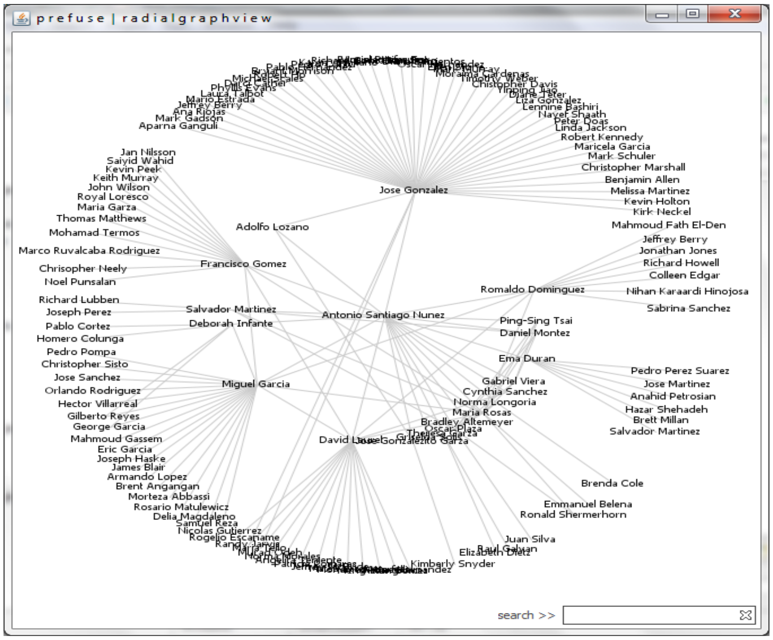


Figure 3.8 Radial graph view layout

When a user right clicks on the application, it will center the tree view which will make everything on the graph fit to the screen. Figure 3.9, shows how the drag function is being used. A user just clicks on the professor's name and drag it to be able to view the relationship the professor has making long lines of relation clearly visible.

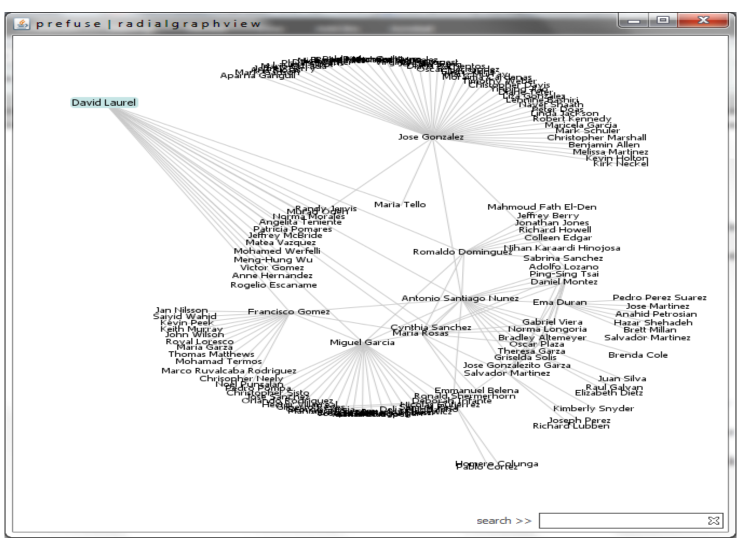


Figure 3.9 Radial graph view layout dragging a node for interaction

The search function is very useful with the radial graph view because it contains a large set of information and the clutter is difficult to overcome. Finding a professor's name with the search function makes it easy to the user to find the desire name as oppose to looking in the crowded graph for the name. Figure 3.10 displays an example of how the search function works on the radial graph view.

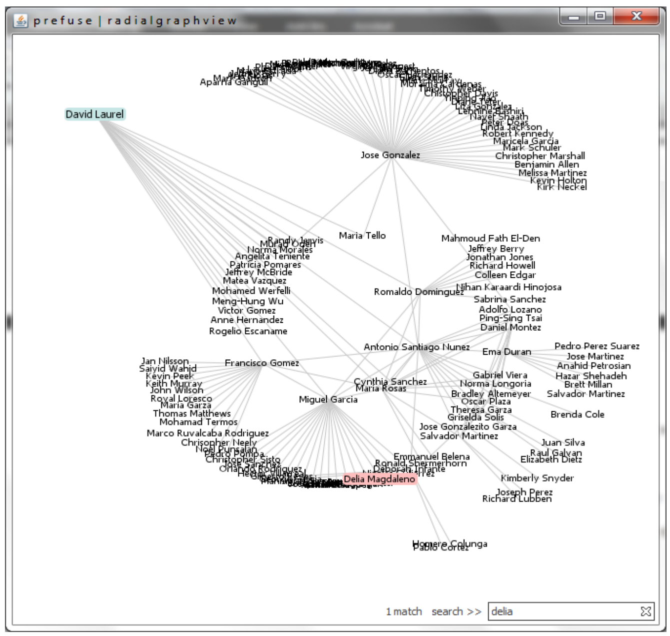


Figure 3.10 Radial graph view search function

3.3 Implementation

Prefuse's tree view, graph view and radial graph view are implemented in a Java based application that can be treated as a standalone desktop application. Reading the graph using Eclipse IDE and *prefuse*, after the graph data is declared in the XML file then the graph is read into the visualization environment using the tree view class, graph view class and radial graph view class that are provided by the *prefuse* framework. A visualization object is created to accommodate this graph data structure. When the graph is constructed, several rules and conditions can be added to improve the visualization on the graph. The search functionality is extra in which it highlights the node that is searched in the search box. The rules add multiple color schemes to the nodes and the fonts of the visual display of the graph can be modified using the rules. By having the Renderer class which is provided by the *prefuse* framework, allows the graph visualization to render the graph which contain all the rules necessary by the application.

Tree View

The main data structures uses a XML file to create the tree to expand to the necessary items. It stores information about the tree, branches and leaves. The graph schema declaration describes the layout of the graph. An XML sample is shown below in Figure 3.11:

```xml
<tree>
 <declarations>
   <attributeDecl name="name" type="String"/>
 </declarations>
 <branch>
  <attribute name="name" value="Distance Education"/>
  <branch>
   <attribute name="name" value="Classes"/>
   <branch>
    <attribute name="name" value="Online"/>
    <branch>
     <attribute name="name" value="Subject"/>
     <branch>
      <attribute name="name" value="ACCT"/>
      <branch>
       <attribute name="name" value="ACCT 2401"/>
       <leaf>
        <attribute name="name" value="Section V01"/>
        <leaf>
        <attribute name="name" value="27 Enrollments out of 28 Capacity"/>
       </leaf>
       </leaf>
       <leaf>
        <attribute name="name" value="Section V02"/>
        <leaf>
        <attribute name="name" value="27 Enrollments out of 28 Capacity"/>
       </leaf>
       </leaf>
       <leaf>
        <attribute name="name" value="Section VF1"/>
```

```xml
       <leaf>
        <attribute name="name" value="Daniel Montez"/>
       </leaf>
       <leaf>
        <attribute name="name" value="Gabriel Viera"/>
       </leaf>
       <leaf>
        <attribute name="name" value="Bradley Altemeyer"/>
       </leaf>
       <leaf>
        <attribute name="name" value="Oscar Plaza"/>
       </leaf>
       <leaf>
        <attribute name="name" value="Benito Garza"/>
       </leaf>
       <leaf>
        <attribute name="name" value="Brenda Cole"/>
       </leaf>
      </branch>
      <branch>
       <attribute name="name" value="VNSG"/>
       <leaf>
        <attribute name="name" value="Antonio Santiago Nunez"/>
       </leaf>
      </branch>
     </branch>
     </branch>
   </branch>
 </tree>
```

Figure 3.11 XML input file of Distance Education tree view graph

For the example the XML data set has a <tree></tree> statement, inside those statement the branches are composed along with each leaf on the branch. The branch is label as

<branch></branch> and its corresponding leaf is <leaf></leaf> which makes it pretty basic to extend the content to whatever the user needs.

Graph View and Radial Graph View

The main data structures for the graph view and radial graph also utilize XML file to create the graphs. Information is stored about the graph, keys, nodes and edges. The graph schema declaration explains the overall layout of the graph and the entry of XML would look like the sample shown below in Figure 3.12:

```
<!-- data schema -->
<key id="name" for="node" attr.name="name" attr.type="string"/>
<key id="gender" for="node" attr.name="gender" attr.type="string"/>

<!-- nodes -->
<node id="1">
  <data key="name">Antonio Santiago Nunez</data>
  <data key="gender">M</data>
  </node>
<node id="2">
  <data key="name">Ping-Sing Tsai</data>
  <data key="gender">M</data>
</node>
<node id="3">
  <data key="name">Daniel Montez</data>
  <data key="gender">M</data>
</node>
<node id="4">
  <data key="name">Ema Duran</data>
  <data key="gender">F</data>
</node>
<node id="5">
  <data key="name">Gabriel Viera</data>
  <data key="gender">M</data>
</node>
```

```
<edge source="20" target="117"></edge>
<edge source="20" target="118"></edge>
<edge source="20" target="119"></edge>
<edge source="20" target="120"></edge>
<edge source="20" target="121"></edge>
<edge source="20" target="122"></edge>
<!-- Faculty with no offices -->
<edge source="21" target="123"></edge>
<edge source="21" target="124"></edge>
<edge source="21" target="125"></edge>
<edge source="21" target="126"></edge>
<edge source="21" target="127"></edge>
<edge source="21" target="128"></edge>
<edge source="21" target="129"></edge>

</graph>
</graphml>
```

Figure 3.12 XML input file for graph view and radial graph view

The XML data set has a <graphml> statement, inside the graph, key, nodes and edges. The graph data scheme <key> statement, follow by the nodes <node >, then the edges <edge>.

3.4 Evaluation

Tasks	Tree View	Graph View	Radial Graph View
Find total number of enrollments in a specific department	Yes	No	No
Find certified faculty members from each department	Yes	No	No
Find courses that have reach maximum enrollments	Yes	No	No
Faculty that teach more than one subject	Yes	Yes	Yes
Find faculty that have an office	No	Yes	Yes

Figure 3.13 Graphs evaluation

Tree View

The *prefuse* tree view display for the Distance Education data was developed to investigate the use of graphs for basic display of information. The effort has been made to keep the application simple and display more of its visualization capabilities, rather than backend functionalities. I presented screenshots of the application which show the capabilities of the tool and its graph drawing capabilities. The different scenarios are created and the graph drawing of the tool is evaluated in this section. The first scenario is if a faculty dean wants to know the exact number of enrollments for the online courses depending on the subject (Figure 3.14).

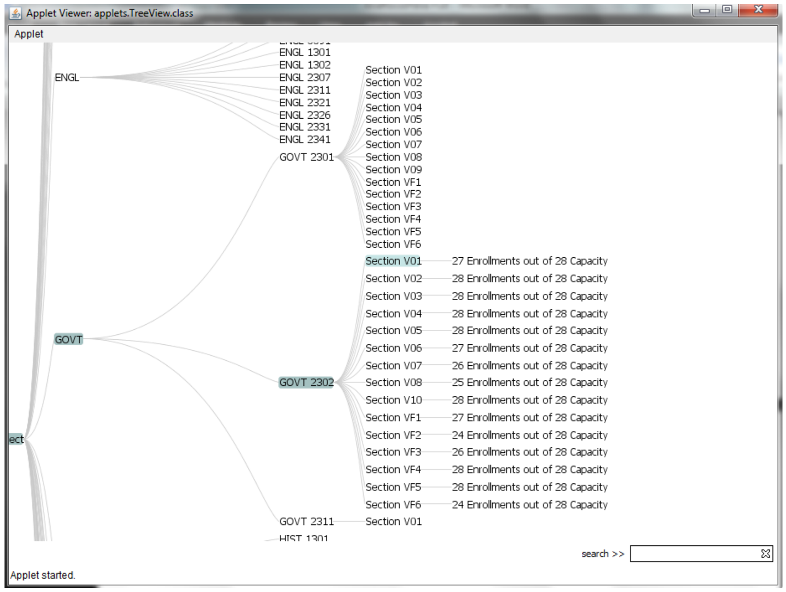


Figure 3.14 Tree view with number of enrollments by subject.

As shown in Figure 3.14, the user is searching for the government class, and once it is found views the number of Government course available. Clicking on the course name will display the number of sections offered online, as well as the enrollment for each section.

Another scenario displays instructors certified to teach online. This is important for faculty members because if they are not certified they aren't able to teach online courses. Having this application facilitates the process of going through a large text file without knowing whether the user is teaching a subject. This application groups the faculty members by subjects to display the available faculty members certified to teach online courses.

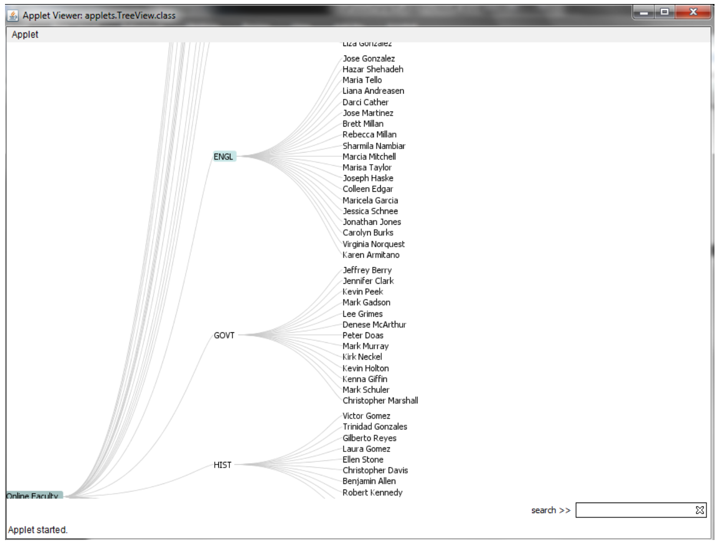


Figure 3.15 Tree view with certified online faculty by subject

Graph View

The *prefuse* graph view of Distance Education was developed to encourage the use of graphs for basic display of information. The first scenario is if a user can see that the select faculty member teaches the same subject, teach more than one subject or have an office making it a relationship between key, node and edges (Figure 3.16).

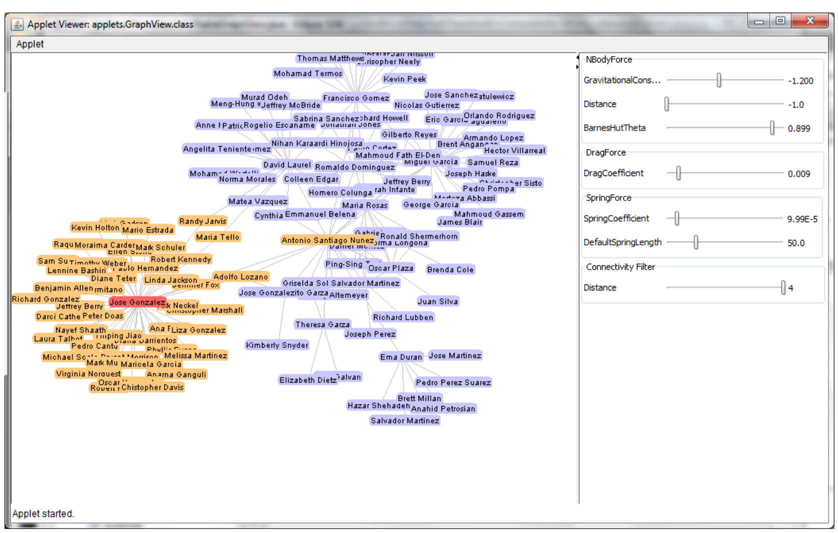


Figure 3.16 Graph view showing faculty by subject

The disadvantage of the graph view is that all the instructors' names are clustered with each other which makes it hard to read. Everything can be spread around, and using the drag option will allow the user to get a better picture of who are the professors that link to the name. The relationship between each node consist of the subject teacht by instructor, and its corresponding edges depend on if the faculty member has a office or is teaching more than one subject.

Radial Graph View

The *prefuse* radial graph view application of Distance Education was developed to encourage the use of graphs for basic display of information just like the graph view. The first scenario is if a user can clearly see that the select faculty member that makes it the central node of the graph and it tells the user viewing the graph, who are the names that teach the same subject, teach more than one subject or have an office making it a relationship between key, node and edges (Figure 3.17). As the edges appear farther away from the central node it means the distance increases between the faculty members.

The disadvantages of the radial graph view are similar to the graph view because it's very clustered as well and so is hard to read. The radial graph view has limitations on how many nodes and edges are allowed before it starts being cluttered, as shown in figure 3.17.

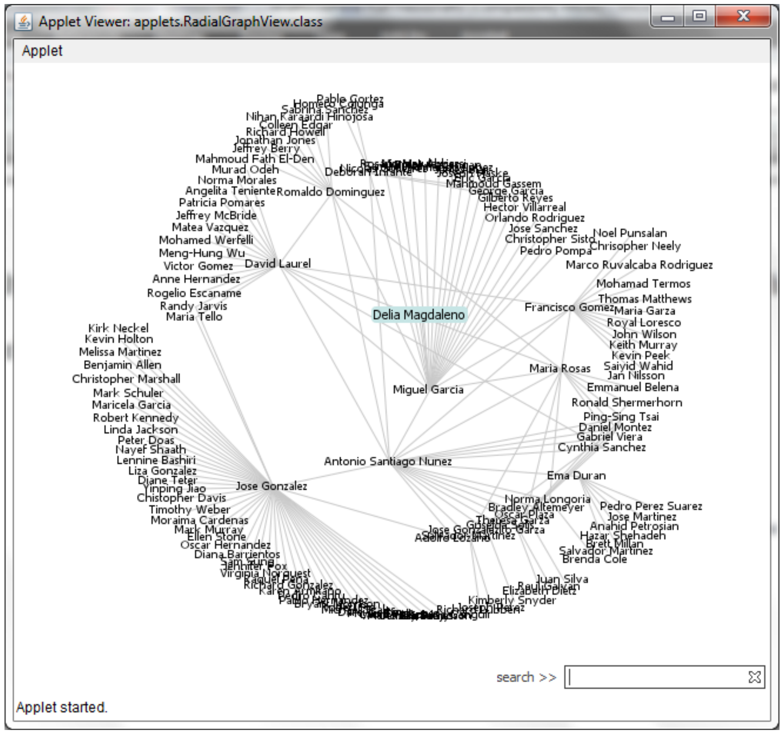


Figure 3.17 Radial graph view central node Delia Magdaleno

CHAPTER IV

SUMMARY AND CONCLUSION

The main goal for this thesis was to investigate the use of visualization for a large data set for the South Texas College Distance Education department. The thesis provided users a graph visualization that utilized human computer interaction design principles in the *prefuse* graph display application.

Information visualization helps users with a variety of backgrounds and knowledge analyze large data sets faster to gain relevant insights. The application enabled users to look at large amounts of complex data and quickly find the information they want. They could navigate and interact with quickly to easily recognize patterns and trends and so obtain a better understanding of the information in the data set. Usability played a major role because users are often more successful when working with visualizations, as opposed to using large data sets from text.

There are many trends in computing that will help the development of information visualization applications. Computer users will more likely accept applications that are possible with the decreasing in price of multimedia capable personal computers, together with increased processing power, bandwidth and speed.

The demand to deal with increasingly large amounts of information has prompted designers to incorporate more graphical information in. Applications that succeed with their wider availability of tools for creating graphical information have lead to more widespread use of information visualization. *Prefuse*'s support for component reuse and extensibility using provided modules such as zooming filters, different layout, renderers and interactive controls throughout visualization. *Prefuse'*s customizable rendering and animation make it fast to implement and explore various design ideas.

REFERENCES

Borner, K., Chen, C., & Boyack, K. (2003). Visualizing knowledge domains. *Annual Review of Information Science and Technology*, *37*, 179-255.

Card, S. K., & Mackinlay, J. (1997). The structure of the information visualization design space. *IEEE Symposium on Information Visualization*, 92-99.

Card, S. K., Mackinlay, J., & Shneiderman, B. (1999). *Readings in information visualization: Using vision to think*. (pp. 1-35). Morgan Kaufman.

Chen, C. (2004). Information visualization: Beyond the horizon. Verlag, London: Springer.

Chi, E. (2000). A taxonomy of visualization techniques using the data state reference model. *Proceedings of InfoVis '00*, 69-75.

Cockburn, A., Karlson, A., & Bederson, B. B. (2008). A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. *ACM Computing Surveys*, *41*(1), 1-31.

DeFanti, T. A., Brown, M. D., & McCormick, B. H. (1989). Visualization - expanding scientific and engineering research opportunities. *IEEE Computer*, *22*(8), 12-25.

Dix, A., Finlay, J., Abowd, G., & Beale, R. (2005, February 13). Human-computer interaction. Retrieved from http://cognac.ai.ru.nl/studie/MMI_summary.pdf

Eick, S. G. (1996). Aspects of network visualization. IEEE Computer Graphics and Applications, 16(2), 69-72.

Furnas, G. (1986). Generalized fisheye views. *CHI '86.*

Gazner, E. R., Koren, Y., & North, S. (2004). Topological fisheye views for visualizing large graphs. *IEEE Transactions on Visualization and Computer Graphics*, 1-12.

Geisler, G. (1998). Making information more accessible: A survey of information visualization applications and techniques.

Gilbert, E., & Karahalios, K. (2007). Codesaw: A social visualization of distributed software development. *C. Baranauskas et al. (Eds.): INTERACT 2007, LNCS 4663, Part II*, 303-316.

Healey, C. G., Booth, K. S., & Enns, J. T. (1993). Harnessing preattentive processes for multivariate data visualization. *Proceedings Graphic Interfaces.*

Heer, J. M., Card, S. K., & Landay, J. A. (2005). prefuse: a toolkit for interactive information visualization. *Proceeding of ACM CHI '05*, 421-430.

Henry, N., & Fekete, J. (2007). Matlink: Enhanced matrix visualization for analyzing social networks. *C. Baranauskas et al. (Eds.): INTERACT 2007, LNCS 4663, Part II*, 288-302.

Herman, I., Melancon, G., & Marshall, M. S. (2000). Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, *6*(1), 24-43.

Keahey, T. A. (1998). The generalized detail-in-context problem. *Proceedings of IEEE Visualization '98.*

Keahey, T. A., & Robertson, E. L. (1996). Techniques for non-linear magnification transformations. *Proceedings of IEEE Visualization '96.*

Mazue, A., Doran, C., & Rice, P. (2002). The use of social network analysis software to analyze communication patterns and interaction in online collaborative environments. *Proceedings of the American Society for Information*, *39*(1), 566-567.

Munzner, T. (1997). H3: Laying out large directed graphs in 3d hyperbolic space. *IEEE Information Visualization.*

Munzner, T. (1998). Exploring large graphs in 3d hyperbolic space. *Computer Graphics and Applications*, *18*(4), 18-23.

Norman, D. (1988). The design of everyday things. New York, NY: Doubleday.

North, C. (2005). Information visualization in handbook of human factors and ergonomics. *G. Salvendy (editor)*, 1222-1246.

Paulovich, F. V., & Minghim, R. (2006). Text map explorer: a tool to create and explore document maps. *Proceeding of IEEE Information Visualization.*

Schulz, H., & Schumann, H. (2006). Visualizing graphs - a generalized view. *Proceedings of IEEE Information Visualization.*

Shi, L., Cao, N., Shixia, L., Qian, W., Tan, L., Wang, G., Sun, J., & Lin, C. (2009). Himap: Adaptive visualization of large-scale online social networks. *IEEE Pacific Visualization Symposium 2009 April 20 – 23.*

Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. *IEEE Proceedings of Visual Languages*, 336-343.

Shneiderman, B., & Plaisant, C. (2010). *Designing the user interface, strategies for effective human-computer interaction*. (5th Ed. ed., pp. 537-562). Addison-Wesley Longman.

Tory, M., & Moller, T. (2004). Rethinking visualizations: A high-level taxonomy. *Proceedings of InfoVis '04*, 151-158.

Viegas, F. B., & Donath, J. (2004). Social network visualization: Can we go beyond the graph?. *Workshop on Social Networks, CSCW*, *4*, 6-10.

Viegas, F. B., Wattenberg, M., Van Ham, F., Kriss, J., & McKeon, M. (2007). Many eyes: A site for visualization at internet scale. *Proceedings of IEEE Information Visualization.*

Ware, C., Purchase, H., Colpoys, L. & McGill, M. (2002, June). Cognitive measurements of graph aesthetics. *Information Visualization*, *1*(2), 103-110.

Wasserman, S., & Kaust, K. (1997). Social network analysis: methods and applications. *Cambridge University Press.*

Wong, P. C., Foote, H., Mackey, P., Chin, G., Sofia, H., & Thomas, J. (2008). A dynamic multiscale magnifying tool for exploring large sparse graphs. *Information Visualization*, *7*(2), 105-117.

BIOGRAPHICAL SKETCH

Ana Rocio Peña was born on January 26, 1983, in Reynosa, Tamps Mexico. She attended the University of Texas Pan American and received the Bachelor of Computer Science degree in Edinburg in December 2005. While completing her bachelor's degree, she was also a member of the tennis team for four years. Ana Rocio lives in fifteen hundred Colosio Lane, Mission Texas, 78572. Ana Rocio is married to Fernando Xavier Lopez and her major interests include technology and tennis. Currently Ana Rocio is working for South Texas College as the Technologies Coordinator for Distance Education department where she is the system administrator for the college's Blackboard system. Her career objectives are to continue at South Texas College as an administrator and as a part time faculty member for the Computer Science department.