COMPUTER SCIENCE HONOURS PROJECT REPORT

Interactive Visualisation of online learning Social Networks and inferring its Bayesian Learnt Models

Mogamat Kaashief Hartley
Department of Computer Science,
University of Cape Town,
Rondebosch,
Cape Town,
South Africa,
7700
Hrtmog005@ uct.ac.za
November 2008

Supervised by:

A/Prof Sonia Berman
Dr. Anet Potgieter

ABSTRACT

The SONET project was proposed as an opportunity to build an intuitive tool for lecturers and students to use to analyse online learning social networks. The SONET system consists of three sections; the database, Bayesian networks subsystem and visualisation subsystem. The database subsystem cleaned the raw data from site forums and formed social networks to mine social network metrics. The Bayesian network subsystem used the data from the database subsystem to create and learn Bayesian models. The Bayesian network can then cluster nodes from the social networks according to their metric levels. This project describes the visualisation and Bayesian inference subsystem. The visualisation takes in data from both database subsystem and Bayesian network subsystem. The visualisation component takes in social networks formed in the database subsystem. It presents these in a human understandable format that can be used to extract knowledge more efficiently and provide insight into communication patterns of online learning forums. This project endeavoured to find an intuitive dynamic view of the social networks as forum networks are more highly connected than the conventional social networks. For this reason, two layouts were considered namely, the graph layout and radial graph layout views. The tests showed that while users performed equally well in both the graph and radial views, the users found that the radial view was much simpler and easier to work with. The Bayesian inference engine was successfully built and was used to predict student marks. The inference engine was tested against a well known Bayesian inference tool, Javabayes. The inference tests showed that the output from the inference engine was exactly the same as the Javabayes output.

Keywords: Bayesian Network Inference, Social Network Visualisation

Contents

List of Figures

1. INTRODUCTION

As part of a Computer Science Honours Project, known as SONET, the author was led to research various techniques for visualisation of online learning social networks and inferring its Bayesian learnt structures. The SONET system attempts to find a correlation between social network metrics and student performances by using a Bayesian network trained on forum and student records data.

The SONET system endeavours to:

1. Analyse the effectiveness of communication on course sites.

2. Viewing student clusters according to social network metric levels.

3. Reveal if there is a correlation between participation in the communication network with performance and attempt to predict how students will perform.

4. Create more intuitive visualisation techniques of social networks

5. Provide detailed documentation for the system.

6. Increased modularity in the system.

This report addresses improvement in 4, in addition to contributing other sections of the system. Visualisation of information is key to any system that analyses data as SONET does. The visualisation provides a natural way to communicate connectivity and allows for fast pattern recognition by humans. Visualisation allows people to use a natural tool of observation and processing, their eyes as well as their brains, to extract knowledge more efficiently and find insights [9].

This project has two stakeholders (also the project's super users), Tony Carr, coordinator at the Centre for Higher Education, UCT. He has interest in teaching with technology and seeking insights into online learning social networks. The other stakeholder is Terence Dowdall, a course convenor for the first year psychology course (PSY1001W) at UCT. He runs the course using online forums and makes it compulsory for his PSY1001W students to participate in the course site forum.

A social network is a social structure made of nodes and links (relationships) between nodes. Nodes are generally individuals or organizations. Social Network Analysis (SNA) is the mapping and measuring of relationships between people and groups, and can be used to analyse the flow of information/knowledge within the social network. This analytical method is useful for understanding relationships within a social domain and can be used for making predictions.

The report basically covers all aspects of the project. Section 2 provides background theory about current research into social network visualisation and visualizing Bayesian networks, effective visualisation techniques and tools available to create interactive information visualisation. Section 3 discusses the methodology used to elicit requirements, lists the final requirements and provides the software engineering behind the project. Section 4 explains the technologies used to implement the inference, visualisation and interface. Section 5 formulates a hypothesis and describes the testing procedures used. Section 6 reviews the results stated in the previous section. Conclusions about the interface and visualisation layouts are drawn and the null hypothesis is reviewed. Section 7 summarizes the project and provides reflection on the project. Section 8 rounds off the report with future work on the project noted by the developer throughout the duration of the project.

2. BACKGROUND

As this project investigates visualisation and Bayesian network prediction in learning forums, this chapter presents related work in each of these areas in turn.

2.1 Related Systems

2.1.1 SocialAction – Social network visualisation

SocialAction is a system that uses attribute ranking and coordinated views to help users systematically examine numerous SNA measures [19].

SocialAction allows users to "(1) flexibly iterate through visualisations of measures to gain an overview, filter nodes, and find outliers, (2) aggregate networks using link structure, find cohesive subgroups, and focus on communities of interest, and (3) untangle networks by viewing different link types separately, or find patterns across different link types using a matrix overview" [19].

SocialAction was demonstrated on a network of terrorist groups. The system is able to accommodate for other data sources and not just this domain on social networks. This system was implemented in JAVA and integrates several open source toolkits. JUNG [16] was used to provide the underlying node-link data structures. Prefuse [5] was used for network visualisation. Piccolo [19] was used for the scatter plot and matrix visualisations.

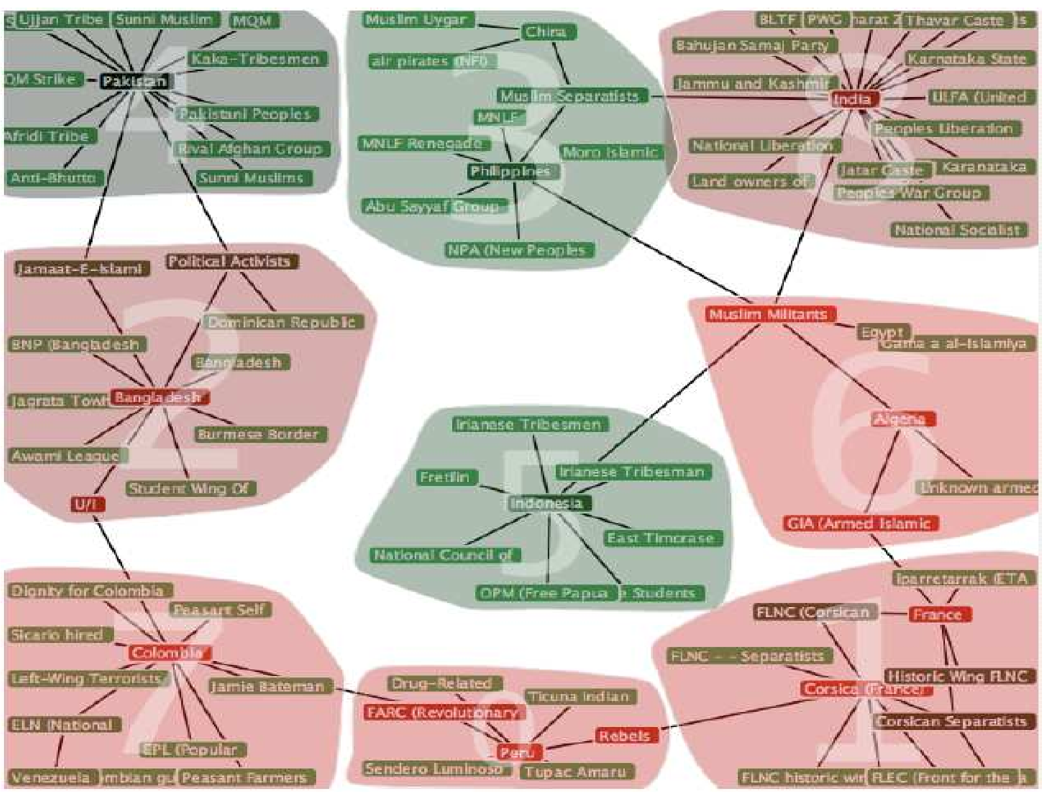


Figure 1 Shows a screen shot of SocialAction. This image displays how communities can be extracted from a social network and how the visualisation can add to highlighting these communities.

2.1.2 Vizster – Social Network visualisation

Vizster is a tool created to visualize online social networks. The design and implementation of the Vizster visualisation system is meant for playful end-user exploration and navigation of large-scale online social networks [4]. This tool provides similar functionality to that of SocialAction. It differs in that it offers more customizability and interaction of the displayed social network graphs. Vizster is meant to be a social tool rather than a research tool. Vizster incorporates a search facility that uses the search query to map out nodes in the graph that have common attributes. Users are able to add more node detail while viewing the graph. Vizster also implements a function to identify communities (by clustering) within the graph and is provided with a slider. The initial state of the clustering is displayed and can be adjusted by moving the slider to merger nearby clusters or communities. This function is displayed in Figure 2 (i) and Figure 2 (ii) below.

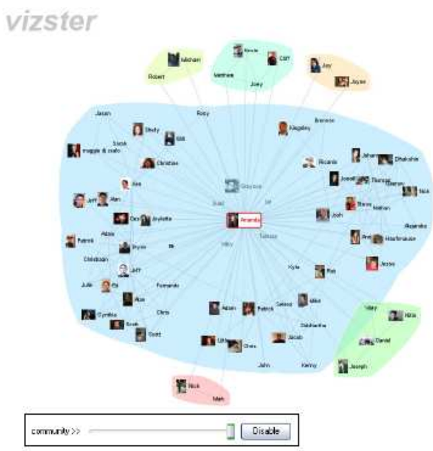


Figure 2(i) Initial State

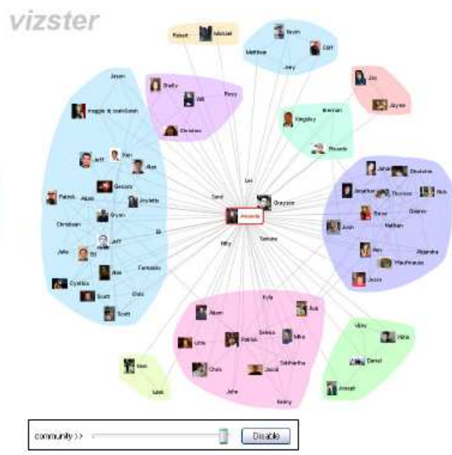


Figure 2(ii) with slider

There are a great number of social network visualisations; but there is still a need for new designs and techniques. "Though limited network views have proven manageable and useful, means for better exploring extended network contexts and providing useful cues for exploration are possible. Visualisation of profile attributes unique to online social networks is needed and techniques for incorporating analytical tools within the simplified domain of end-user visualisation may prove useful" [4].

2.2. VISUALISATION

Visualisations play an important role in abstracting new insights in social network analysis. Visualisations provide a natural way to communicate connectivity and allows for fast pattern recognition by humans. However, there are great challenges when visualizing networks.

2.2.1 Interaction

Basic interaction with the social network is done with simple mouse operations and at times simultaneously with the keyboard. Most network visualisation tools incorporate basic principles of cartography e.g. interactive zooming. User navigation could be hampered by large networks. This problem can be alleviated by aggregation of nodes based on link structure [19].

2.2.2 Formatting

There are a large number of layout algorithms that attempt to calculate the position of each node and the curve of each link to minimize link crossing.

Formatting of the network is essential as it gives the aesthetic of the closeness of interaction between the nodes observed. One popular formatting algorithm is Newman's community identification algorithm [14]. Newman's community identification algorithm is used in many visualisation systems because it is fast enough to support interactive real-time adjustments. This algorithm has been successfully applied in the SocialAction and Vizster social network visualisation systems.

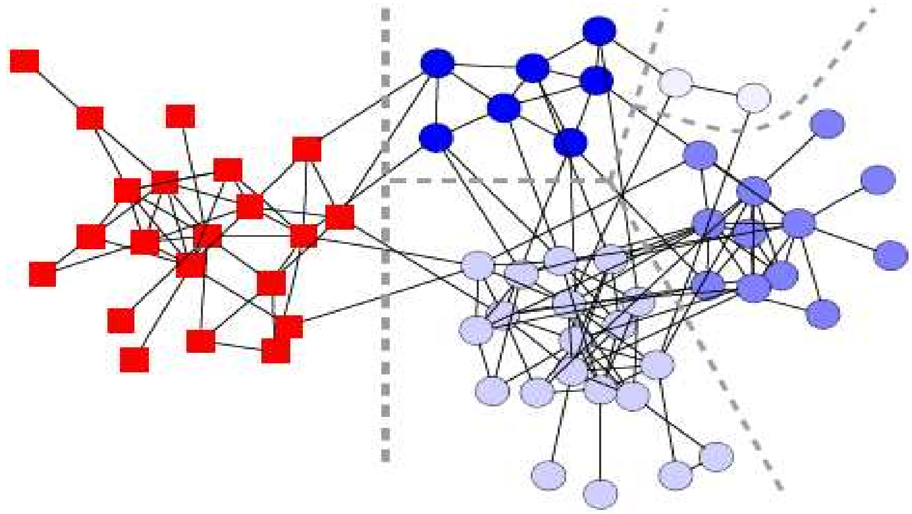


Figure 3 - Newman's community identification algorithm

Newman's community identification algorithm identifies group structures based solely on link analysis. "The algorithm employs hierarchical agglomerative clustering, first placing each node in its own community, and then greedily merging groups based upon a metric that attempts to maximize within-cluster linkage while minimizing between-cluster linkages" [19].

2.3. TOOLS

There are a number of software tools designed to help researchers design, understand and view social networks. These tools usually are composed of useful tools that apply to social network analyses.

2.3.1 Jung

Jung is a Java based software library that provides a common and extendible language for the modeling, analysis, and visualisation of data that can be represented as a graph or network [8]. Jung provides the underlying node-link data structures, as well as an implementation of some of the SNA ranking algorithms.

Jung also provides a visualisation framework that makes it easy to construct tools for the interactive visualisation and exploring of network data. Layout algorithms are provided, or the framework can be used to create custom layouts. In addition, filtering mechanisms are provided which allow users to implement algorithms on specific portions of the graph [8].

2.3.2 Prefuse

Prefuse is a set of software tools for creating rich interactive data visualisations. Prefuse supports a rich set of features for data modeling, visualisation, and interaction. It provides optimized data structures for tables, graphs, and trees, a host of layout and visual encoding techniques, and support for animation, dynamic queries, integrated search, and database connectivity. Prefuse is written in Java, using the Java 2D graphics library, and is easily integrated into Java Swing applications [22].

2.3.3 Piccolo

Piccolo is a toolkit that supports the development of 2D structured graphics programs, in general, and Zoomable User Interfaces (ZUIs) [20], in particular. It allows for building of structured graphical applications without worrying so much about the low level details. The infrastructure provides efficient repainting of the screen, bounds management, event handling and dispatch, picking (determining which visual object the mouse is over), animation and layout.

Piccolo is a layer built on top of a lower level graphics API. There are currently three versions of the toolkit: Piccolo.Java, Piccolo.NET and PocketPiccolo.NET (for the .NET Compact Framework). The java version is built on Java 2 and relies on the Java2D API to do its graphics rendering. The .NET version is built on the .NET Framework and relies on the GDI+ API to do its graphics rendering. This makes it easy for Java and C# programmers [20].

2.4. INFERENCE

A Bayesian network is a Directed Acyclic Graph. The nodes of the graph represent variables and the edges represent causal links between variables. Each node has a conditional probability table that represents the relationship between a child node and its parent.

Inference is the process of extracting probable outcomes based upon the relationships in the model and the evidence known about the situation at hand (the query).

When a Bayesian network is actually used, evidence or observations are applied. This information is applied to the network by assigning a variable to a state that is consistent with the observation. Then the inference algorithm is applied to update the probabilities of all the other variables that are connected to the variable representing the new evidence. This allows queries to the Bayesian model.

After inference, the updated probabilities reflect the new levels of belief in all possible outcomes coded in the model.

The beliefs originally encoded in the network are known as prior probabilities, because they are entered before any evidence is known about an observation. The beliefs computed after evidence is entered are known as posterior probabilities, because they reflect the levels of belief computed in light of new evidence.

Bayes' theorem adjusts probabilities given new evidence in the following way:

Equation 1

$$P(H|E) = \frac{P(E|H)\ P(H)}{P(E)}$$

Where

- *H* represents a specific hypothesis, which may or may not be some null hypothesis.
- *P*(*H*) is called the *prior probability* of *H* that was inferred before new evidence, *E*, became available.
- *P*(*E* | *H*) is called the *conditional probability* of seeing the evidence *E* if the hypothesis *H* happens to be true. It is also called a *likelihood function* when it is considered as a function of *H* for fixed *E*.
- *P*(*E*) is called the *marginal probability* of *E*: the *a priori* probability of witnessing the new evidence *E* under all possible hypotheses. It can be calculated as the sum of the product of all probabilities of any complete set of mutually exclusive hypotheses and corresponding conditional probabilities:

Equation 2

$$P(E) = \sum P(E|H_i)P(H_i)$$

P(*H* | *E*) is called the *posterior probability* of *H* given *E*.

Bayesian Inference is known to be NP hard [21].

2.4.1 Examples of Bayesian Network Inference Algorithms

Bayesian Network inference algorithms are generally classified into two categories; **exact inference** and **approximate inference**.

2.4.1.1 Exact Inference
The posterior probability of a hypothesis is calculated in totality.

Examples of Exact Bayesian Network Inference algorithms

Variable Elimination (VE) involves summations over variables in order to eliminate them from the network. The complexity of VE can be measured by the number of numerical multiplications and numerical summations it performs [23].

Judea Pearl [18] published an efficient *message propagation inference algorithm* for polytrees. The algorithm is exact and has polynomial complexity in the number of nodes, but works only for polytrees. Polytrees are graphs with at most one undirected path between any two vertices which is a Directed Acyclic Graph.
Pearl also presented an exact inference algorithm for multiply connected networks called *loop cutset conditioning* [18]. In Bayesian Networks, multiply connected belief networks are graphs that contain cycles.

The most popular exact Bayesian Network inference algorithm is Lauritzen and Spiegelhalter's *clique-tree propagation algorithm*.

It is also called the “clustering” algorithm. “It first transforms a multiply connected network into a clique tree by clustering the triangulated moral graph of the underlying undirected graph, then performs message propagation over the clique tree” [23].

For sparse Bayesian Networks exact inference works well, however for large and strongly connected Bayesian Networks it doesn't. For this reason, a heuristic approach to belief revision in Bayesian Networks is investigated [9] as described in 2.4.1.2 below.

2.4.1.2 Approximates Inference
Approximate Inference is based on randomized sampling algorithms. Approximate answers are given to queries.

Examples of Approximate Bayesian Network Inference algorithms

Model Simplification Methods
Model simplification methods first simplify the model until exact methods become feasible and then run an exact algorithm.

Loopy Belief Propagation
Loopy belief propagation (BP) – the use of Pearl's polytree propagation algorithm in a Bayesian network with loops has been researched by Murphy [12] and Weis [24]. The messages might circulate indefinitely around loops, but the values obtained are usually very close to the true values [21].

2.4.2 Algorithms for real time Bayesian network inference

Most interactive systems using Bayesian Networks such as SNA systems require computation in real time. The Bayesian Network inference would be required to run the user queries in a suitable time. This requires an inference algorithm to not only depend on its accuracy but also on its timeliness.

Anytime Bayesian Network Inference Algorithms
Theoretically, any Bayesian networks inference algorithm that temporarily ignores partial information contained in a Bayesian Network, and recovers ignored information whenever the allocated computational time allows, is an anytime inference algorithm. This partial information could be for e.g. partial nodes, partial edges, partial probabilities in Conditional Probability Tables (CPTs) or partial node states. Therefore, most approximate inference algorithms can be easily used as an anytime algorithm by applying them iteratively [23].

Example Anytime Algorithms
Variational probabilistic inference transforms the network to a simpler network by variational de-linking nodes and executes an exact algorithm when the resulting graph is sparse enough. Its accuracy generally improves as more nodes are treated exactly and hence has an anytime characteristic of time-accuracy trade-off [4].

2.5. CONCLUSION

Various Inference algorithms and visualisation tools have been discussed in this chapter. From the research presented we can conclude there are a great number of social network visualisations; but there is still a need for new designs and techniques.

JUNG is a widely used, well supported and probably most useful open source visualisation toolkit. This toolkit can be integrated with other open source tools, such as Prefuse, to take advantage of more contributions. Also Newman's community identification algorithm is highlighted as popular in many visualisation systems. In terms of inference algorithms; we may conclude that many types of algorithms exist to analyse Bayesian Networks. The belief propagation algorithm (also known as message passing algorithm) is extremely efficient, yet relatively simple when compared to other general case algorithms. It was chosen to be implemented in this project.

3. DESIGN SPECIFICATION

3.1 Environment

As Java is a well supported language and there are a variety of open source tools available it was the selected language to develop the SONET social network tool. There are well known Open Source Bayesian network tools that are built in Java, namely JavaBayes [7] and BNJ [10].

Most languages have corresponding environments created to allow easier development in the preferred language. As such, the IDE chosen should also have capabilities to ease development tasks. NetBeans was selected as the development environment for this project. It supports project integration which allows for easy maintenance and updates to the project.

3.2 System Overview

The SONET system was proposed to mine a social network from raw data extracted from an online Vula course site. The data from this course as well as student records are stored. The idea was to implement a Bayesian network using the given data in order to profile students and predict student performance. The system was divided into three sections, namely the data subsystem, Bayesian network creator, and thirdly the system described here that performs inference and visualisation.

Data Mining Subsystem

Ivandro Ismael Issufo was responsible for the data mining subsystem. The data mining subsystem obtained the raw data from Vula forums and cleaned it up by removing noise such as outlying observations and incomplete data. This cleaned data was then be fed into an automated mining tool that extracted a social network structure from the data and stored this structure in a new database. In addition to the social network, data pertaining to a user's profile (such as age, gender, culture etc) will also be extracted and placed in the new database. The subsystem then also extracts the forum information (reads and writes) to form complete undirected communication graphs.

Bayesian Network Subsystem

Aviskar Bhoopchand was responsible for the Bayesian network subsystem. This involved the learning the Bayesian network structure and generating the Conditional Probability Table (CPT) from given past data. This will determine the probabilistic relationship between a student's profile and their marks.

Visualisation Subsystem

This report describes the visualisation subsystem as well as the inference engine (prediction tool). Once profiles have been inferred from the Bayesian network using known information (evidence), findings are applied to the social network. These profiles contain detailed information which must be incorporated into the visualisation system. This is split into three parts, user requirements, inference engine, and visualisation layout.

3.3 Methodology

The iterative development model was applied in creating the interface of the SONET system and visualisation of the social network. This methodology was chosen to mitigate requirements for the system that were vague and to elicit requirements a prototyping method was used. Figure 4 below shows the tasks associated with the model and shows that evaluation is central through each cycle.

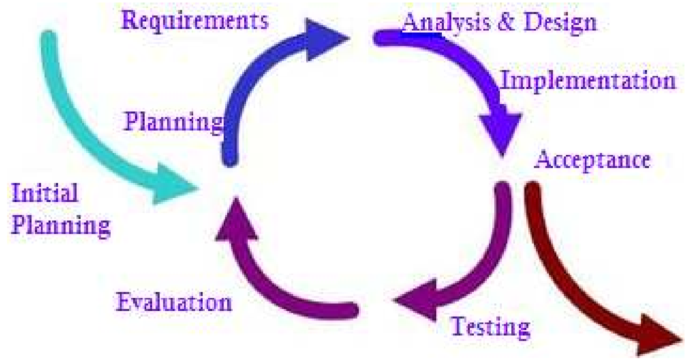


Figure 4 Iterative Methodology

This project was initiated with the supervisors and project team members, and super users were identified. The super user(s) identified were:

Tony Carr, Center for Educational Technology.

From the project kick off meeting held with the super user, requirements were ascertained and key success factors were produced and analyzed. The major issue on building the SONET system was to determine which data would be used to model the system on. There were a variety of choices to model on.

Initially, DFAQ (Dynamic Frequently Asked Questions) was chosen but after inspecting the content it was not conducive to this study.
The second choice was the UCT first year Psychology course (PSY1001W). After inspection these forums were found to be more useful to our study and were thus chosen for the project/investigation. Hence, we required another super user familiar with these forums which was Terence Dowdall, PSY1001W Psychology Course convener.

3.3.1 Final System Requirements

Functional Requirements

Based on the interviews and prototyping with super users, user requirements were elicited. These requirements were then further analyzed in conjunction with the supervisors to create the functional requirements specified below.

These requirements were used to measure the success of the project. Each member of the team expanded on their requirements for success.

This project aimed to develop a tool that can:

- analyse the effectiveness of communication on course sites
- identify the spreading of knowledge amongst users
- identify knowledge hubs
- identify users not contributing knowledge to a community
- correlate participation in the communication network with performance and attempt to predict how students will perform

- help users identify social patterns in the social network
- Visualize the social network to create meaningful.
- the system should be extensible to work with data from any source in the specified format
- components of the system should be modular to allow for reuse
- be navigated intuitively and requires little training to use

3.3.2 Package Diagrams

The SONET system consists of a main SONET package. The SONET package consists of three subsystems, namely the:

- Data Mining subsystem
- Bayesian Network subsystem
- Visualisation subsystem

Figure 5.1 shows the SONET package and indicates the dependencies between subsystems. The Bayesian network subsystem depends on the Data Mining subsystem. The Visualisation subsystem depends on both the Data Mining subsystem and Bayesian Network subsystem.

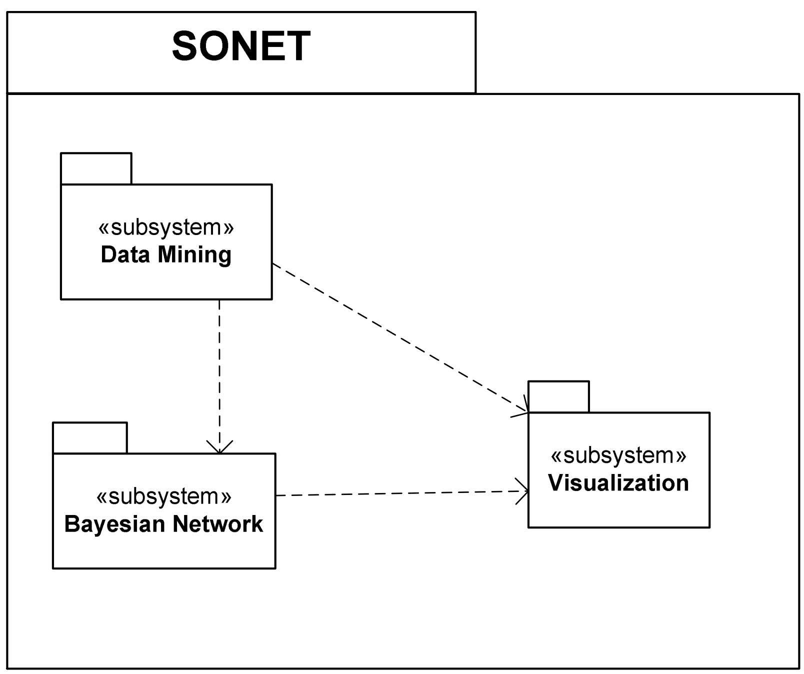


Figure 5.1 SONET Package diagram

The rest of this report will focus mainly on the Visualisation subsystem. A lower level package diagram of this subsystem is displayed below in Figure 5.2. The Visualisation is developed as a modular component and can be treated as a package on its own.

The Visualisation package consists of three subsystems, namely the:

- Inference subsystem
- Graph layout subsystem
- Interface

Figure 5.2 shows the Visualisation package and indicates the dependencies between subsystems. The Interface subsystem depends on both the Inference subsystem and Graph layout subsystem.

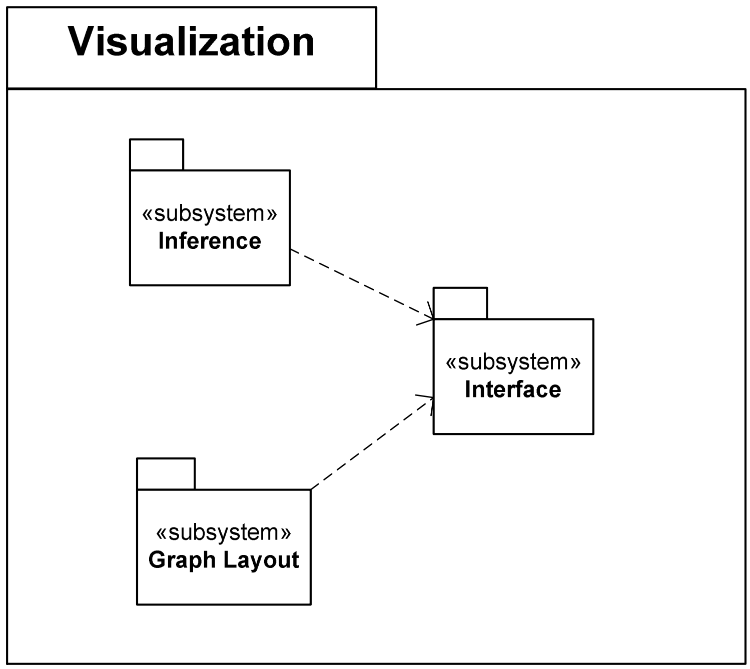


Figure 5.2 Visualisation Package diagram

The user requirements, specified in the previous section are visually represented by Use Case Diagrams (UCD). These software artefacts are described next.

3.3.3 Use case diagrams

Use case modelling was used to identify and describe the requirements of the SONET system and its interactions with users and also between subsystems. This modelling gave some insight into the integration issues when developing SONET. The use case diagrams below are used to describe sequence of actions that provide value to a user or external system.
The interface and inference use case diagrams are discussed separately. The inference UCD is however contained in the interface UCD.

The interface UCD is shown in Figure 6.1. Figure 6.1 demonstrates how a user interacts with the system.

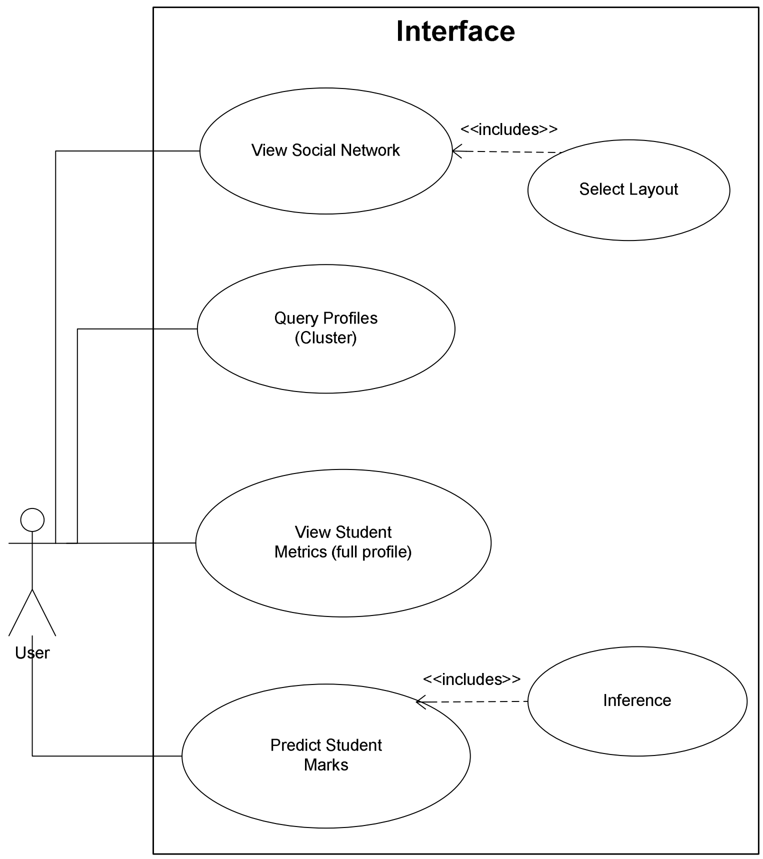


Figure 6.1 Interface UCD

The Inference UCD is shown below in Figure 6.2. Figure 6.2 demonstrates how a user interacts with the inference engine which is used for prediction. The inference engine is also used in the expectation maximization algorithm of the Bayesian network subsystem [2].

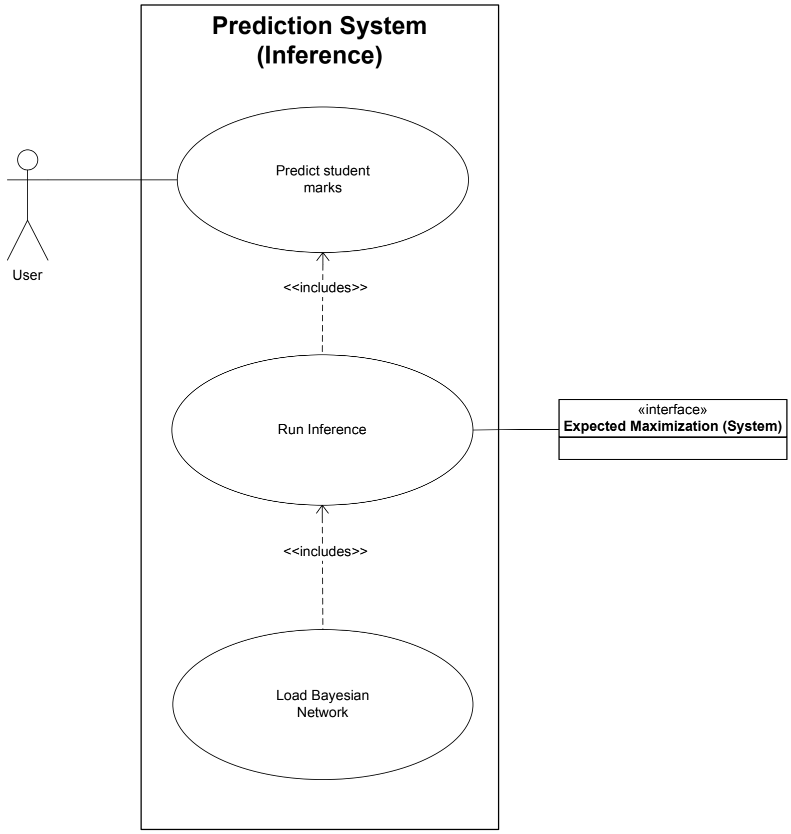


Figure 6.2 Inference UCD (Prediction)

3.3.4 Subsystem Integration

The System flow diagram in Figure 7 depicts the different sub-systems and how they are integrated. The data set (raw data) is fed through the Database subsystem. The Database subsystem cleans the data set and generates a social network. From the generated social network, metrics are calculated for each node. These metrics are stored in an intermediary database.

The metrics are then retrieved from the database into the Bayesian Network subsystem. Here Bayesian Learning is implemented and CPT tables are generated.

In the Visualisation subsystem, the social network generated by the database subsystem is retrieved from the intermediary database. The social network data is then automatically structured to social network graphs. The Bayesian networks computed by the Bayesian learning that are stored in the intermediary database are then fed into the inference engine (prediction tool) to provide the prediction features on the interface.

The Visualisation subsystem is divided into three subsections, namely the

- Inference engine - to handle user queries concerning predictions and evidence evaluations.
- Data representation - to store the social network graphs. This subsection is where the data is pumped into data structures.
- Interface – handles user interactions.

Data representation in the intermediary database will be discussed in the next chapter.

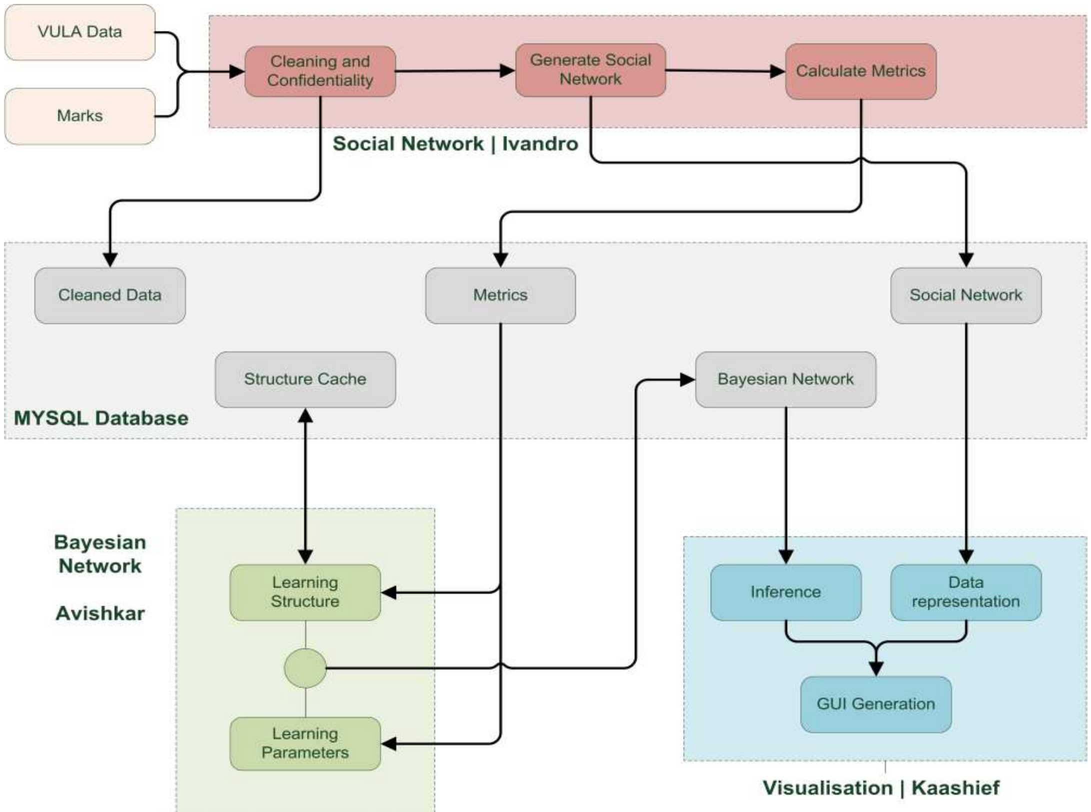


Figure 7 System flow - process integration diagram of SONET

3.3.5 Class Diagrams

The class diagrams below were used to model two core subsystems of the Visualisation package, namely the Graph Layout subsystem and inference engine subsystem.

The high level detailed Class Diagram of the Graph Layout subsystem is shown in Figure 8.1. Figure 8.1 shows the classes involved in creating the different layouts. Two frames are added to complete the interface namely, the student profile frame and the prediction frame.

The student profile frame is used to display a student's complete profile and an overview of the student's metrics. The prediction frame incorporates the inference engine and implements the inference UCD. The main interface, NetworkVis implements the interface UCD.

Further discussion of the implementation will be given in the next chapter.

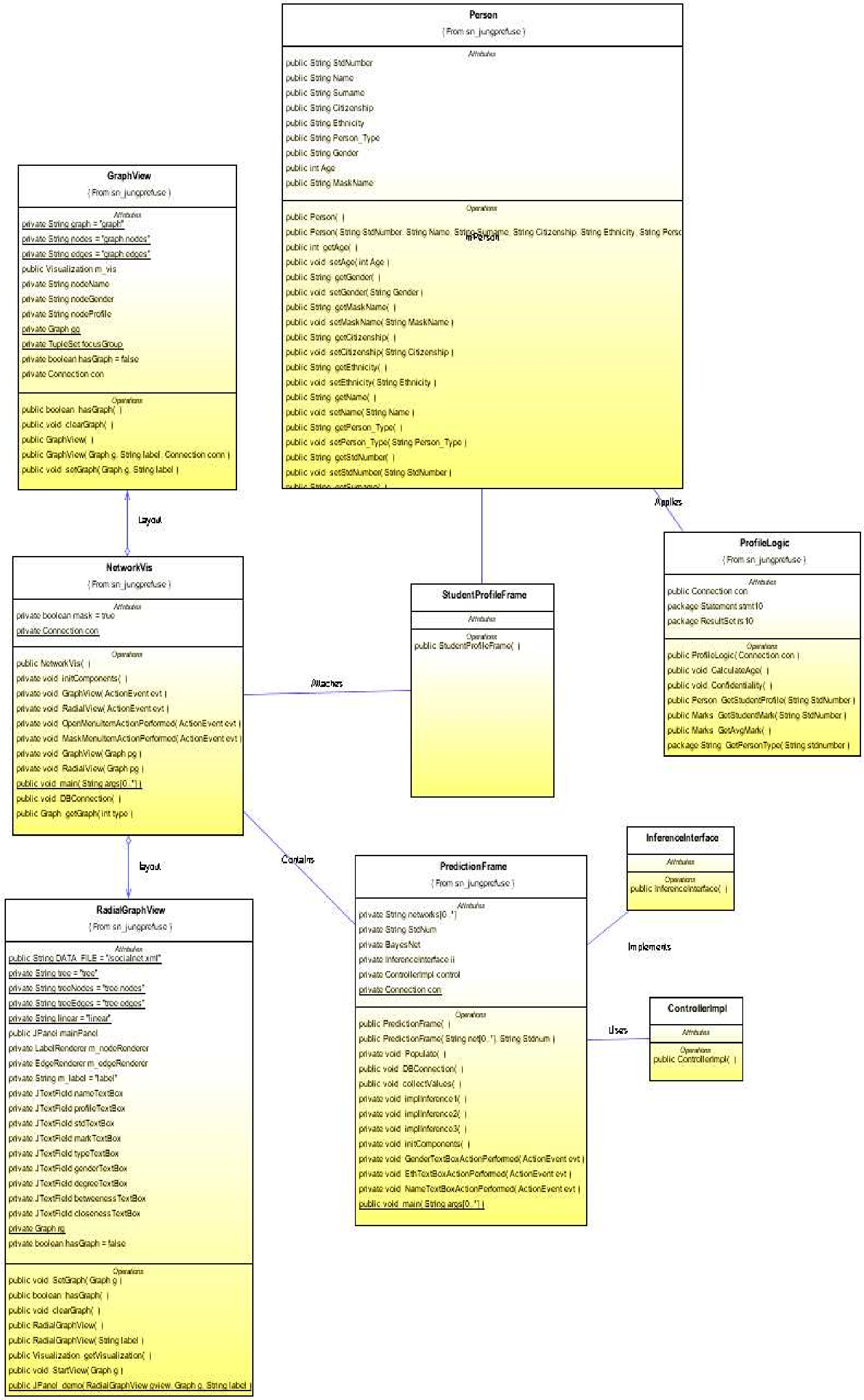


Figure 8.1 Interface and Graph Layout subsystem Class Diagrams

Figure 8.2 shows the relationships between classes in the Inference Engine.

The BayesNode class contains States and a CPT table (BayesCPT). Links have been created between children nodes and parent nodes, these nodes fill the Bayesian network (BayesNet). The inference interface provides a simple interface allowing the Inference Engine to be modular so that it can be plugged into almost any system. A further discussion of Inference implementation will be given in the next chapter.

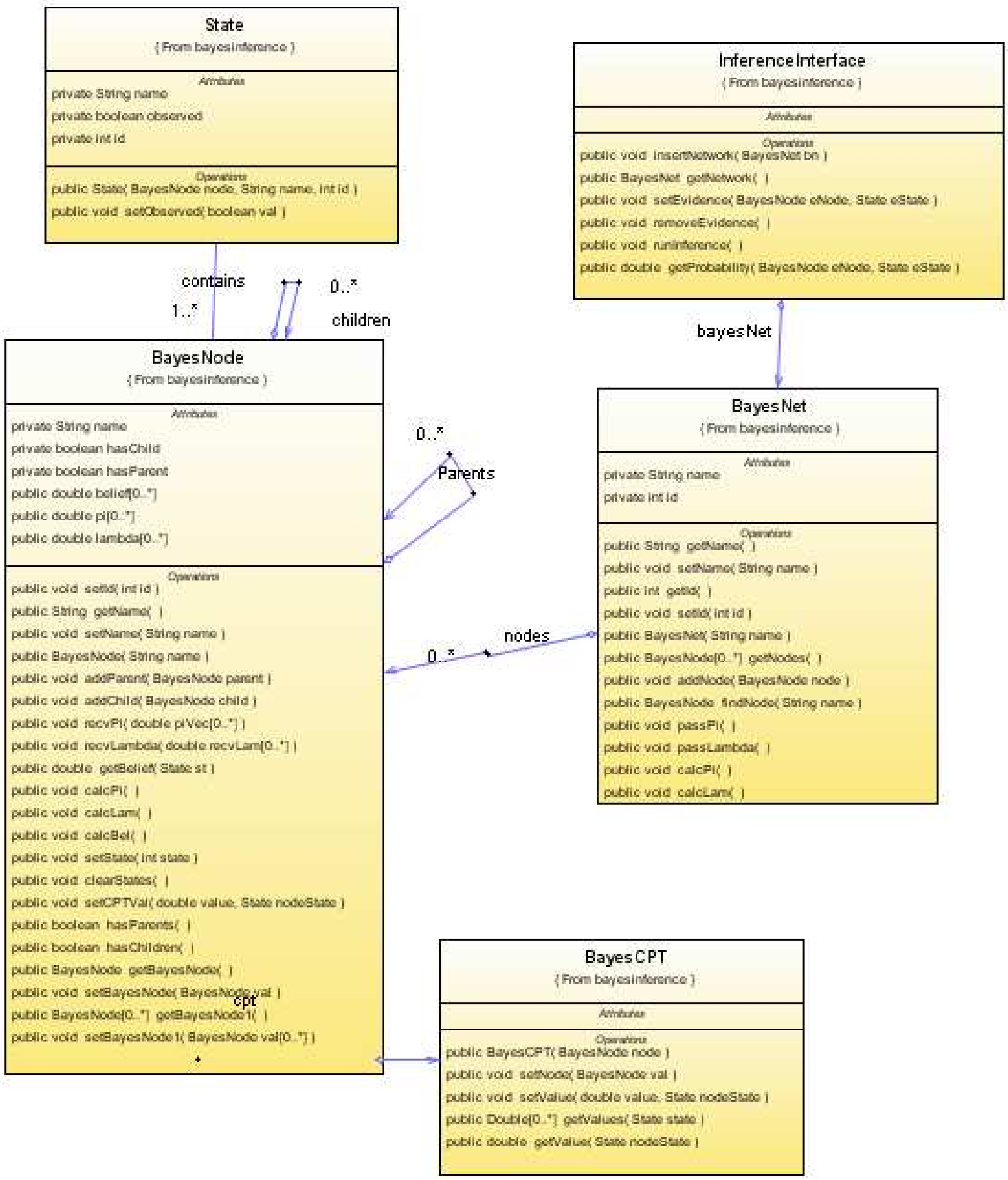


Figure 8.2 Inference Engine Class Diagram

3.3.6 Sequence Diagrams

The Visualisation sequence diagram is shown below in Figure 9.1. Figure 9.1 illustrates how a user would interact with the interface of the SONET system. User would want to view the social network, initially a layout has to be selected or by default one would be selected. The user has the option of either selecting the graph view or radial view. Once a view has been selected the graph has to be loaded and sent to the selected layout class.

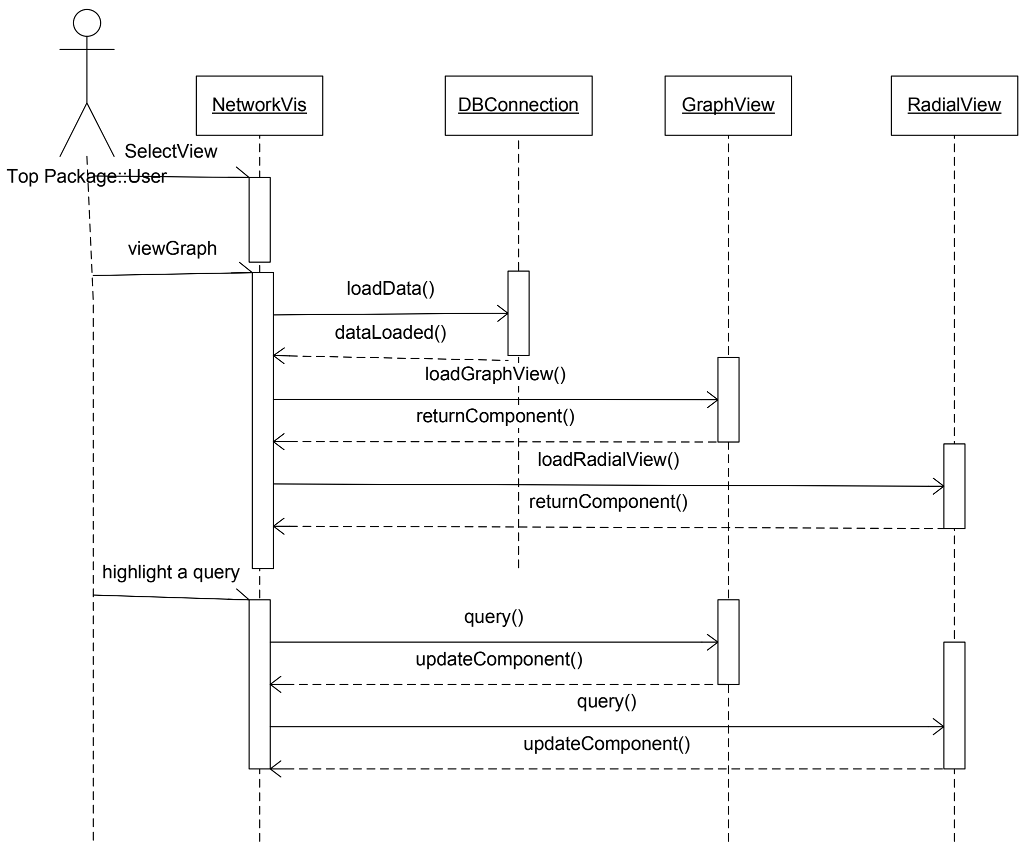


Figure 9.1Visualisation Sequence diagram

Prediction sequence diagram

Figure 9.2 displays the Prediction sequence diagram. It illustrates how a user interacts with the system to predict a student's mark. The social network has to be displayed before a user can select a student. Once a student is selected the user can choose the option to predict student marks.

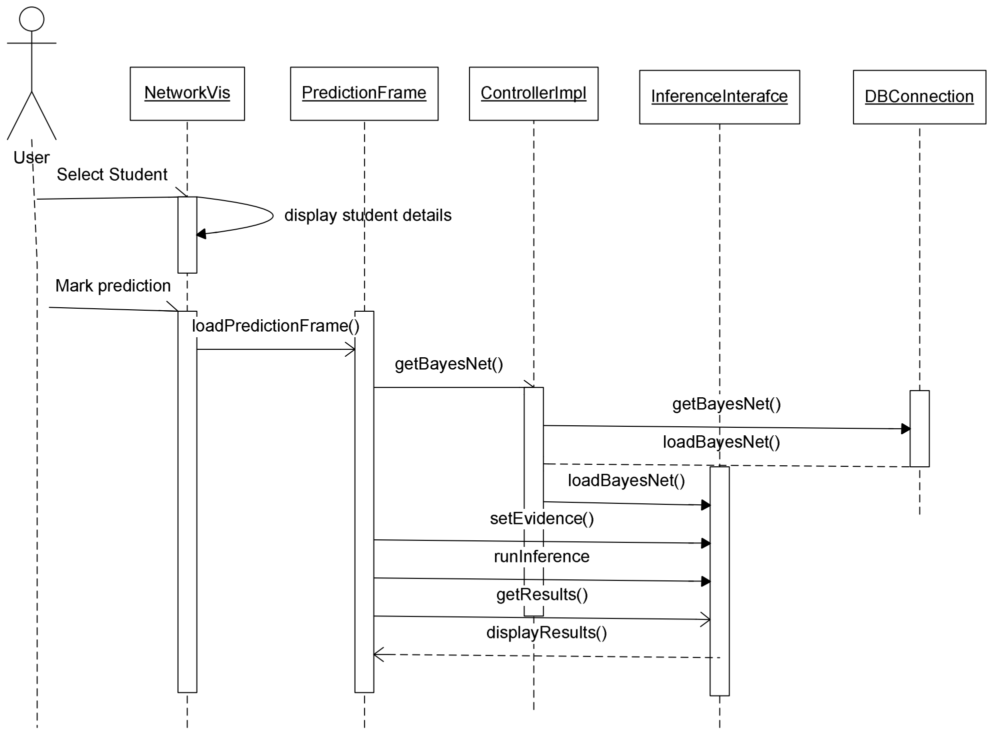


Figure 9.2 Prediction Sequence Diagram

3.3.7 Interface Requirements

Interface requirements are selected from the functional requirements that pertain to visualisation. The super users require an intuitive interface that requires little training to use. The subsections of the interface can be seen in Figure 6.1. Each use case scenario of the Interface UCD is handled by the interface.

Features that are provided on the layouts are:

- Graph view layout
    - Connectivity filter – reduces the graph according to the number of distance links is specified by the filter. The links originate from the focus node (demonstrated in example below).
    - Force panel – this panel contains the layout properties used by the force directed algorithm
        - Spring length
        - Distance between nodes
        - And force simulation parameters
    - Highlight control – when hovering over a node, the nodes directly linked to it are highlighted.
    - Double clicking on a node sets that node to be the focus node. Its details are shown in the information panel.

- Radial view layout
    - Angled view – by checking the angled view option transforms the graph into a 180°or 360°shape.
    - Radius length – the length between rings of the layout can be increased or deceased using this feature.

- o Highlight control – when hovering over a node, the nodes directly linked to it are highlighted.
- o Double clicking on a node sets that node to be the focus node. The graph is then shaped around that node. Its details are shown in the information panel.

The user of SONET has the option of choosing either the graph view or radial view from a menu option as displayed in Figures 10.1 and 10.2.

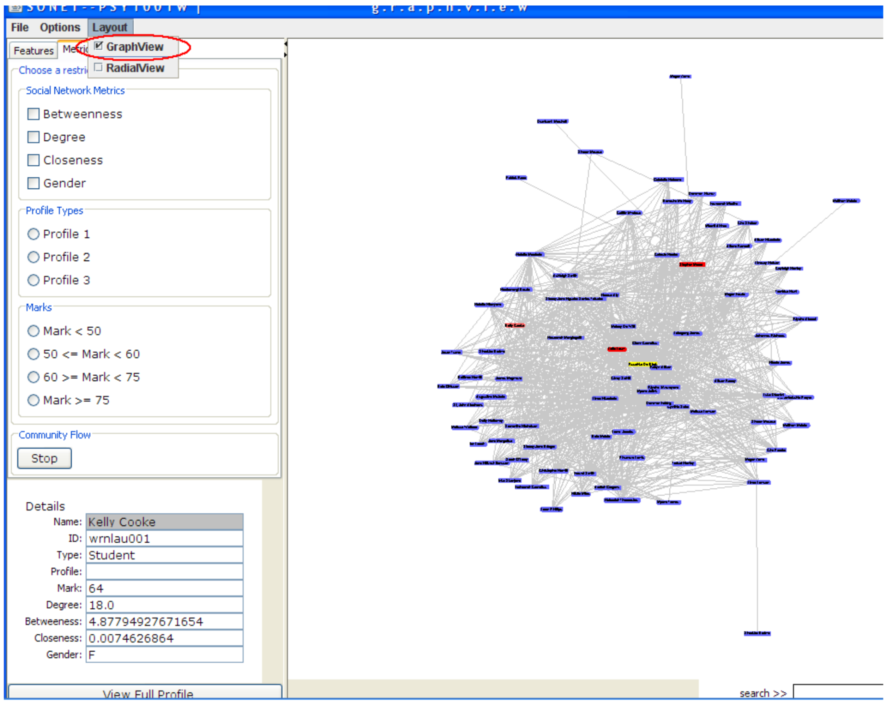


Figure 10.1 Graph View selected, nodes shown with fictitious name and ID

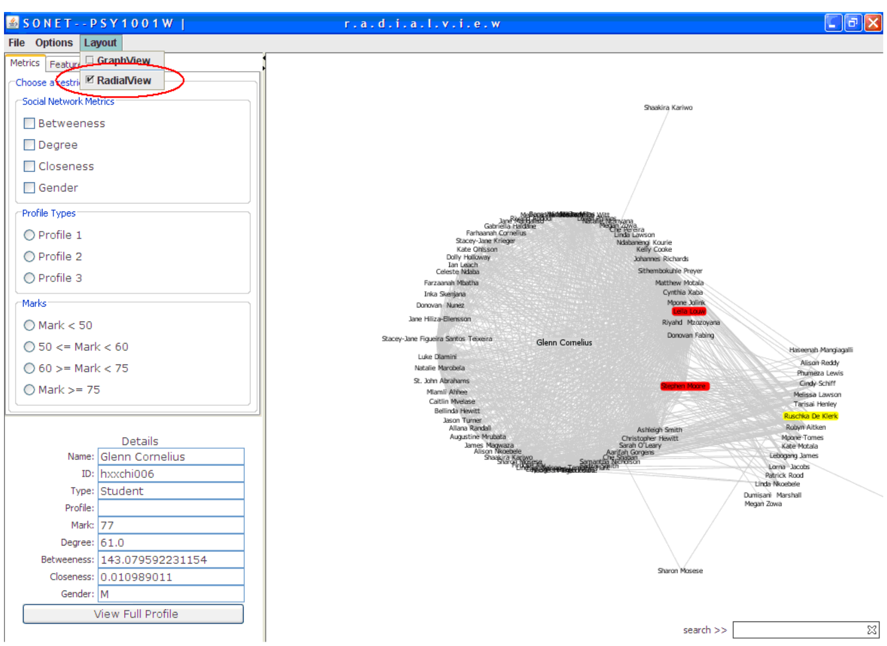


Figure 10.2 Radial Graph view selected, nodes shown with fictitious name and ID

The features included in the interface to allow for user exploration of the social network are as follows. The user can submit queries to the social network and the corresponding nodes will be highlighted. Queries available:

- Social network metrics
  - Betweenness – a node with a high level of betweenness connects mini networks within the larger network.
  - Closeness – a node with a high level of closeness has direct or indirect ties allowing them to access many nodes in the network more quickly.
  - Degree – a node with a high level of degree has many connections to other nodes.

- Profile – These are the clusters derived from the Bayesian clustering.
  E.g. Profile 2 – nodes with good social network metrics and achieving an average mark (55% - 65%).
  Profiles are created in the Bayesian network subsystem [2].
- Marks – Students' results.

Layout and Graph Colouring

The graph view can be seen below in Figure 11.1. The focus node is in orange, the students are coloured in blue, lecturers in red and the tutors in yellow.

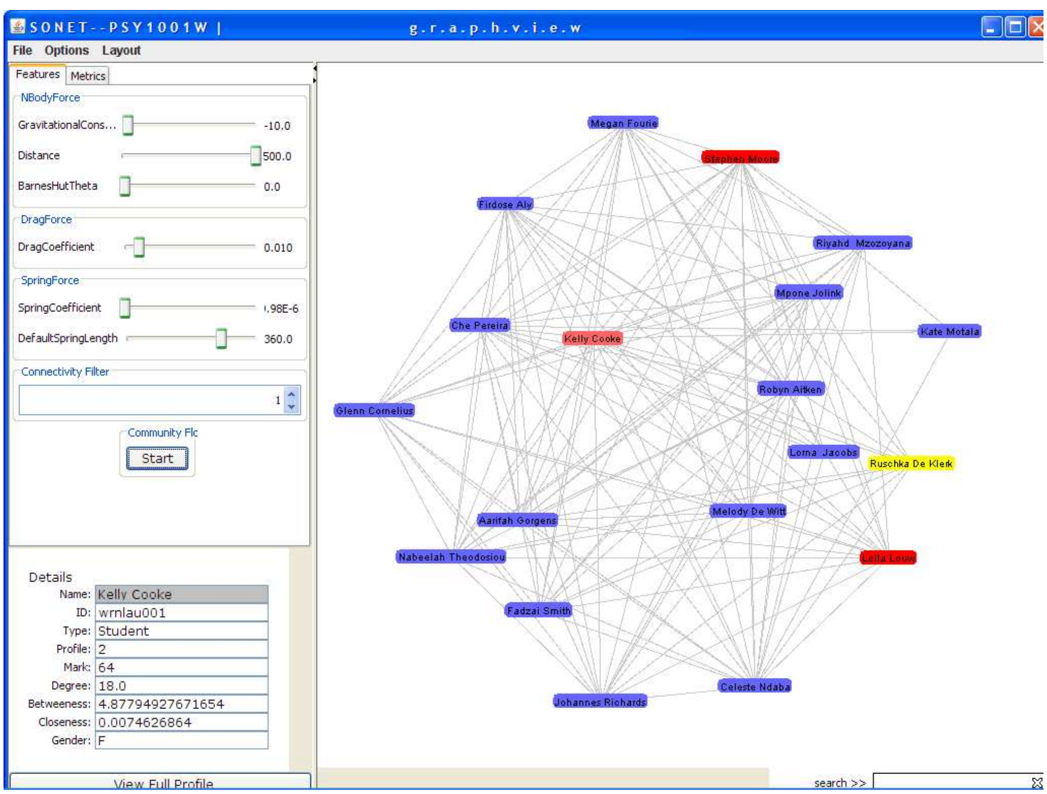


Figure 11.1 Graph view with fictitious names and ID

The radial view shown in Figure 11.2 displays the circular graph laid out. The focus node is coloured in light blue, the student nodes are not coloured with black font names, lecturers are coloured in red and tutors in yellow.

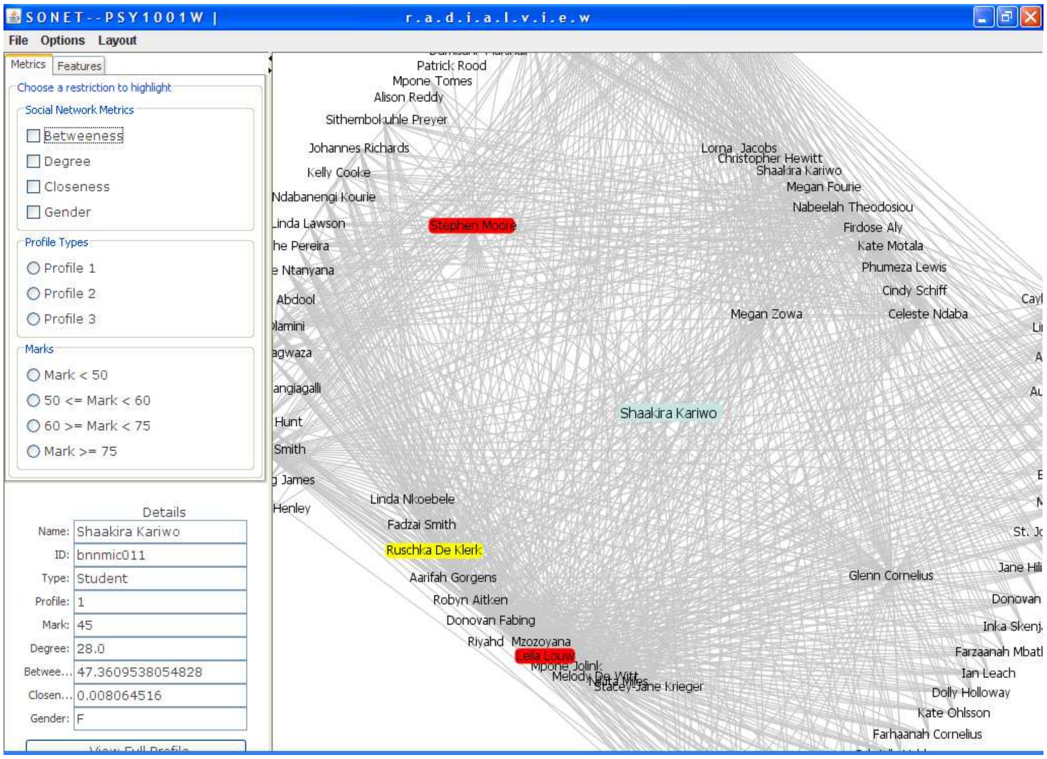


Figure 11.2 Radial view with fictitious names and ID

The use of connectivity filter is displayed in the Figures 12.1 and 12.2. Figure 12.1 shows a graph with focus node in orange and all its immediate neighbours. This node communicates with one lecturer and seven students directly. The graph demonstrates the connectivity filter with a value of 1.

Figure 12.2 displays the same focus node with connectivity filter value of 2.

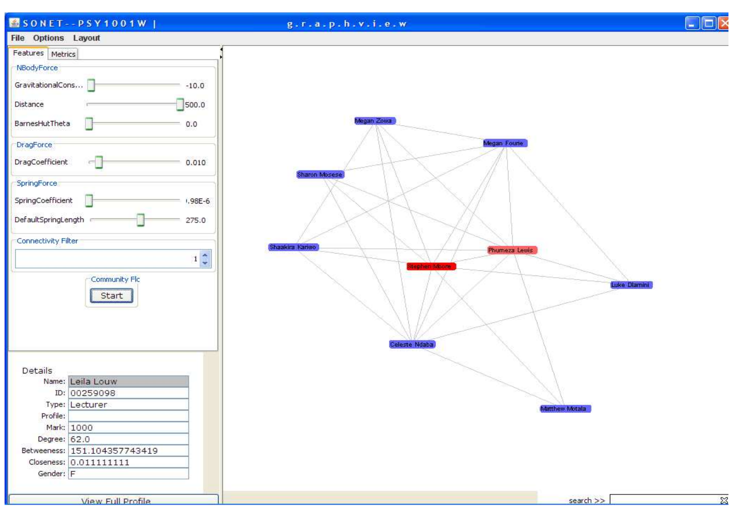


Figure 12.1 Graph view filter with connectivity 1

The nodes shown in Figure 12.1 are highlighted in orange in Figure 12.2. This displays the use of the highlight control when hovering over the focus node.

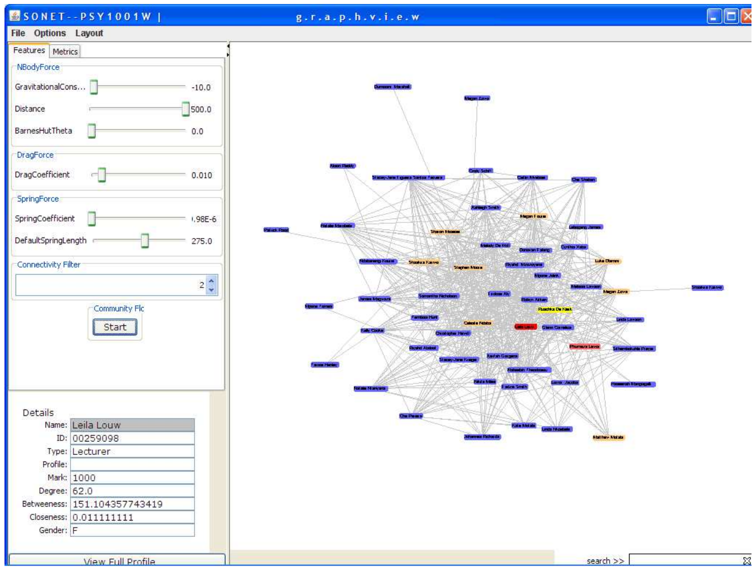


Figure 12.2Graph view filtered with connectivity 2

Queries can be viewed in Figure 13 below. Figure 13 displays how a user queries the social network using the graph view, by checking the requested detail boxes. The green nodes are the queried nodes. In this example high betweenness and a mark between 60 and 75 was chosen as a query.

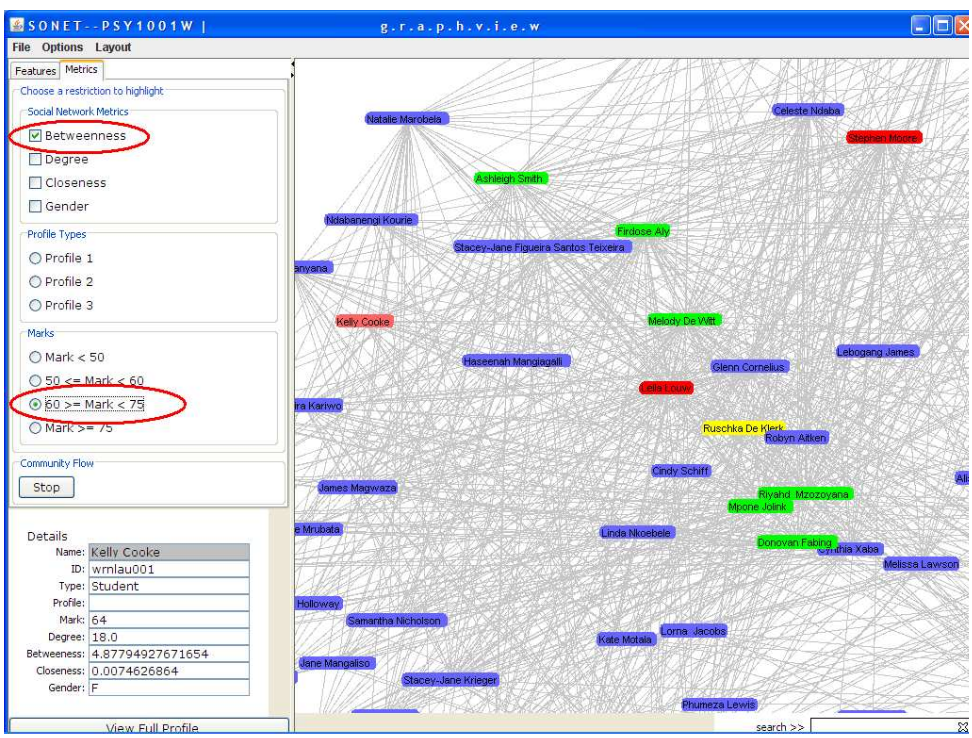


Figure 13 Queries and colour coding

4. IMPLEMENTATION

This chapter describes in detail how the system was implemented.

4.1 Tools

To build the graph layouts we have decided to use open source visualisation libraries. The libraries considered were the following:
Jung, Prefuse, JungPrefuse and OntoViz.

Jung provided tools for not only social network visualisation but for mining social network metrics as well. The first social network in the SONET project was implemented using Jung. However, Prefuse provided better visualisation graphics than Jung.

The second implementation was developed using JungPrefuse. This tool allowed social network structures created in JUNG to be visualized in the JungPrefuse tool. This was used to develop the first visual prototype.

However, this implementation was still inferior to the Prefuse visualisation graphics. JungPrefuse does not allow for all Prefuse features.

Finally, a full Prefuse implementation was developed and proved more successful than the previous attempts. But this implementation requires its own data structures other than the Jung data structures used to mine social network metrics.

The intermediary database allowed the Prefuse implementation to be more easily integrated with the social network developed by the Database system.
The OntoViz tool specializes in hyper graphs. However, Ontoviz was not considered for implementation as Prefuse incorporated the features provided by OntoViz and was more dynamic.

Prefuse Discussion

Prefuse is a set of software tools for creating rich interactive data visualisations. Prefuse supports the design and implementation of novel visualisations.
Another driver for using Prefuse was the successful implementation of Prefuse in other projects such as Vizster [5] and SocialAction [19]. The social network graphical representations in these projects had overwhelmingly better quality. We also found that Prefuse's highly-customizable rendering and animation greatly supported the exploration of design ideas. This suited our project as the course site forums are not conventional social networks as forums are highly connected. SONET requires manageable and useful features to explore the highly connected forum social networks.

4.2 Layouts implemented

4.2.1 Graph View

The graph view implements the Force Directed algorithm which attempts to form communities by using forces.
This popular formatting algorithm is also known as Newman's community identification algorithm [9]. Newman's community identification algorithm is used in many visualisation systems because it is fast enough to support interactive real-time

adjustments. This algorithm has been successfully applied in the SocialAction and Vizster social network visualisation systems.
Newman's community identification algorithm identifies group structures based solely on link analysis. "The algorithm employs hierarchical agglomerative clustering, first placing each node in its own community, and then greedily merging groups based upon a metric that attempts to maximize within-cluster linkage while minimizing between-cluster linkages" [9].

4.2.2 Radial View

In the radial tree layout a single node is placed at the centre of the display and all the other nodes are laid around it. The entire graph is like a tree rooted at the central node. The central node is referred to as the focus node and all the other nodes are arranged on concentric rings around it. Each node lies on the ring corresponding to its shortest network distance from the focus. Any two nodes joined by an edge in the graph are referred to as neighbours. Immediate neighbours of the focus lie on the smallest inner ring, their neighbours lie on the second smallest ring, and so on.

The Radial tree layout has been applied to visualisations of social networks and of the Gnutella file-sharing network [21].

The conventional implementation of social networks is implemented using the force directed algorithm. This algorithm identifies communities based on the properties of the social network. However, forum data is not the conventional social network as nodes are more highly connected to one another. The challenge of this problem is then how to provide good visualisation of the social network derived from a forum when used by a large class. Features are needed that allow users to explore the social network and make useful observations.

Basic class diagram of layout implementation

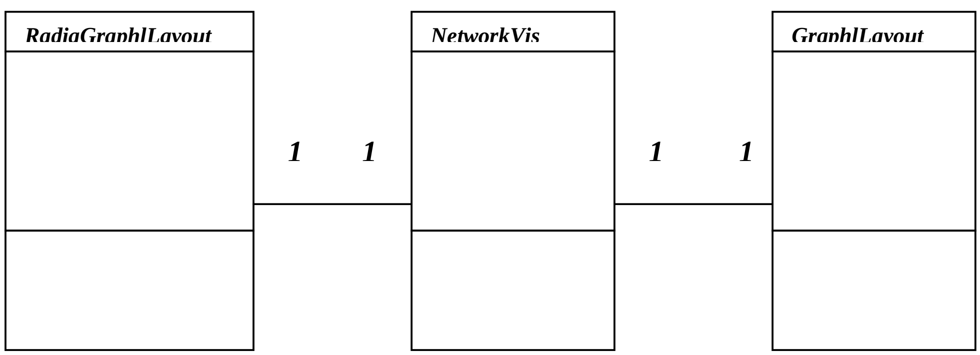


Figure 14 Basic Visualisation Class Diagram

The NetworkVis class is the main class. It is the driver class for the project. The radial graph view is implemented in the RadialGraphLayout class. This class can be added to the NetworkVis class as a panel component.
The graph view is implemented in the GraphLayout class. This class can be added to the NetworkVis as a panel component.

Figure 15 below depicts the process flow of changing the source data into the social network visualisations observed by the user.

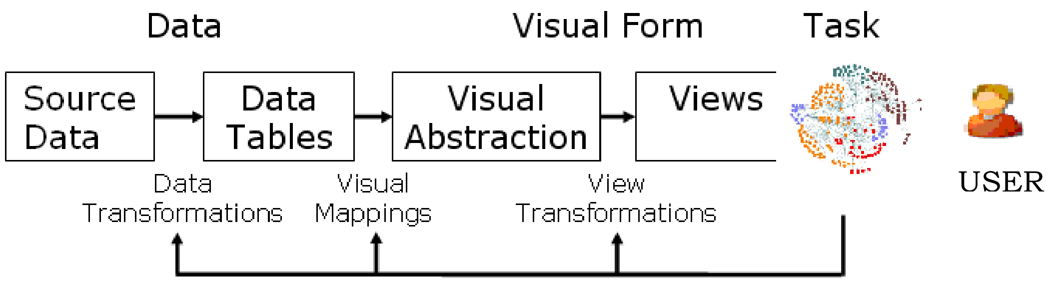


Figure 15 Data flow from source to user visualisations

4.3 Knowledge Representation

System integration is facilitated by using a shared intermediary database. This database is implemented in MySQL. MySQL [13] is a popular open source database because of its consistent fast performance, high reliability and ease of use.
The database tables used by the visualisation subsystem are as follows.

The tables coming from Database subsystem:

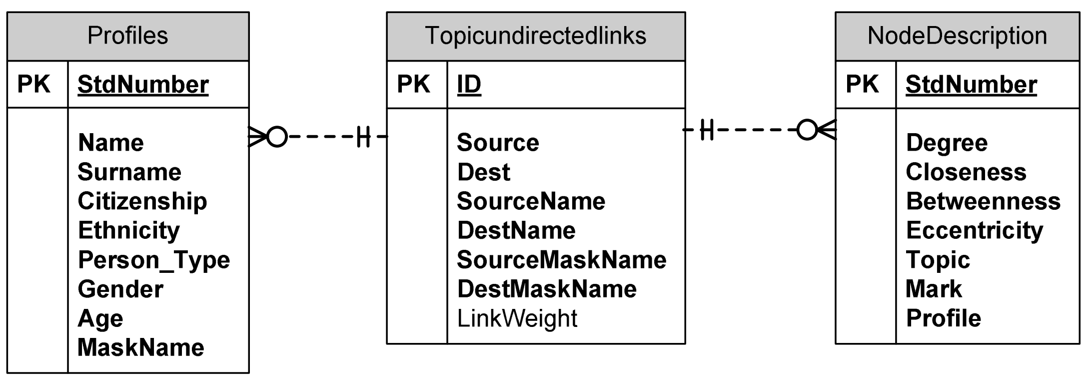


Figure 16.1 Social Network ER diagram

Brief description:
TopicIndrectedLinks is used to generate social network graph.
NodeDescription and Profiles tables are used to fill in node details.

The tables coming from the Bayesian network subsystem:

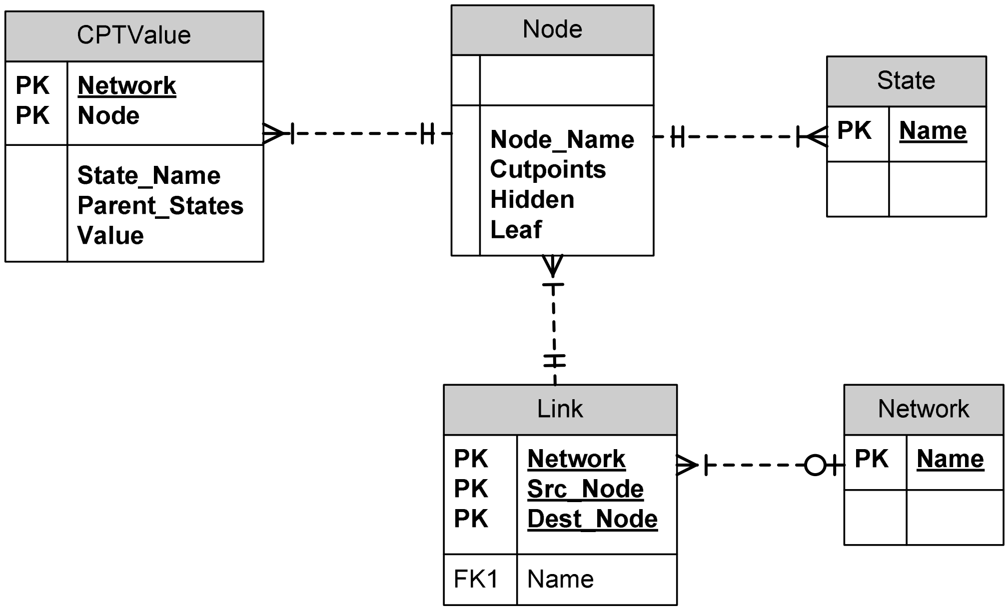


Figure 16.2 Bayesian Network ER Diagram

Brief description:
Each node contains a CPTValue Table. A node has states which are stored in the State table. Nodes are linked by the Link table. Finally each Bayesian network is uniquely defined by its name pointing to linked nodes.

4.4 Inference Engine

Two inference algorithms were considered in implementing the inference engine namely, Variable Elimination algorithm and Judea Pearl's Belief message passing algorithm. It was found that Variable Elimination did not work correctly in all cases [4] because the output depended on the elimination order. An optimal elimination ordering is one that results in the least complexity, but the problem of finding an optimal elimination ordering is NP-complete [4]. This swayed us to use the message-passing algorithm. Another issue was that structure learning is implemented in the Bayesian network subsystem. Structure learning requires an inference algorithm implementation that can handle complex Bayesian networks. This issue will be further discussed in the testing section.

While an overview of Pearl s message passing algorithm has been given in the Background chapter, this section gives a detailed description of the how the algorithm was implemented.

4.4.1 Overview of Inference structure

The Bayesian network subsystem contains a ControllerImpl class that handles Bayesian network retrieval from the database. Because of the subsystem modularity this ControllerImpl (Façade) class loads the network from the database.
The uploaded network can now be loaded into the Inference Engine through the InferenceInterface (Facade) class as show in figure 10.2 Inference Class Diagram.

The implementation of the Judea Pearl's algorithm is discussed below.

Every node contains a *belief*, π and λ array. After the network has been created, boundary conditions can be set. These are:
Root nodes values are set to the same values found in the CPT.
All nodes λ arrays are set to 1 in each position.

The network is now ready to begin inference for student mark predictions. Before this process is explained, a detailed description of how the π and λ messages were implemented is given.

The message passing is dived into two sections forward and backward inference.

Forward Inference (π *messages*)

1. $\pi_{Y_j}(x) = P(x|\mathbf{e}) / \lambda_{Y_j}(x)$ (Message sent from X to Y_j)
2. $\pi(x) = \sum_{u1..un} P(x|U_1..U_n) \prod_{i=1..n} \pi_X(U_i)$

$P(x|\mathbf{e})$ is the belief of X given evidence e.
Once all the messages have been sent, the second formula can be used to calculate the π value of each node. This is calculated through matrix multiplication.

Backward Inference (λ messages)
The formulae below are used to propagate the evidence backward through the network.

1. $\lambda(x) = \prod_{j=1..m} \lambda_{Y_j}(x)$
2. $\lambda_{Y_j}(x) = \sum_{Y_j}[\lambda(Y_j) \sum_{v1..vp} P(Y_j|x,v1..vp) \prod_{k=1..p} \pi_{Y_j}(Vk)]$ (λ value sent to destined parent)

The first formula multiplies all λ messages received.
The second formula multiplies all the π values coming from parents other than the destined parent into a permutation vector. This formula uses matrix multiplication and multiplies the child's evidence and beliefs with the permutation vector. These values are then summed according to the states of the destined parent node. The resulting λ message is then passed to the parent.

Figure 13 represents the message passing between parent and child nodes.

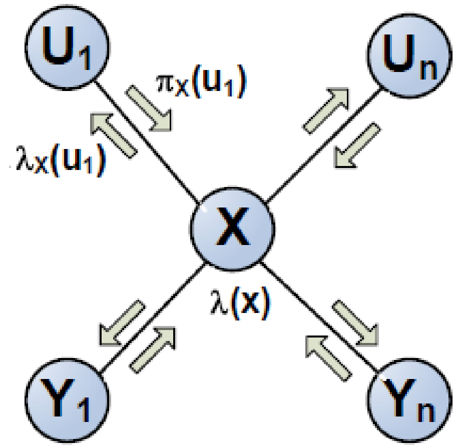


Figure 17 Evidence Propagation using message passing

5. TESTING AND EVALUATION

5.1 Inference Testing

The inference accuracy tests were done by comparing the implemented Judea Pearl Belief propagation algorithm in SONET to JavaBayes version 0.347. JavaBayes [20] is a well known inference tool. The inference algorithms implemented in JavaBayes are Variable Elimination and Bucket tree elimination.

The Bayesian network subsystem implemented a structure learning algorithm to discover the best Bayesian network structure corresponding to the given data. This structure learning requires the inference engine to be more robust and should be able to handle any model, and not only the naïve bayes classifier model. Therefore the inference engine required more rigorous test cases.

The results of the SONET inference engine were correct. A number of tests were run, each testing a different aspect of the inference. The first test was on a naïve bayes classifier.

Test case 1: Naïve bayes classifier.

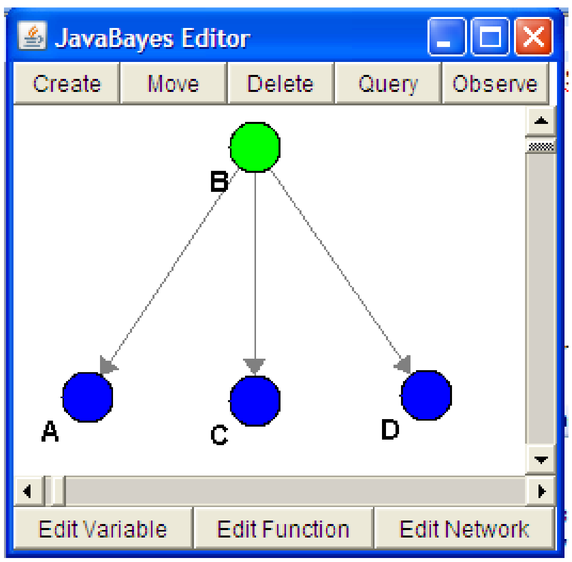
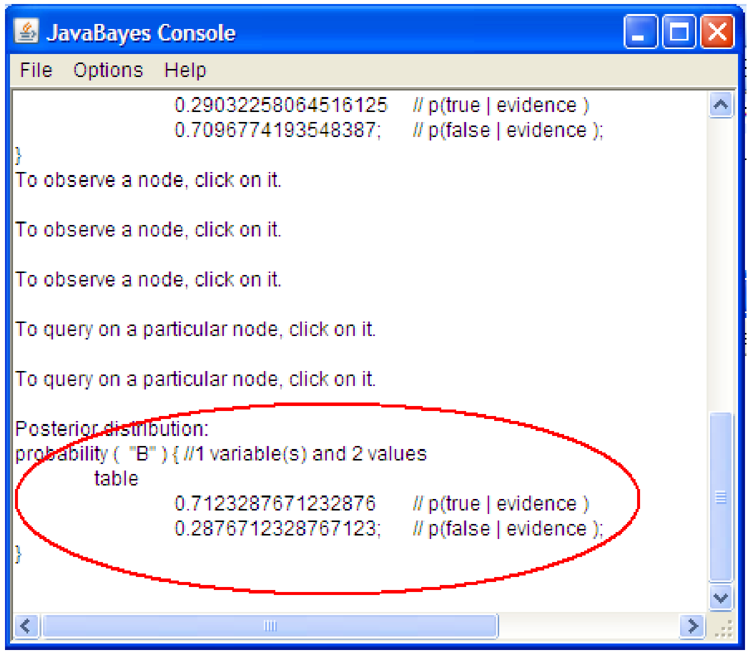


Figure 18.1 Naïve bayes structure as shown in JavaBayes

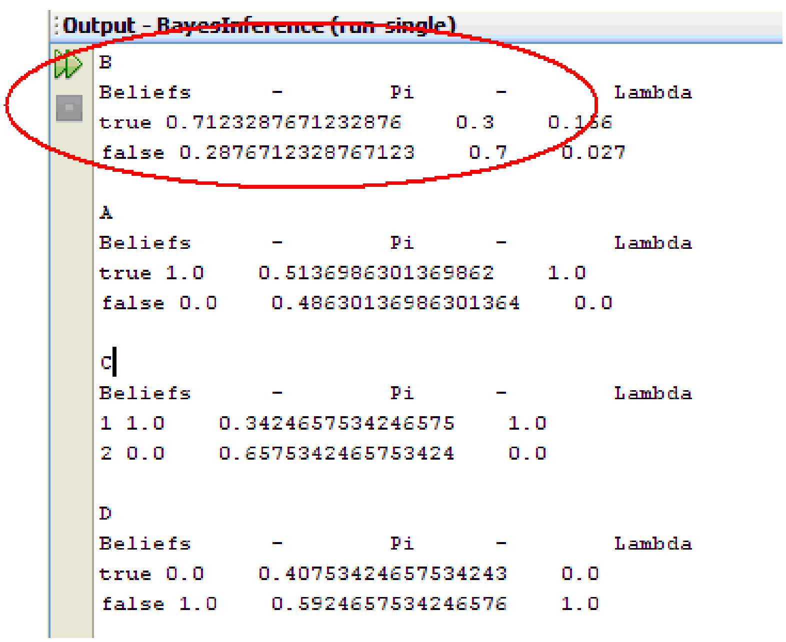
Note: In test case 1 we are testing a naïve bayes classifier. All the leaf nodes are instantiated. This ensures backwards inference is working correctly.



```
JavaBayes Console
File  Options  Help
                    0.29032258064516125    // p(true | evidence )
                    0.7096774193548387;    // p(false | evidence );
}
To observe a node, click on it.

To observe a node, click on it.

To observe a node, click on it.

To query on a particular node, click on it.

To query on a particular node, click on it.

Posterior distribution:
probability ( "B" ) { //1 variable(s) and 2 values
          table
                    0.7123287671232876     // p(true | evidence )
                    0.2876712328767123;    // p(false | evidence );
}
```

Figure 18.2 Output from JavaBayes

SONET Bayesian Inference output



```
Output - BayesInference (run-single)
B
Beliefs      -         Pi       -        Lambda
true 0.7123287671232876    0.3     0.156
false 0.2876712328767123    0.7     0.027

A
Beliefs      -         Pi       -        Lambda
true 1.0    0.5136986301369862    1.0
false 0.0    0.4863013698630136 4    0.0

C
Beliefs      -         Pi       -        Lambda
1 1.0    0.3424657534246575    1.0
2 0.0    0.6575342465753424    0.0

D
Beliefs      -         Pi       -        Lambda
true 0.0    0.4075342465753424 3    0.0
false 1.0    0.5924657534246576    1.0
```

Figure 18.3 Output from SONET inference

The queried values for node B are exactly the same.

Test case 2: Structured Bayesian network (i)

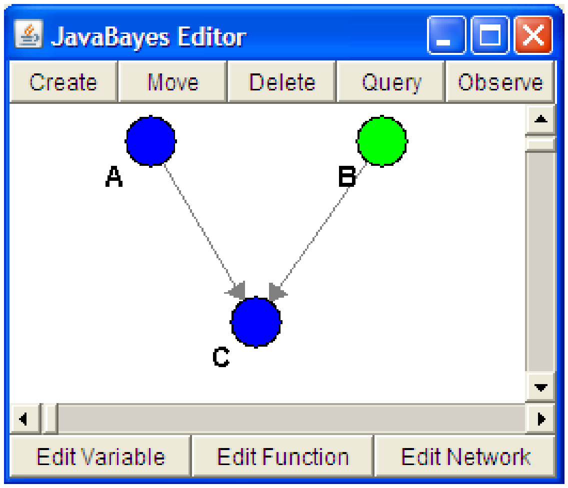


Figure 18.4 Structured Bayesian Network (i)

In test case 2 we are testing both forward and back inference.

JavaBayes Output

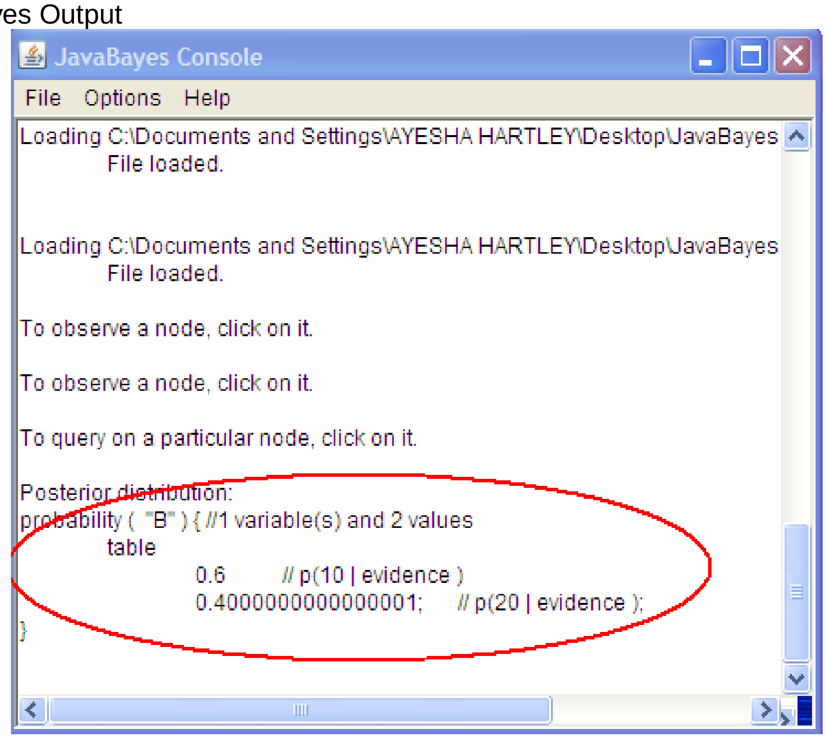


Figure 18.5 JavaBayes output for Bayesian Network (i)

SONET inference output

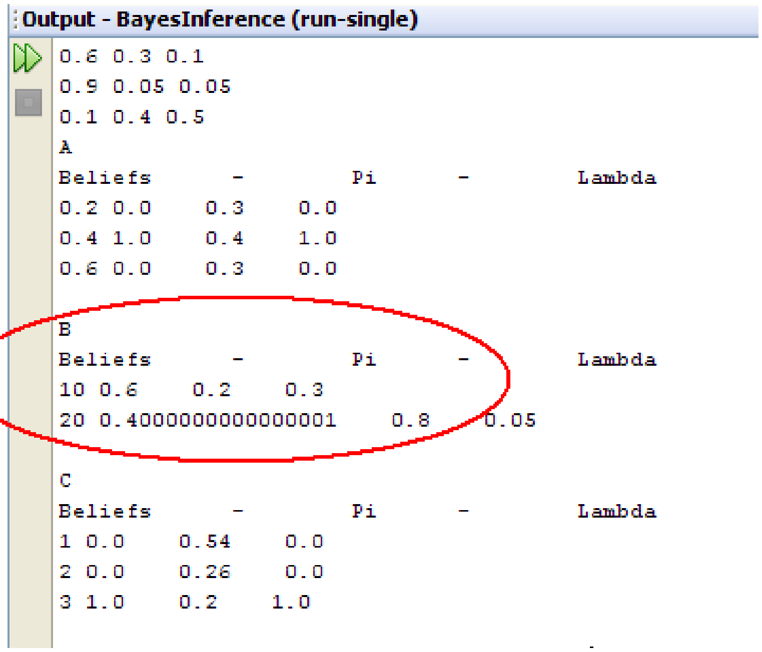


Figure 18.6 SONET inference output for Structured Bayesian network (i)

The output values for test case 2 proved to be exactly the same.

Test case 3: Structured Bayesian network (multilevel) (ii)

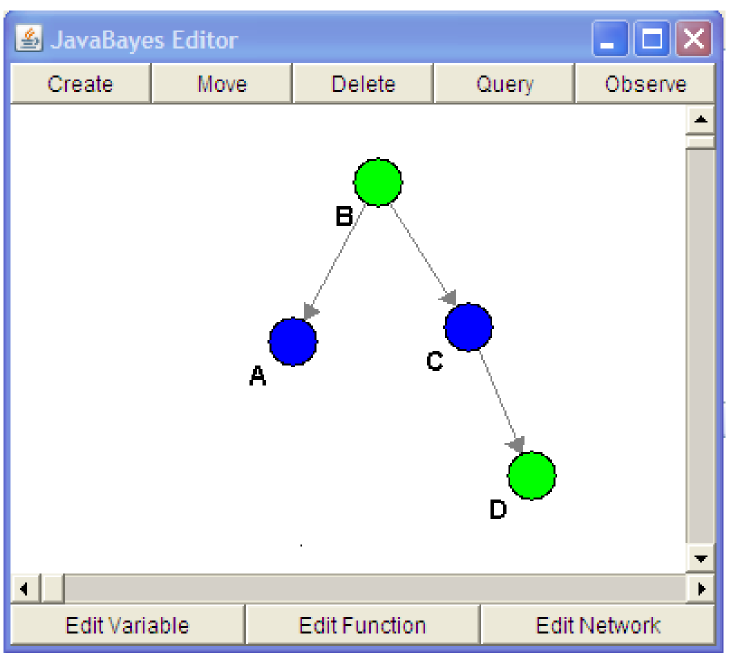


Figure 18.7 Multilevel Structured Bayesian network (ii)

Note: In test case three we are testing both forward and back inference

JavaBayes Output

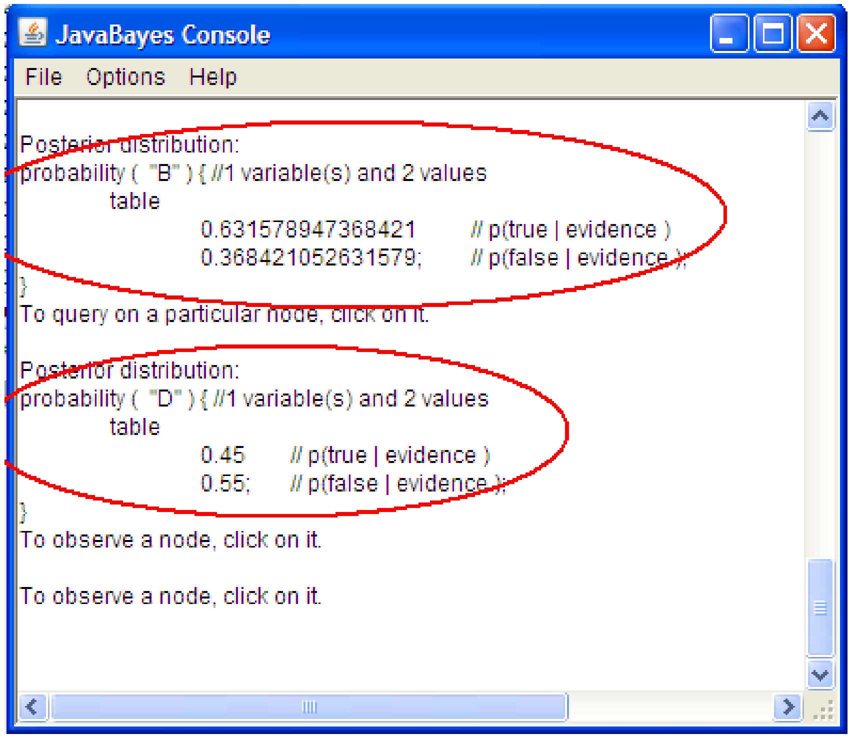


Figure 18.8 JavaBayes output for multilevel test case

SONET Inference output

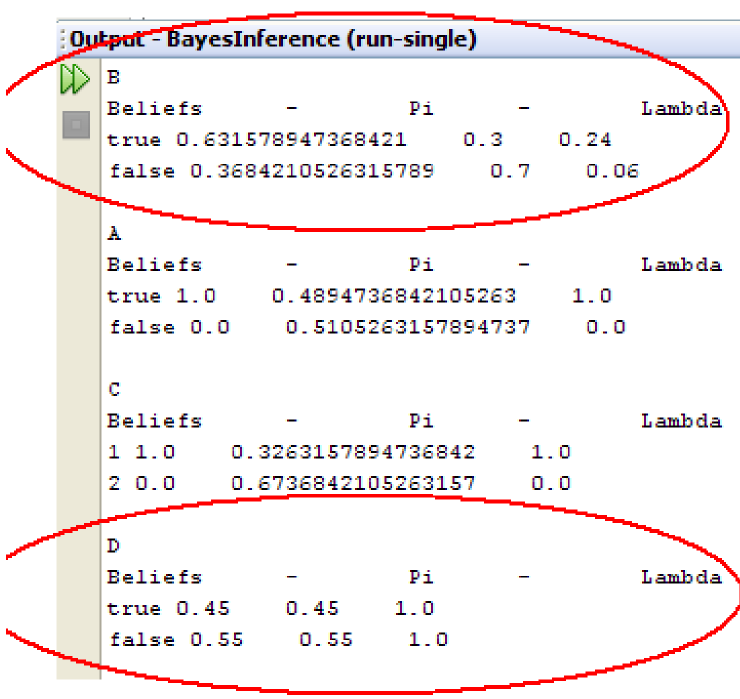


Figure 18.9 SONET inference output for multilevel test case

The output values for test case 3 proved to be exactly the same.

5.2 Testing and Evaluation of the Interface

Evaluating the Graph view and Radial View to represent the forum social networks

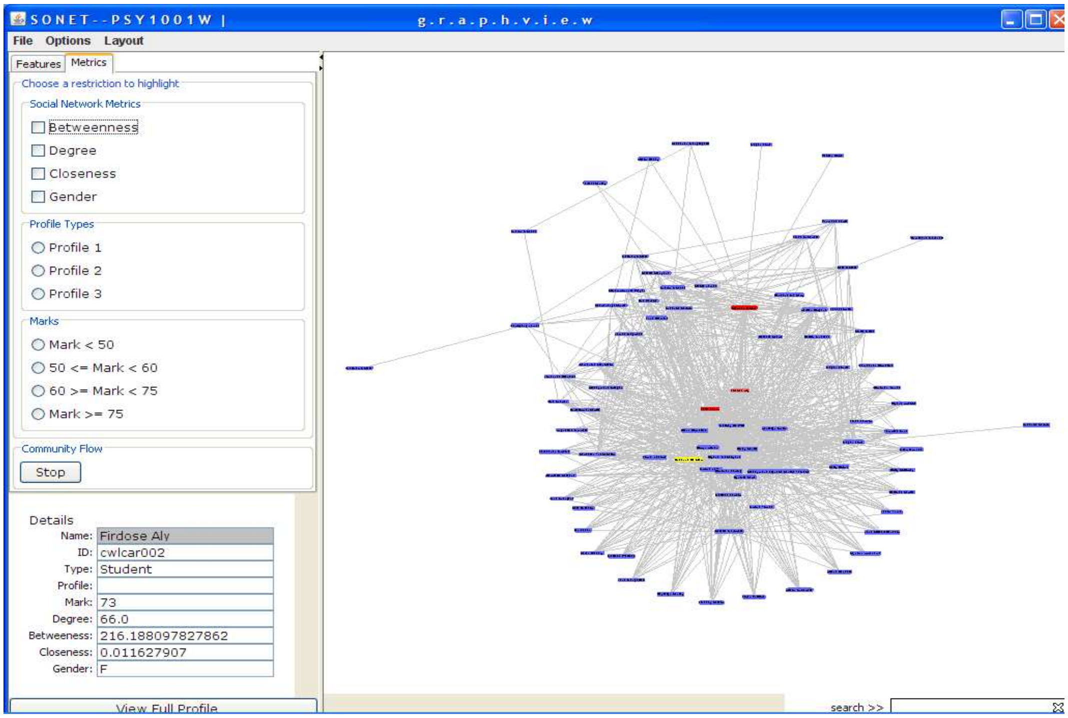


Figure 19.1 Graph View

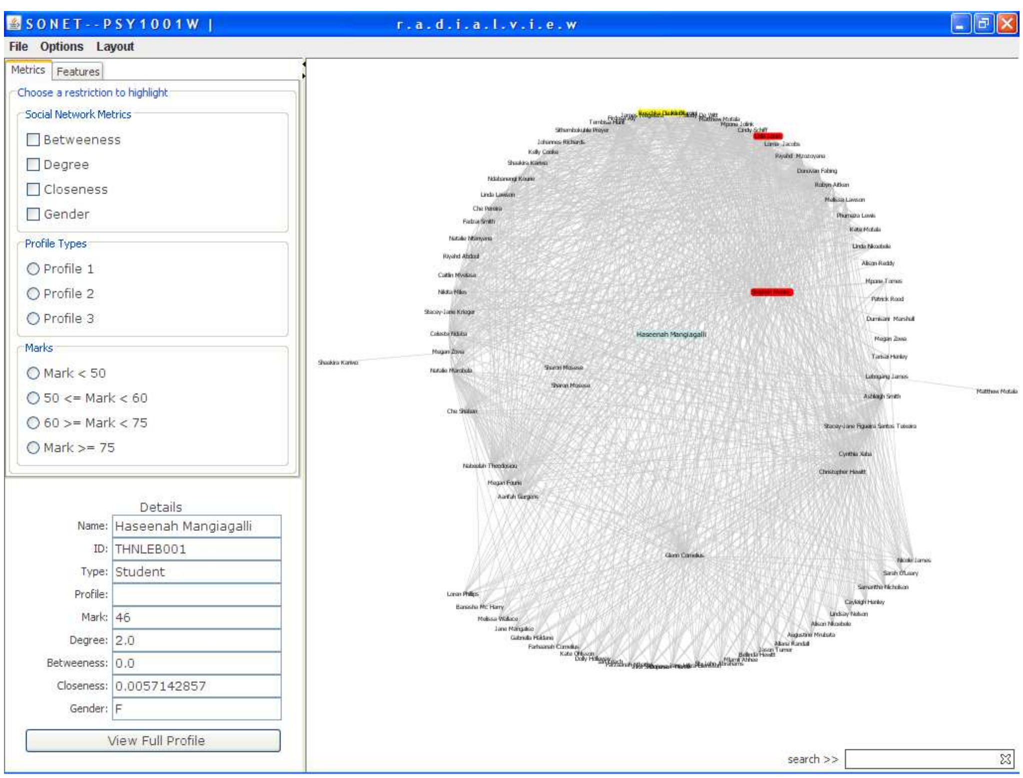


Figure 19.2 Radial View

Screen shots of the SONET layouts are shown above in Figures 19.1 and 19.2.

The purpose of this study was to compare the representation of the social networks using the Graph Layout and the Radial Layout. Most social network tools use the Graph view layout (e.g. SocialAction and Vizster). The comparative study was to determine which layout allows users to better observe social network patterns. Both layouts have the same query highlighting feature. Both layouts are implemented in SONET.

5.3 Hypothesis

Based on research into social network visualisations, it is believed that the graph view layout in Figure 19.1 will be easier to visualize social network patterns than the radial layout in Figure 19.2. The assumption also includes the filtering feature added to the graph view layout. Also we feel that the natural layout of the social network in a graph view allows users to identify more easily social network patterns.

The following hypothesis and null hypothesis were formulated.

H_1: The graph view representation of the social network will be easier to use than the radial view representation.

H_0: The radial view representation of the social network will be easier to use compared to the graph view representation.

The criteria to reject the null hypothesis are based on three factors:

1. The time taken for users to complete tasks. The author assumes the faster a participant completes a task, the easier the system is to use. This is a quantitative measure that can be studied for statistical significance.
2. How the participants understand each of the graph visualisations, shows how well the participants interpret the social networks. This is a qualitative measure, and is subject to the authors' interpretation of facts.
3. Which layout the participants prefer the most. The results are qualitative as well as quantitative.

5.4 Effects of learning

In order for the evaluation to hold any weight statistically, it is essential that the participants in the experiment are taken from students and lecturers that have exposure to forums. They should have the same age profile and have similar levels of literacy and similar amount of exposure to computer technology. For this experiment, a basic knowledge of computers is all that is required. The age group should be between 20 and 25, the age group of most students.

The effects of learning are relevant to this experiment. Students who work with one interface become more skilful. The order of exposure to the systems can affect the results of the experiment. To counter the learning effect, participants were split into equal groups and presented with the two layouts in different orders.

5.5 Pilot Study

The pilot study was conducted to test the effectiveness of the tasks allocated to the participants. This study did not test the hypothesis in any way, but indicated flaws in the testing procedure. The test was conducted on a small sample. Students selected were more senior. A third year science student and a third year Humanities student were selected. Each frequently use online site forums

5.5.1 Tasks

To compare the views and find out the comparative ease of use, each participant needed to complete a series of tasks. These tasks should test their understanding of social networks. The tasks should also test their understanding of social network metrics and how it fits in with student profiles and marks.

5.5.2 Method

The participants were tested individually on a computer containing both representations. Participants were requested to sign User Consent Forms viewable as Appendix A. The participants were given a set of tasks to complete while the author recorded the time taken to complete each task.

Post usage, the participants were asked to explain the social network and how they found it to be useful. They were requested to choose which view they found easier to use.

5.5.3 Initial Results

Initial user tests showed that users made use of check boxes incorrectly. Check boxes are used to input/specify queries. They would check a new box and then uncheck the previous one, but the system unchecks the previous one automatically. This results in the user unchecking the intended box. The check boxes were then replaced with radio buttons which suit this feature much better.

Another problem was that users struggled to distinguish between the search highlight colour and the query highlight colour. This problem was solved by taking two completely contrasted colours.
Questions on explaining social positions were too vague and had to be better articulated.

Participants found it quite easy to interpret the social network metrics and came up with good solutions to questions on finding a social pattern. “Who are the knowledge hubs, and why?”

One participant selected high degree and first class pass. The participant’s reason is that a student in a forum that is frequently active and has good marks can be trusted on his opinions.

The other participant selected high closeness and high betweenness. This participant’s reason was that marks are not the only factor to having knowledge, but also good exposure.

5.6 Main experiment

The methodology was similar to the pilot study. The task lists was simplified to only questions that were necessary. The participants were tested individually on a notebook computer. Participants were requested to sign User Consent Forms viewable in appendix A. The participants were given a background of the project. The purpose of the study was explained. The social network metrics betweenness, closeness and degree were explained using appendix C adapted from [17].

Each participant was given a quick overview of the layout and features were explained. The participant was given three minutes to explore the layout and get a grasp of the tool.

Participants were spilt into two groups, one group completed the tasks with the graph view first and then the radial view while the second group completed the process in reverse.

Post usage, the participants were asked to come up with a solution to "Who are the knowledge hubs, and why?" and also what observations they could derive from viewing the social networks. The idea was to see if users of SONET can make meaningful observations and find social network patterns from the forums. They were then requested to choose which view they found easier to use.

5.6.1 Participants

Ten participants were used in this experiment. All participants were aged between 20 and 25 years of age. Their backgrounds were not specific to a certain faculty but all had frequent exposure to using online forums. The male to female ratio was six to four.

5.6.2 Equipment

1. Stop watch
2. Notebook with SONET
3. Task list pages pen and pencil

5.6.3 General results

After completion of the task lists the participants were asked which view they preferred. The findings are shown in Figure 20 below from Table 2 in appendix D. This shows that participants preferred the radial view more. Participants thought of many interesting ways to query the social network. Most people used a combination of social network metrics and student marks to come up with a solution to find "knowledge hubs". There were a few participants who initially felt that there would be no correlation between student marks and social network metrics or a slight correlation. All these participants changed their mind once they interacted with the SONET system.

Most participants made use of the query system to describe who "knowledge hubs" are. Others said lecturers and tutors, and were requested to use the query panel to find student "knowledge hubs".

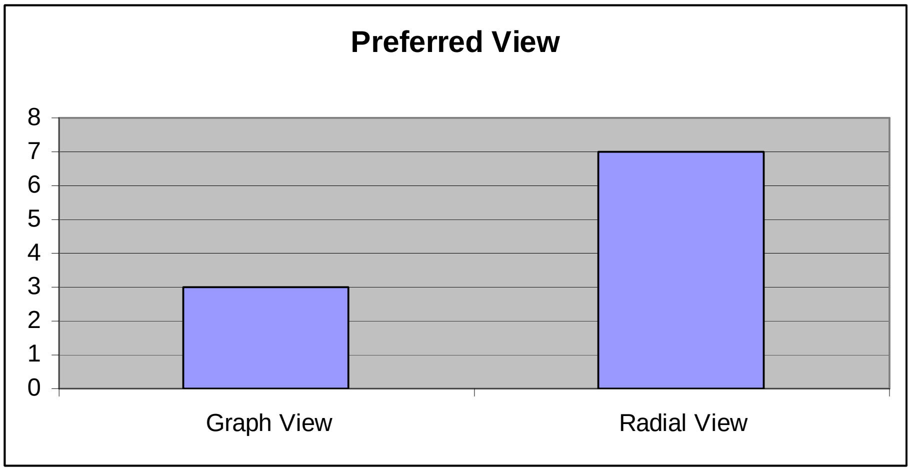


Figure 20 Interface preference

Results Task set 1 – Graph clarity

The participants could identify the different types of nodes easily. They adapted to using the zooming and panning features quite quickly. Participants found it easy to identify the different member types of the network i.e. lecturer, tutor and student. The social network roles (central nodes and clusters) could be quite easily explained by the participants when observing the network.

Results Task set 2 – Data Searching

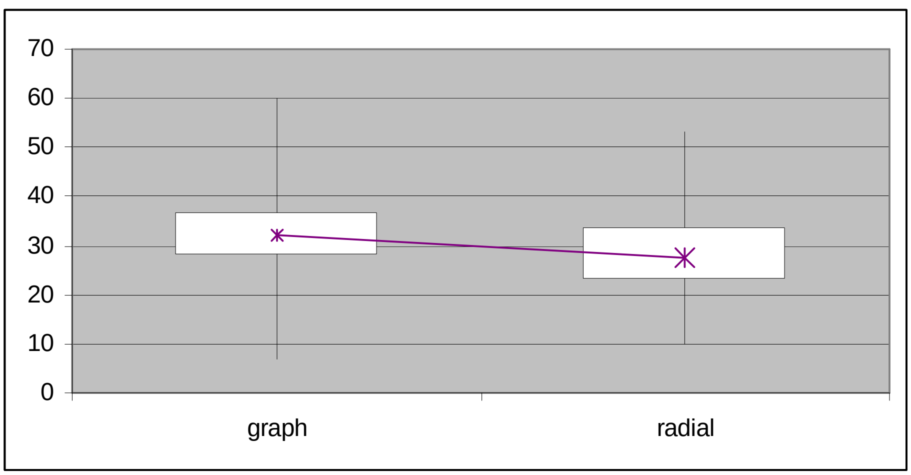


Figure 21 Box and whisker comparing performance times of each interface

From the plot in Figure 21, one can see that the medians of the graph view and radial view are close to 32 and 27.5 respectively. The difference of 4.5 seconds shows that users are almost similar but they are better with using the radial view. A general measure is that if one box can fit into the other. If this occurs, the results are not significantly different. The graph view box does overlap the radial box so there is almost a statistical difference.

Results Task set 3 – Social network analysis

All participants could see the difference in social positions of the different student profiles. All participants could create a query that they thought would find “knowledge hubs”. Theses queries were all valid and not random thoughts from the participants.

5.7 Evaluating the SONET interface

The interface of the SONET system was created using an iterative approach. Prototypes were created through much iteration. The system was developed primarily in accordance to super users wishes and hence the system could conform to their ideals. General usability was tested with super users and a walk through of the system was done.

5.7.1 Participants

Two super users.

5.7.2 Equipment

1. Notebook with SONET
2. Pen and note pad.

5.7.3 Method

The functional outcomes of the system were discussed with the users and what achievements have been made by the developing team. The developer explained the idea behind the system, and what assumptions were made to generate the social network. An explanation was given on how links are created between nodes.

As the network was explored the users were asked what areas of social network analyses they were interested in. This was done to give the developers better feedback on what use certain features can provide. The participants were given a view of a particular section of the forum. This section of the forum was used to explore the features of SONET.

The super users noted that they were not interested in individual nodes but mainly in finding patterns within the social network. In the case of serious intervention where students (nodes) are performing poorly SONET should be able to highlight these students and a drill down on their details should be provided.

5.7.4 Results

Queries
Super users as well as participants requested more queries to be added to the query panel, such as high and low degree. It is not always easy to view outliers in the radial view.
Also, participants commented that a count be shown for the number of nodes highlighted by a query.

Menu names
Some participants requested larger fonts and better background colouring for details box in the information panel.

General

Participants were asked if they thought this tool would help lecturers and students understand a class better. They felt that overall SONET is a useful tool that should be implemented into online learning systems. It will allow students to better identify knowledge hubs and improve communication of course work.

6. DISCUSSION

6.1 Layout (Hypothesis result)

There is no statistical significance for the author to accept or reject the hypothesis. However, results show that the radial view was the more popular of the two views. Participants felt that it was much simpler because it had less features and was “easier on the eye”.

6.2 SONET Interface

Overall, the participants felt that the system was easy to use. Participants were particularly interested in the queries that displayed the nodes in the social network that interested to them. Most of them found that the information panel was very useful.

6.3 Inference tests

After running multiple inference test cases it was found that the SONET inference provides exactly the same values as JavaBayes. We can safely conclude that the inference implementation was a success.

7. CONCLUSION

7.1 Success of the SONET visualisation

In conclusion, the author feels that the visualisation section of the SONET project was a success. The requirements of the system were met and super users were impressed with the system and satisfied with the results. The interface met all requirements for the system and was found to be intuitive enough for lecturers and students to use. The inference implementation was a success and its use in Bayesian networks can be trusted. The author is confident that the SONET system will be valuable to lecturers interested in teaching with technology.

7.2 Lessons Learnt

From this project, the author has come to understand the difficulty in communicating among members of a team and the necessity for effective risk management. The integration of the system is only achieved smoothly if communication occurs from the initial stages of project. Integration of the project was left too late. Keeping in contact with stakeholders is essential and those people appreciate the effort for keeping them updated with the project status.

7.3 Summary of the project

Overall the author feels that the visualisation and inference of the project was a success. Participants enjoyed the experience of using the SONET system and some were impressed because it was the first time they were exposed to such a system. The author feels that the project idea is novel as there is no exact system. Another key factor that made the SONET project team realise that the project was a success was the invitation to present the SONET project at the Centre of Higher Education at UCT.

8. FUTURE WORK

This section presents possible extensions to the SONET system.

8.1 Additional functionality

The developer has endeavoured to implement link weights to show link strength between nodes of the social network. However, due to time constraints this was not possible given the length of time for an honours project. The modularity of the SONET system allows for other layouts to be adapted to the system.
Clustering of communities within the social network is a feature that seems to be promising to reveal insights.

8.2 Integration into Vula

There has already been some interest in the possibility of integrating SONET into Vula, to allow lecturers and course convenors to analyse the communication patterns within their classes. Fortunately, the use of the Java platform would facilitate the necessary movement from a desktop to a server environment. The visualisations created can easily be transformed into Java applets that would integrate with Vula sites.

9. REFERENCES

[1] Bederson, B. B, Grosjean, J., and Meyer J., "Toolkit Design for Interactive Structured Graphics", IEEE Transactions on Software Engineering, 30, pages 535-546, 2004.

[2] Bhoopchand, A., "Bayesian Classification of Students based on Participation in eLearning Forums: An empirical analysis of structure learning algorithms", Computer Science Honours, University of Cape Town, November 2008.

[3] Diez F.. *Local Conditioning in Bayesian Networks.* Department of Artificial Intelligence, Madrid, Spain, 1996

[4] Guo, H., Horvitz, E., Hsu, W. H., and Santos, E., *"A survey of algorithms for real-time Bayesian network inference"*. AAAI/KDD/UAI-2002 Joint Workshop on Real-Time Decision Support and Diagnosis Systems, Edmonton, Alberta, Canada, 2002.

[5] Heer, J., Boyd, D., "Vizster: Visualizing Online Social Networks" IEEE Symposium on Information Visualisation (InfoVis), 2005

[6] Heer, J., Card, S. K., and Landay, J.A., "Prefuse: A Toolkit for Interactive Information Visualisation", ACM Conference on Human Factors in Computing Systems, 2005.

[7] JavaBayes "Bayesian Networks in Java". Available from http://www.pmr.poli.usp.br/ltd/Software/javabayes/; Accessed 31 October 2008

[8] JUNG: Java Universal Network/Graph Framework. Available from http://jung.sf.net/; Accessed 10 July 2008.

[9] Le Grand, B., "Topic Maps Visualisation" in Information Visualisation Proceedings. Sixth International Conference**,** pages 344- 349, 2002

[10] Mengshoel, O. J., *Efficient Bayesian Network Inference: Genetic Algorithms, Stochastic Local Search and Abstraction.* Ph.D. Dissertation, Department of Computer Science, University of Illinois at Urbana-Champaign, May, 1999.

[11] Mengshoel, O.J., and Wilkins, D.C., "Visualizing Uncertainty in Battlefield Reasoning Using Belief Networks." ARL Federated Laboratory Advanced Displays and Interactive Displays Consortium, Advanced Displays and Interactive Displays First Annual Symposium, Adelphi, MD. January 28-29, 1997.

[12] Murphy, K.P., Weiss Y., and Jordan, M. "Loopy belief propagation for approximate inference: an empirical study". In Proceedings of Uncertainty in AI, pages 467-475, 1999.

[13] MySQL, "My SQL Open source database" Available from http://www.mysql.com/why-mysql/ ; Accessed 26 October 2008

[14] Newman, M. E. J., "Fast algorithm for detecting community structure in networks." Physical Review E, 69, 2004.

[15] Network Bench – "Visualise data". Available from https://nwb.slis.indiana.edu/community/?n=VisualizeData.RadialTree; Accessed 25 October 2008.

[16] O'Madadhain. J, Fisher. D, Smyth. P, White. S and Boey. Y, " Analysis and Visualisation of Network Data using JUNG", Journal of Statistical Software, 2005.

[17] Orgnet.com, "Social Network Analysis software & services for organizations, communities, and their consultants". Available from http://orgnet.com; Accessed 26 October 2008.

[18] Pearl, J., Fusion, propagation and structuring in belief networks. UCLA Computer Science Department Technical Report 850022 (R-42); *Artificial Intelligence*, Vol. 29, No. 3, pages 241-288, September 1986.

[19] Perer A, Schneiderman B. IEEE. Transactions on visualisation and computer graphics, Vol 12, No. 5, September/October 2006.

[20] Piccolo "Piccolo Toolkit – A Structured 2D Graphics Framework", Available from http://www.cs.umd.edu/hcil/piccolo/ ; Accessed 16 July 2008.

[21] Potgieter, A., "Bayesian Networks as Hyperstructures", Honours Course Agents 2008.

[22] Prefuse "the prefuse visualisation toolkit", Available from http://prefuse.org/; Accessed 16 July 2008.

[23] Rose, F. L., "Analysis of Three Bayesian Network Inference Algorithms: Variable Elimination, Likelihood Weighting, and Gibbs Sampling" 2004

[24] Weiss, Y. and Freeman, W., "Correctness of belief propagation in gaussian graphical models of arbitrary topology". In NIPS, volume 12, 1999.

[25] Zapata-Rivera J.D; Greer J.E 2001 Externalizing Learner Modeling Representations AI-ED 2001 Workshop, San Antonio, Texas, May 2001.

Appendices

Appendix A

Usability Evaluation Consent Form

Supervised by: M Kaashief Hartley
Email: hrtmog005@uct.ac.za

Project Purpose and Procedures
The purpose is to test the usability and performance of two different interfaces. You will be given a brief tutorial on Social Network Analysis and will be asked to complete a few tasks on each interface. Upon completion of the tasks, you will be asked a few questions regarding usability. The duration of this evaluation is approximately twenty minutes.

Confidentiality
The personal information obtained from the interview and the experiment will not be disclosed to any third party and will only be used for this assessment.

Contact Information About the Project

Dr Anet Potgieter
Department of Computer Science
University of Cape Town
Private Bag
Rondebosch
7700
Tel: +27 21 650 4056
Email: anet@cs.uct.ac.za

A/Prof Dr Sonia Berman
Department of Computer Science
University of Cape Town
Private Bag
Rondebosch
7700
Tel: +27 21 650 2668
Email: sonia@cs.uct.ac.za

Consent
The developers intend for your participation in this experiment to be pleasant and stress-free. Your participation is completely voluntary and you may refuse to participate or withdraw from the experiment at any time.

Signature...................

Appendix B

Interview Questions and Task List

Questions on understanding

1. Have you participated in forums e.g. Vula forums before?
Never / rarely / frequently
Why do you participate? E.g. for learning or chatting

2. Do you know what a social network is? If so, can you explain or draw one?
(and explain what it means to the interviewer, I presume)

3. How would you classify a good member of a social network?

4. To what extent do you think there's a correlation between forum participation and performance? Why?

5. Would you want to know how the social networks for the forums you participate in looks like? How would you prefer it to look and what features would you want on it? Why?

Task set 1

1. Name the lecturers and tutors? (All, there aren't many. Testing legibility, zooming and panning features).

2. How are they positioned in the network? How would describe this relative to the lecturers and tutors social position within a classroom.
Exactly / similar / not quite / totally different

3. What do you observe in the network? Any separate clusters, central nodes.
Would this be a good representation of the actual forum? Why?

Task set 2

1. Find the personalized network of Tembisa Hunt (Masked Name).

2. What is her final exam mark?

3. What profile group is she in?

Task set 3

1. Which are the profile 1 students?

2. How many first class (Mark > 75%) students are in profile 1?

3. How many second class (75 > Mark > 60) students in profile 2?

4. Which nodes would you say are the knowledge hubs and why? Could you display this by selecting them with a social network query? (if the participant has not done so yet)

Was participant able to do so (Y / N)

5. Who are the outliers? Individuals participating the least in the social network?

Post task questions

1. How easy did you find it to identify the nodes from the Social Network?
Graph View
Hard Easy
1 2 3 4 5..6..7
Radial View
Hard Easy
1 2 3 4 5..6..7

2. Was the information panel helpful?
Never Always
1 2 3 4 5..6..7

2. Did you find that you got lost looking for a node once the graph had moved? Why?
Graph View

Radial View

3. Which interface did you prefer to use? (graph view or radial view) why?

4. Which interface was easier to use? Why?

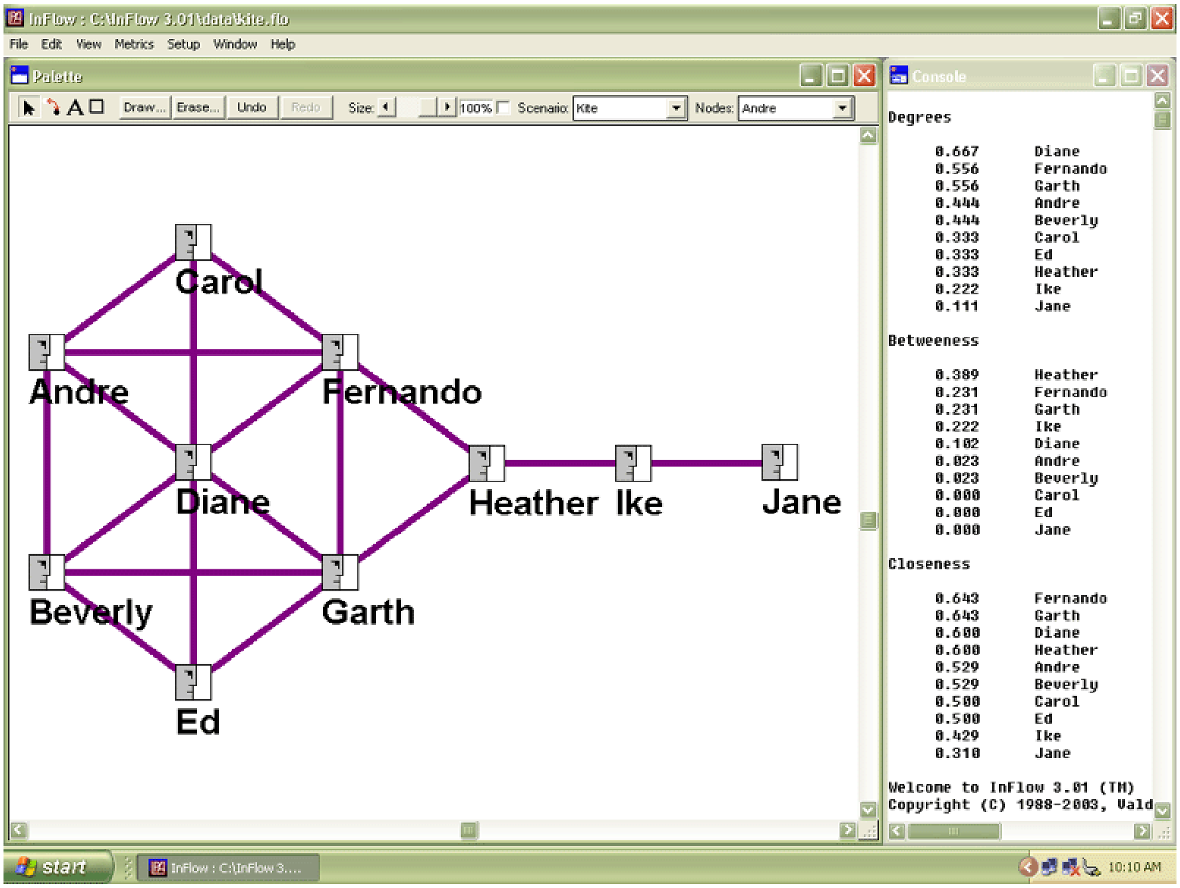
5. Which view provided better clarity and navigation of the social network? Why?

6. Do you think this tool would help educators or learners understand a class better (Lecturer) and improve individual students? Why?

7. Would you want to use this tool in practice? (Lecturer)

Appendix C

Adapted from <orgnet.com>



Appendix D

Results from the user testing.

Table 1Times for user tests

		Times	
User		Graph View	Radial View
1		60	30
2		37	29
3		42	10
4		27	23
5		25	35
6		30	23
7		33	39
8		36	23
9		31	26
10		7	53

Table 2 Preferences of participants, G – Graph view, R- Radial View

User	Preferred View	Easier to use
1	R	R
2	R	R
3	G	G
4	R	R
5	R	R
6	R	R
7	G	R
8	R	R
9	R	R
10	G	G