University of Cape Town

Computer Science

2008 Honours Project Report

Social Networks Analysis in E-Learning Systems at UCT

Ivandro Issufo
(ISSIVA001)

Project Supervisors:

A/Prof Sonia Berman

Dr Anet Potgieter

Abstract

Knowledge management is becoming more and more important to bring organizations competitive advantage. The way that knowledge flows and is shared is also very important in learning environments, the focus of this project's investigation.

The e-learning systems at UCT were investigated, in particular Vula forums. Vula was the system chosen to be investigated due to the quality and amount of data available. A tool, SONET, was developed to investigate the student's participation within the Vula forums by performing social network analysis, correlate this participation to their marks and profile them based on their social network metrics and on their personal details. The project aimed at using the social network analysis to visualize the participants and see if the metrics generated in the social network analysis are a useful indicator of performance or not. The system comprises three semi-independent subsystems. The social network system for extracting a social network from raw course-system data and perform social network analysis. The second is a Bayesian network subsystem that will learn the correlations between a student's profile and his/her performance and infer performances. The third is a Visualization subsystem that will allow users to effectively interact with social network data and to explore the structural properties.

This paper discusses the development of the Social Network subsystem. This subsystem is broken in three phases namely the data loafing, the social network generation and the metrics calculation.

Acknowledgements

I would like to thank my supervisors, Prof. Sonia Berman and Dr. Anet Potgieter, and fellow SONET project partners, Avishkar Bhoopchand and Kaashief Hartley, for the time and wealth of ideas that they contributed to the content of this report. I would also like to thank Tony Carr from the UCT Centre for Educational Technology for the information provided about Vula forums.

Table of Contents

List of Figures

List of Tables

1. Introduction

Knowledge management is becoming more and more important to bring organizations competitive advantage. The way that knowledge flows and is shared is also very important in learning environments, the focus of this project investigation. In order to analyze knowledge sharing we investigated e-learning systems at UCT and observed how the knowledge flows across students. This project purposes, implements and evaluates a system that can, amongst many other things, be used to analyse the effectiveness of communication on course sites, identify knowledge hubs, correlate participation in the communication network with performance, as well as attempting to predict future student's performance. Vula was the system chosen to be investigated due to the quality and amount of data available. A tool, SONET, was developed to investigate the student's participation within the Vula forums by performing social network analysis, correlate this participation to their marks and profile them based on their social network metrics and on their personal details. SONET can improve e-learning systems by showing students that use the forums heavily and the ones that never give their contribution. The site responsible, especially lecturers, can use SONET to improve their students' performances by encouraging the participation of the weakest students or promoting interactions between top and weak students. The system comprises three semi-independent subsystems. The social network system for extracting a social network from raw course-system data and perform social network analysis. The second is a Bayesian network subsystem that will learn the correlations between a student's profile and his/her performance and infer performances. The third is a Visualization subsystem that will allow users to effectively interact with social network data and to explore the structural properties. This paper describes the social network subsystem development including the requirements identification, system design implementation and testing. This paper discusses the development of the Social Network subsystem. The paper is ten different chapters. All background information regarding social network analysis is covered in Chapter 2. Chapter 3 describes SONET completely in terms of its software artifacts. System's requirements and design are presented with the support of UML diagrams. In Chapter 4, we describe the implementation phase where a description of the classes used is given. A description of the data used by the SONET system is also presented in this chapter. Chapter 5 shows how the different subsystem were integrated and in Chapter 6 we present the results obtained and we describe the testing we have done to illustrate the correctness of the social network metrics. The system's evaluation and conclusions are discussed in Chapter 7 and 8 respectively. Future implementations and extensions to SONET are outlined in Chapter 9.

2. Background and Related Work

A social network consists of a group of actors and connections between them [10]. In order to study the social network, a social network analysis has to be performed. Social network analysis is an approach to analyze a social network based on the relationship between people. Wasserman [10] describes social network analysis as a “distinct research perspective within the social and behavioural sciences because social network analysis is based on an assumption of the importance of relationships among interacting units”. Instead of focusing on the attributes of the individuals as is done in standard social analysis, social network analysis focuses on ties, the interactions and relationships between the individuals as a way of characterizing their behaviour. This analysis is characterized by adopting mathematical models especially from graph theory. Graph theory provides both an appropriate representation of social networks as well as a set of concepts that can be used to study formal properties of social networks [12].

2.1Fundamental Concepts in Network Analysis

Wasserman [10] describes some key concepts that are fundamental in the analysis of social networks. These concepts include: actor, relational tie, subgroup, group, relation and network. Below we de discuss some of these key concepts.

Actor- As discussed above, social network analysis investigates the relationships between different social entities, the actors. These actors can be representing unique individuals within a group, departments in an organization, cities in a country, etc.

Relational Tie – Is the link between a pair of actors in a social network viewed from one side of the communication. The relational tie from A to B might differ from the one from B to A. Example of ties within a social network can be:

- Evaluation of one person by another (friendship, respect)
- Behavioural actions (reading a message, sending a message)
- Transfer of material resources (business transaction, lending money)

Subgroup – Is a subset of actors and the ties between them. A subgroup can be formed by emphasizing a specific condition relevant to the social network analysis

Group – Is a finite set of actors who for conceptual, theoretical or empirical reasons are treated as a finite set of individuals on which network metrics are performed.

Relation – is the collection of ties of a specific kind measured on pairs of actors within a group. Different relations can be measured between a pair of actors. For example, for a pair of nations we can measure the formal diplomatic ties as well as the dollar amount of trade for a given year.

Social Network – From the definition provided above, we can define a social network as a finite set or sets of actors and the relation or relations between them.

2.2Social Network Data

2.2.1 Data Type: Structural and Composition Variables

Network data can contain two different types of variables: structural and composition [11]. Structural variables measure ties between pairs of actors like for instance friendships between people or transactions between corporations. They are just defined for a pair of actors. Composition variables are used to measure an actor's attributes. They are defined for each actor and can be gender, age or ethnicity for individuals, or address or expenses for an organization.

2.2.2 Network Population

The definition of actors to include in the Social Network population can be a difficult issue to deal with. Some actors might not be fixed and be always moving or it may just be difficult to determine whether an actor belongs to a specific set of actors. A good example would be the study of elites within a specific community. Drawing the boundary of the network set, with just the elites, may be difficult or even impossible to determine.

In [13], Laumann, Marsden and Prensky discuss two different approaches to specify the boundaries in a social network namely *realist* approach and *nominalist* approach. In the realist approach, the actors are part of an already formed group and they acknowledge themselves as members of the group. An example can be the members of a street-corner gang, which are acknowledged as a social entity by its members. In the *nominalist* approach, the boundaries are drawn based on the researcher requirements. For example if studying the performance of students within a course site forum, the researcher might exclude the lecturers and tutors that are part of the forum activity.

2.3. Mathematical Representation of Social Networks

According to Hanneman [5], social network analysts use two different mathematical techniques to show the social network relations: graphs and matrices.

2.3.1 Graph Theory

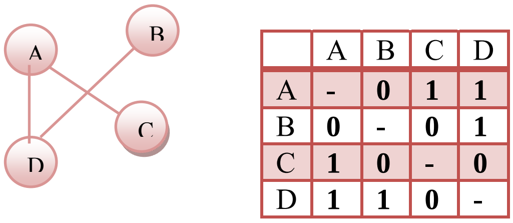
A graph G is a structure consisting of a set of nodes N and a set of links L [13]. In a social network representation, nodes represent people and links represent relationships between people.

Graph theory provides many benefits to the analysis of social networks. First, social network attributes can be mapped onto the vocabulary provided by graph theory. Second, graph theory allows these attributes to be measured by using its mathematical operations. Lastly, by using both the vocabulary and the mathematical operations we can prove graph theorems and consequently social network representations [14]. The main advantages of using graph theory to represent social networks, is the ability to visual representation of the data that a graph offers and the ease of identification of the most important actors in a social network..

In the real world, many relations are directional relations [12]. A relation is said to be directional when the ties are oriented from one actor to another. If we consider friendship preferences among people, we can verify that it is a directional relation. For example person A may choose person B as a friend but the opposite may not be true. The difference between a graph and a directed graph are just the links that have the direction specified. .

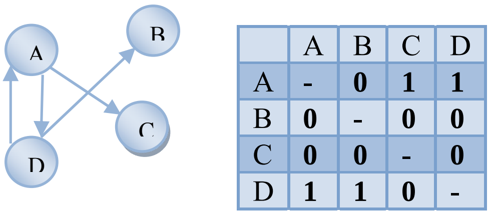
2.3.2 Matrices

Matrices are an alternative mathematical model to represent and study social networks. The more actors' are added to the social network, the harder it gets to visualize it when using a graph according to Hanneman [5]. All the mathematical and computer tools used in graphs can also be used in matrices. The basic matrix used for social network analysis is called an adjacency matrix or sociometrix. The matrix is of size N*N (N rows and N columns) where each matrix entry shows whether two nodes are adjacent or not. A 1 represents a relation and a 0 the contrary. If dealing with a directed graph, the sender is the row and the receiver the column. The sociomatrix for a non directional graph is symmetric.



	A	B	C	D
A	-	**0**	**1**	**1**
B	**0**	-	**0**	**1**
C	**1**	**0**	-	**0**
D	**1**	**1**	**0**	-

Figure 1 – Matrix representation of an undirected graph



	A	B	C	D
A	-	**0**	**1**	**1**
B	**0**	-	**0**	**0**
C	**0**	**0**	-	**0**
D	**1**	**1**	**0**	-

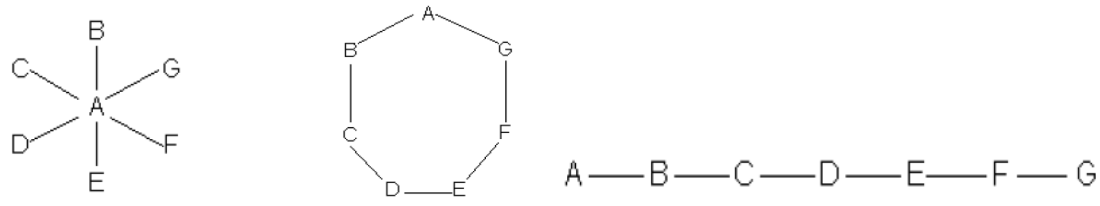
Figure 2 – Matrix representation of a directed graph

2.4 Social Network Analysis

The analysis of a social network can be made at three different levels. At a node level, monadic metrics are analysed, at link level, dyadic metrics are analysed and finally at graph level where entire graph metrics are produced [2].

2.4.1 Centrality and Prestige

Many writers have already proposed definitions of importance of an actor within a social network. Wasserman [10] divides the concept of importance in two parts: centrality and prestige. Centrality looks to the extent to which an actor is involved in the network without examining whether an actor is the source or destination of a tie. A non-directional graph is the more appropriate to investigate an actor's centrality where a central actor would be involved in many ties. Prestige can only be investigated when the source and destination of a tie can be identified. An actor with high prestige would be someone involved in many ties as the receiver. Being involved in many ties, but as the source, does not increase an actor's prestige. The measurement of prestige can only be done by using directed graphs. Both centrality and prestige are first calculated for individual actors but they can later be combined to obtain a group level centralization or prestige. The variations of centrality and prestige in a graph can be seen in the figure 3 and are discussed further below.



A has maximum possible centrality

No actor has higher centrality than anothe

D has higher centrality while A and G have lower centrality

Figure 3 – Three illustrative networks for the study of centrality and prestige

Centrality Metrics

Wasserman [10], proposed three different metrics to measure an actor's centrality: degree centrality, closeness centrality and betweenness centrality.

Degree Centrality

Degree centrality looks at the number of ties that an actor has to other actors in the graph G. From figure 3, we can see that in the star graph, A is more central that any other node while in the circle graph that all have the same centrality. In the star graph, A would have a degree centrality of 6 while the other nodes would have a degree centrality of 1. In the circle graph, all actors have the same degree centrality. The degree values vary from 0 to G-1. In order to standardize the measure and allow different networks to be compared, a normalized degree was proposed. The normalized degree ranges from 0 to 1. If we consider T the number of ties of an actor and G the graph size (number of nodes), the normalized degree centrality for a node ni is given by:

$$\mathrm{D} = \frac{\mathrm{T(ni)}}{\mathrm{G} - 1}$$

Once the individual degree is calculated, the group degree can also be analyzed. The group degree is given by:

$$\mathrm{Dg} = \frac{\sum_{\mathrm{ni} \in \mathrm{N}}[\mathrm{degree(nx)} - \mathrm{degree(ni)}]}{(\mathrm{G} - 1)(\mathrm{G} - 2)}$$

where nx has the maximum degree value in N

The group degree can be used to see how centralized the actor's degree is within a specific network. The metric ranges from 1, when an actor is connected to all G-1 actors and the others only interact with this one like in the star graph, to 0 when all degree are the same like in the circle graph.

Closeness Centrality

The second aspect of Wasserman's centrality is focused on closeness or distance. This metric sees how close an actor is to all the other actors in the network. An actor is considered central if it can interact with the all the other with the least steps. Closeness is based on the idea that centrality is inversely related to distance meaning that as an actor moves further apart from other actors, its centrality decreases. From the star graph in figure 3 we can see that A has shortest possible paths to all the other network actors therefore has maximum closeness. The closeness centrality is given by:

$$\mathrm{C} = \frac{1}{\sum_{\mathrm{nj} \in \mathrm{N}, \mathrm{nj} \neq \mathrm{ni}} \mathrm{dist(ni, nj)}}$$

The closeness value for a node ranges from 0 (far away) to 1/(G-1) (very central).
Closeness centrality can also be calculated at a group level. It shows on average how close all the nodes are to each other. The group closeness centrality is given by:

$$\mathrm{Cg} = \frac{\sum_{\mathrm{ni} \in \mathrm{N}}[\mathrm{cloeseness(nx)} - \mathrm{closeness(ni)}]}{2(\mathrm{G} - 1)(\mathrm{G} - 2)/(\mathrm{G} - 3)}$$

where nx has the minimum closeness value in N

As with the group degree centrality, this metric reaches its maximum when one actor interacts with all the other actors, with a minimum possible shortest path (length 1), while the other actors take 2 steps to interact with the remaining G-2 actors what occurs in the star graph. The minimum is 0 when all the actors have the same closeness value like in a fully connected graph or in circle graph.

Betweenness Centrality

If two non adjacent actors want to interact with each other, they will have to rely on the actors that lie in the paths between them. These actors placed in between a pair of actors have some control over the interactions between the two non adjacent actors. From the star graph above, we can see that actor A lies on all paths linking the other six actors. The line graph in figure 3 also shows that the actors in the middle are determinant for the communication across the network. The more an actor lies in the shortest path between a pair of nodes, the higher is its “betweenness” centrality and the more central the actor is. The “betweenness” centrality of a node is given by the sum of all shortest paths between all nodes that contain the node as a percentage of all shortest paths between all nodes. It is given by:

$$\mathrm{B} = \sum_{\mathrm{nj} \in \mathrm{N}} \sum_{\mathrm{nk} \in \mathrm{N}, \mathrm{nk} \neq \mathrm{nj}} \frac{\#(\mathrm{P}(\mathrm{nj}, \mathrm{nk}, \mathrm{ni}))}{\#(\mathrm{P}(\mathrm{nj}, \mathrm{nk}))}$$

where

$\mathrm{P}(\mathrm{nj}, \mathrm{nk})$ is the set of all shortest paths from nj to nk

$\mathrm{P}(\mathrm{nj}, \mathrm{nk}, \mathrm{ni})$ is the set of all shortest paths from nj to nk that passes through node ni

It has a minimum of 0 if a node falls on no shortest path and a maximum of (G-1)*(G-2)/2 which is the number of pairs of actors not including ni.
The group betweenness centrality allows the comparison of different networks with regard to the heterogeneity of the betweenness centrality across the network. It is given by:

$$\mathrm{Bg} = \frac{\sum_{\mathrm{ni} \in \mathrm{N}} [\mathrm{betweenness}(\mathrm{nx}) - \mathrm{betweenness}(\mathrm{ni})]}{(\mathrm{G} - 1)}$$

where nx has the maximum betweenness value in N

It also ranges from 0 in a circle graph (equal betweenness for all nodes) to1 in a star graph (completely unequal).

Prestige Metrics

In a non directional graph, the degree of a node measures the number of adjacent nodes to it. When dealing with a directed graph, a node can be adjacent to or adjacent from another node according to the direction of the relation [11].

Indegree

The indegree of a node dI(ni) is the number of nodes adjacent to ni, number of arcs ending at ni. Indegree measure how popular and accessible an actor is within a network. In a friendship environment, an actor with a high indegree is someone whom many others consider a friend.

Outdegree

The outdegree of a node, dO(ni), is the number of nodes adjacent from ni, number of arcs starting at node ni. The outdegree measure how expansive an actor is. Following the friendship example, an actor with a large outdegree is one who appoints many others as friends.

2.4.2 Web Link Metrics

The links between websites also form networks therefore tools that have been used to analyze the Web structure can also be used in social network analysis [2]. The majority of these techniques have been developed by search engines to rank pages across the Web. Web analysis requires the use of directional relations where the actors with a high indegree are called authorities and the ones with a high outdegree are called hubs. An authority is a site that is trusted by others on a specific topic while a hub is a site that can show the most important authorities. In [15], two algorithms are proposed to get a network’s authorities, HITS and PageRank (used by Google). They are both recursive algorithms as they consider the importance of the sites being linked.

HITS

The HITS algorithm was developed by J. Kleinberg [17]. The base of the algorithm is that a web page serves two purposes: to provide information on a topic, and to provide links to other pages giving information on a topic. Therefore a website can be an authority if it provides good information about the subject or a hub if it provides links to good authorities on the subject. The HITS algorithm is an iterative algorithm developed to quantify each page's value as an authority and as a hub. Hubs and authorities exhibit what could be called a mutually reinforcing relationship: a good hub is a page that points to many good authorities; a good authority is a page that is pointed to by many good hubs.

$$\vec{a}^* = \frac{1}{\lambda^*} A^T A \vec{a}^*, \qquad \vec{h}^* = \frac{1}{\lambda^*} A A^T \vec{h}^*.$$

PageRank

The PageRank method was developed by Brin and Page [18]. It is defined recursively and depends on the number and PageRank metric of all pages that link to it ("incoming links"). A page that is linked to by many pages with high PageRank receives a high rank itself. If there are no links to a web page there is no support for that page. PageRank is a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. A probability is expressed as a numeric value between 0 and 1. A 0.5 probability is commonly expressed as a "50% chance" of something happening. Hence, a PageRank of 0.5 means there is a 50% chance that a person clicking on a random link will be directed to that document. PageRank is given by:

$$\mathrm{PageRank\ (ni)} = \sum_{\mathrm{nj} \in \mathrm{N}} \sum_{\mathrm{ui,uj} \in \mathrm{U}} \frac{\mathrm{PageRank(ni)}}{\#(\{\mathrm{uj,k: uj,k} \in \mathrm{U}\})}$$

where Un is the set of bidirected links of the social network.

2.4.3 Other Monadic Metrics

Eccentricity

Eccentricity is a graph distance metric discussed by Wassermans [10] where we calculate for each node the length of the shortest path to the node furthest away. It is given by:

$$E = \max\left(\{dist(ni, nj) : nj \in N\}\right)$$

It summarizes how long it will take for a node to reach the node furthest away from it in the graph. It ranges from 1 (if a node is adjacent to all other nodes) to G-1.

BaryCenter

This algorithm is discussed in [19], and it just gives the total shortest paths of a node to all other connected nodes. It can be seen as an inverse of the closeness defined above. More central nodes in a connected component will have smaller overall shortest paths, and 'peripheral' nodes on the network will have larger overall shortest paths. It is given by:

$$\mathrm{BC} = \sum_{\mathrm{nj} \in \mathrm{N}, \mathrm{nj} \neq \mathrm{ni}} \mathrm{dist(ni, nj)}$$

Degree Distribution

Degree distribution is just an extension of the degree centrality defined above. It is the probability distribution of these degrees for the whole graph. The degree distribution P (k) of a network is then defined to be the fraction of nodes in the network with degree k. Thus if there are n nodes in total in a network and n_k of them have degree k, we have $P(k) = n_k/n$. If each node has a probability p, the degree distribution is given by:

$$P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k},$$

2.4.4 Other Graph Metrics

Density

Density looks at the number and proportion of links in the graph as a whole. The maximum unordered pairs of nodes that a graph (non directional) can have are G*(G-1)/2, therefore the maximum number of links are G*(G-1)/2. By calculating density, we can see what proportion of these links actually exist. It is given by the number of links existent in the graph, L, to the maximum number possible:

$$\mathrm{Density} = \frac{\sum_{\mathrm{nj} \in \mathrm{N}} \mathrm{degree(nj)}}{\mathrm{G(G-1)}}$$

It ranges from 0, if the graph has no links, to 1 if the graph is fully connected or complete.

Diameter

The diameter of a graph measures the length of the longest shortest path in the graph. It can be seen as the maximum value for nodal eccentricity. It ranges from 1 in a complete graph to G-1.

$$Diameter = \max\left(\{eccentricity(ni) : ni \in N\}\right)$$

2.4.5 Dyadic Metrics

The link strength between two actors is another important aspect of the social network analyses that can be measured at different levels. According to Hanneman [5], five different approaches can be used, each one giving a different level of detail. The first approach is the binary measure that is the most common and just defines whether a relation exists (coded 1) or not (coded 0). The second approach is the multiple-category nominal measure of relations. It has different categories and assigns each of the relations to a category. This measure applies a multiple choice concept. For example the links within a company can be defined as friendship, informal or formal. The two approaches described are nominal scales of measurement. Hanneman also discusses ordinal scales of measurement. The first is the grouped ordinal measures of relations where the link strength value is a grouped ordinal scale. The link value would be increased or decreased according to certain conditions like frequency or intensity. For instance, actors that communicate daily could have a stronger link strength compared to those that only communicate monthly (frequency). The other ordinal approach is the full-rank ordinal measure of relations. This method scores all the links of an actor in a rank order from strongest to weakest. It would use a scale to give to each relation a different level of intensity, not requiring these differences to be constant, meaning that the difference from choice one and two might not be the same as the difference between choice two and three. Lastly there is the interval measure of relations that is the most advanced approach of measurement allowing link strength to be defined according to another link, for example, “this tie is twice as strong as that tie”. Here the difference between a rank one and two is the same as the difference between rank eight and nine. The ordinal measures provide much more detail than the nominal as the link values reflect much more accurately what is happening in reality. Even having such a level of detail, benefiting from it is a challenge as most of the social network analyses algorithms use binary data. Therefore, even having ordinal data, it often has to be converted to some binary format [5].

3. Software Engineering

Initially, the requirements for the SONET system are defined and then visually represented through use case diagrams (UCDs), sequence diagrams (SDs), and finally, activity diagrams (ADs). Classes are identified using a high-level class diagram (CD), and their relationships outlined.

3.1 Data Gathering

One of the main factors that would determine the success or failure of the SONET system was the data used. The system required appropriate data in terms of quality and quantity. The quality of the data would reflect how easy it would be to draw relationships among network actors. The quantity, in contrast, would determine the number of people that would be investigated and the period under investigation. Therefore, we required a dataset that allowed the creation of a network with more than 200 nodes and for a period of at least a year to strengthen the social network metrics as well as the predictions made by the Bayesian network. In order to know what data sources were available, we interviewed Tony Carr from the Centre for Educational Technology at UCT. Tony Carr suggested and introduced us to two systems that were further analysed and evaluated namely VULA and DFAQ. DFAQ, Dynamic Frequently Asked Questions, was a site used to support a number of courses at UCT. It allowed students to ask questions online or by mobile text messaging which other students, lecturers or tutors could answer. DFAQ did not meet the requirements because first we could see who posted messages but not who read them and second because the data provided was for a small period. In contrast, VULA and more specifically the PSY1001W 2007 forum logs, met all the system requirements. The fact that the PSY1001W site was used by more than 400 students, with records of who sent and who read each message and for a one-year period contributed to the choice of this data source. A detailed description of the data provided by VULA is given in the next chapter.

3.2 Requirements Specification

Raw Data Storage and Cleaning
Vula and PeopleSoft provided the data used in the system. An appropriate subset of the data must be identified that is suitable to perform the social network analysis as well as the correlations and inferences by the Bayesian network. These data must be converted into csv format and stored in the database. The information that is not relevant for the final system must be removed from the database in order to save disk space.

Create Sender-Receiver Relations
The construction of the social network using forum messages is different from for example constructing a social network using e-mails. When using e-mails, we know who the sender and the receiver are but when using forums, there is a sender but there is no receiver until someone reads the message. For this reason, we only have a relation between two students when one posts a forum message and another one reads or replies to the message. A sender-receiver relation must be made for every forum message.

Confidentiality
Student records and marks are part of the information required by the system to profile and investigate each student's performance. For ethical reasons, students' identity needs to be kept private in the final system.

Social Network Creation
The system must allow the creation of different social networks depending on the part of the data being investigated. These social network links need to be stored in the database in a

format that can be used by both the visualization subsystem and the metrics calculations section.

Calculate Social Network Monadic Metrics
Centrality and web link metrics must be calculated and stored. These metrics need to be stored in a format that can be used by both the Bayesian network subsystem and a query interface. The more metrics calculated, the better are the predictions and the comparisons made by the Bayesian network.

Calculate Social Network Dyadic Metrics
The links between pairs of actors in the network are not the same. These links must be assigned a weight that allows the user to see whether a relation is stronger than another. These weights are also used by the visualisation subsystem to enhance the most important links.

Calculate Social Network Graph Metrics
For each different social network analysed, the overall social network behaviour must be calculated and stored to allow the comparison of different social networks.

Calculate Forum Metrics
Knowing if a student is interacting more with lecturers rather than students or if he/she is reading more than posting messages can be very important to understand a student's behaviour. Forum metrics must show the number of messages posted and read by each student to a specific person type, whether it is a lecturer, a tutor, a staff member or just another student.

View Student's Details
The end user must be able to view for each student, the personal information, the marks, the monadic metrics and the forum metrics. Charts must also be generated with the given information to aid in the interpretation of the results.

3.3 Design

3.3.1 System Overview

The SONET system was proposed as a way to analyse the effectiveness of communication on course sites by correlating the participation in a communication network with performance as well as attempting to predict future student's performance. The idea was to use social network analysis to get the importance of an actor within a social network and a Bayesian Network to predict future student's performance. The proposed system design overview is show in figure 4. The system was divided in three sections namely social network, Bayesian network and visualization.

Social Network
Ivandro Issufo was responsible for the social network creation and metrics calculation. The section is split into three packages, data loading, social network and metrics. First the data is cleaned and stored in the database. Second, using the cleaned data, a social network is generated and stored in the database. Lastly, social network metrics are calculated for the network previously generated and stored in the database as well. As can be seen from figure 4, the communication between the social network section and the Bayesian network and visualization is done through the database. This section is explained in more detail throughout the paper.

Bayesian Network

The Bayesian network subsystem comprises of learning and inference aspects. Avishkar Bhoopchand was responsible for the learning that involves generating the Conditional Probability Table (CPT) from given past data. This step determined the probabilistic relationship between a student's profile and their marks (or other important factors requiring prediction). Initially, the conditional probabilities for a naïve Bayesian Classifier model are learnt. The Bayesian network can then be extended to learn the actual network structure that best fits the given data. Kaashief Hartley was responsible for the inference engine. It can be queried by the visualization subsystem to predict marks given a student's profile (social network metrics). The data used by the Bayesian network subsystem, the metrics, are read from the database were they are stored once calculated by the social network subsystem.

Visualization

Kaashief Hartley was also responsible for the SONET's visualization. The first aspect is to visualize a social network. The social network is read by the visualization subsystem from the database when the network is to be visualized. Profiles of users within the network are available that map a user's position in the network as well as other profile data to their marks. SONET can also be queried, to predict marks based on students' social network metrics and other known variables.

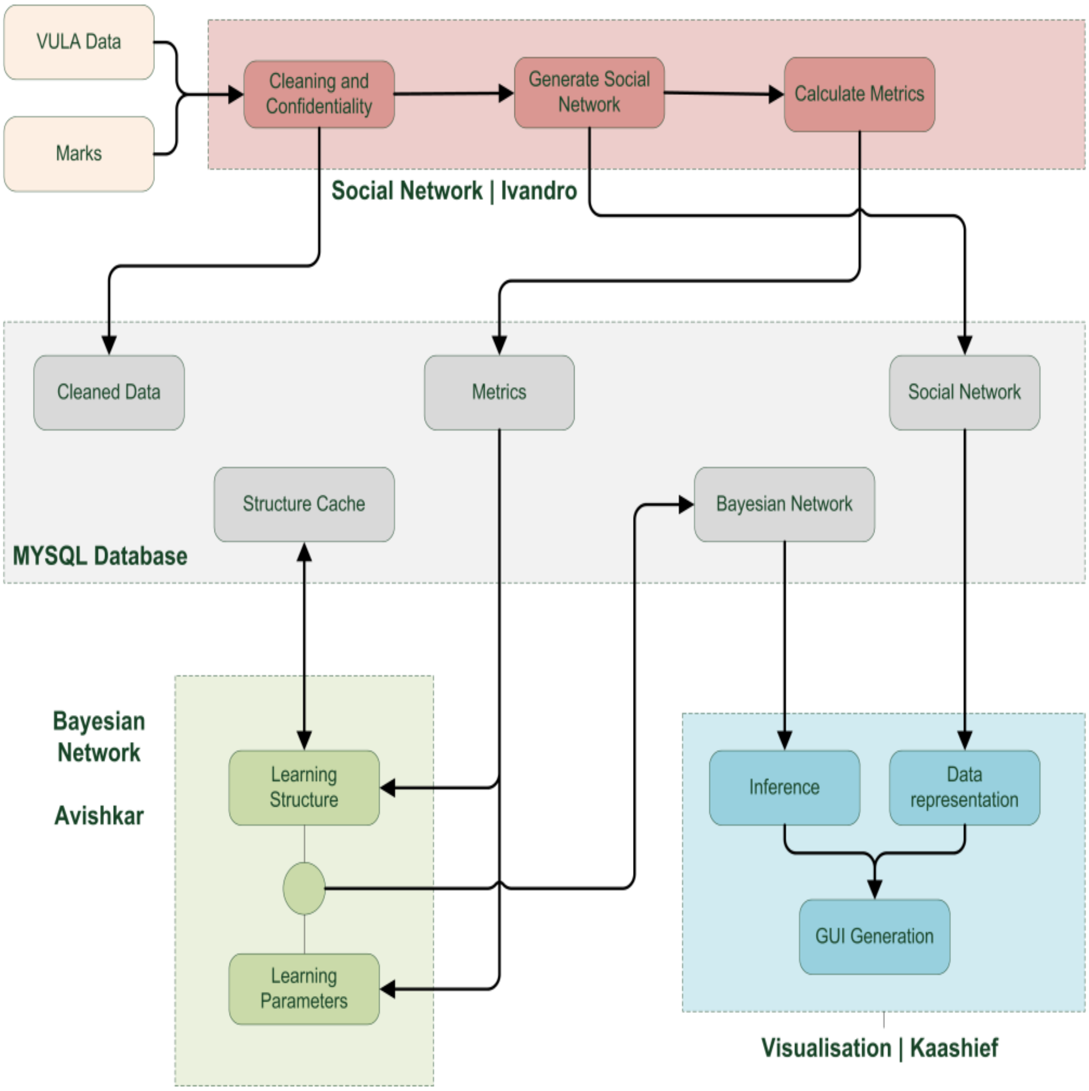


Figure 4 – SONET system description

3.3.2 Social Network Subsystem Overview

The social network subsystem comprises three different packages: the data loading, the social network generation and the social network metrics calculation.
The data loading package consists of three sub packages namely:

- Structure Data
- Load Reads
- Load Posts

The Social Network package consists of two sub packages namely:

- Undirected Graph
- Directed Graph

The Metrics package consists of four sub packages namely:

- Monadic Metrics
- Dyadic Metrics
- Graph Metrics
- Forum Metrics

Figure 5 shows the interaction between the different packages. The Data Loading is responsible to read all the required data into a database, convert it into a structured format and create the relationships for both the reads and the posts. The Social Network package then can create a directed or undirected graph from the data loading relationships. These graphs are then used by the Metrics package to calculate the most relevant metrics.

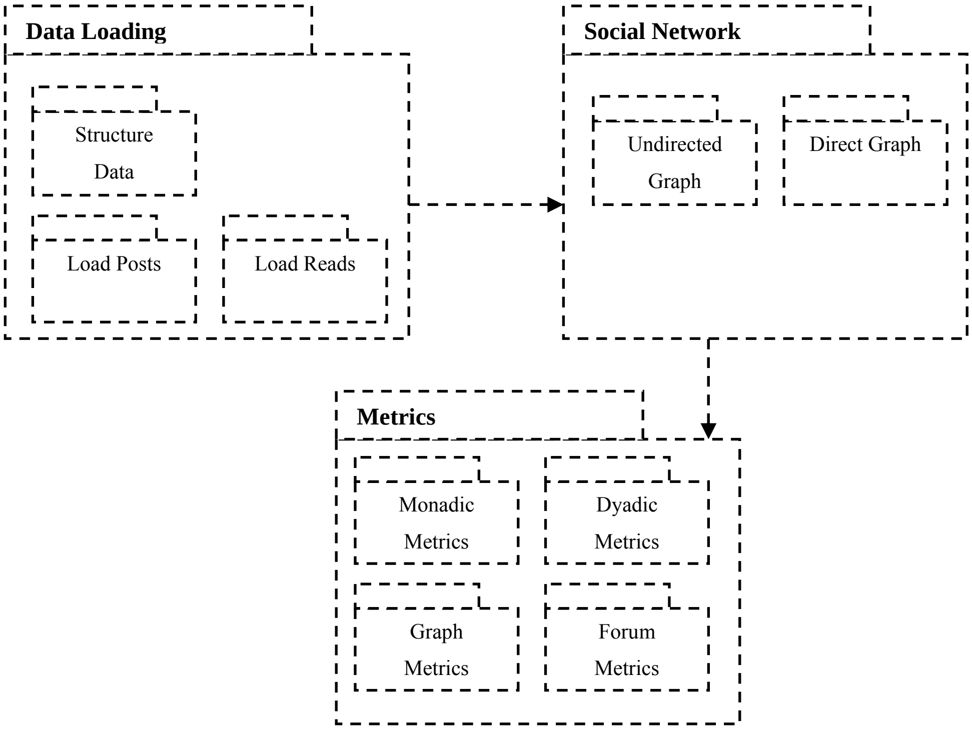


Figure 5 – Package diagram and interactions overview

The user requirements, specified in the previous section, are visually represented by use-case diagrams (UCDs), sequence diagrams (SDs), activity diagrams (ADs) and class diagrams (CDs). These software artefacts are described next.

3.3.3 Use Case Diagrams

Use case diagrams were developed to specify the functionality that the subsystem will offer from the users' perspective. They were used to document the scope of the subsystem and the developer's understanding of what it is that the users require. The different packages contained within the subsystem, namely the Data Loading, the Social Network and the Metrics packages are addressed individually. The Data Loading package UCDs are shown in figure 6. Figure 7 shows how the Social Network can be generated. The UCDs for the calculation of the different metrics are shown in figure 8.

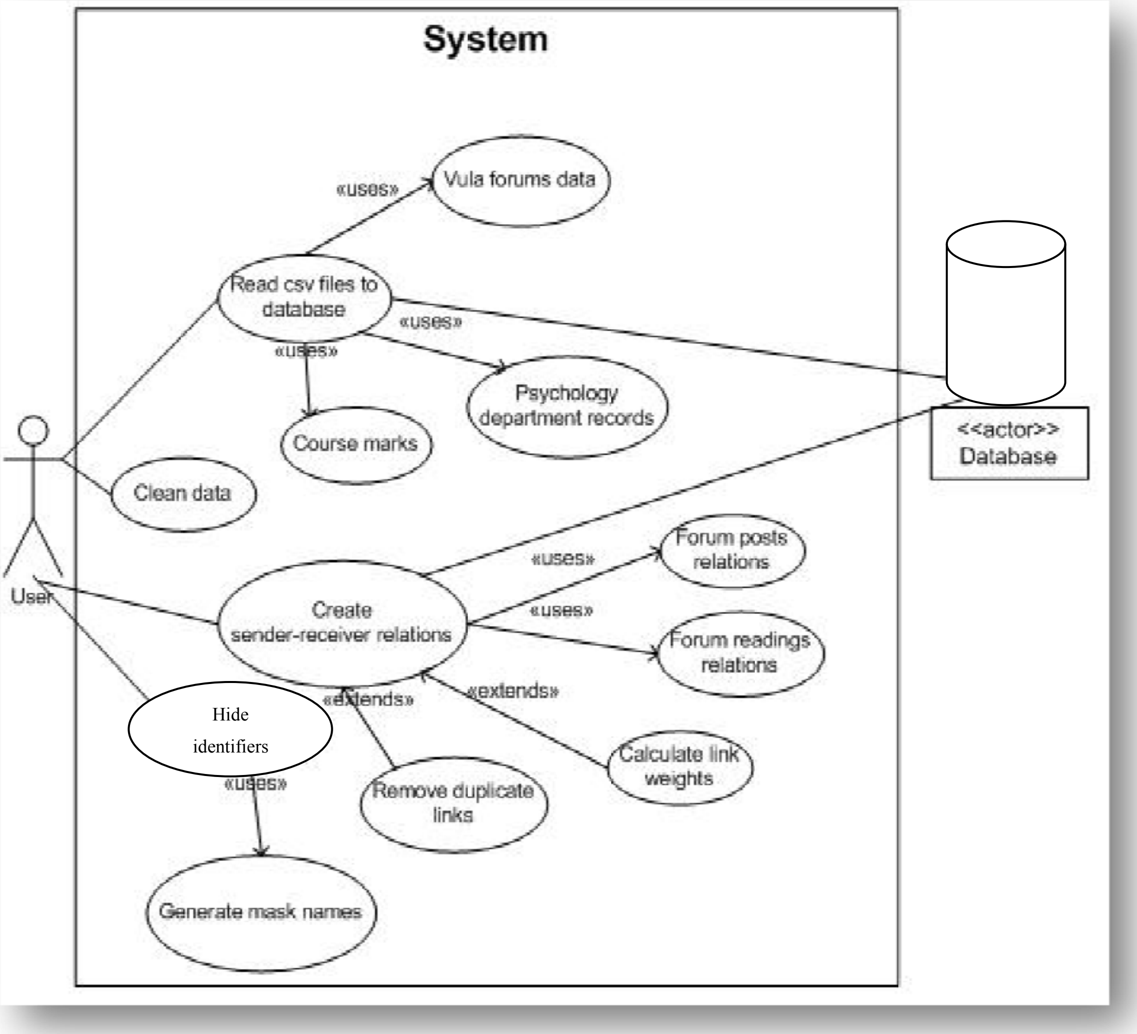


Figure 6 – Use case diagrams for the data loading package

From the diagram it is possible to see that the Data Loading subsystem has four main use cases. The first one is the process of reading the csv files containing the data into a database. These files come from different sources and they include Vula forum posts, the course marks and the PeopleSoft department records. The data is then cleaned to remove the irrelevant information. The sender-receiver relations are created for both the forum posts and forum reads. When creating the relations, the link repetitions can be counted and the duplicates removed. Mask names are also generated to protect the student's identity.

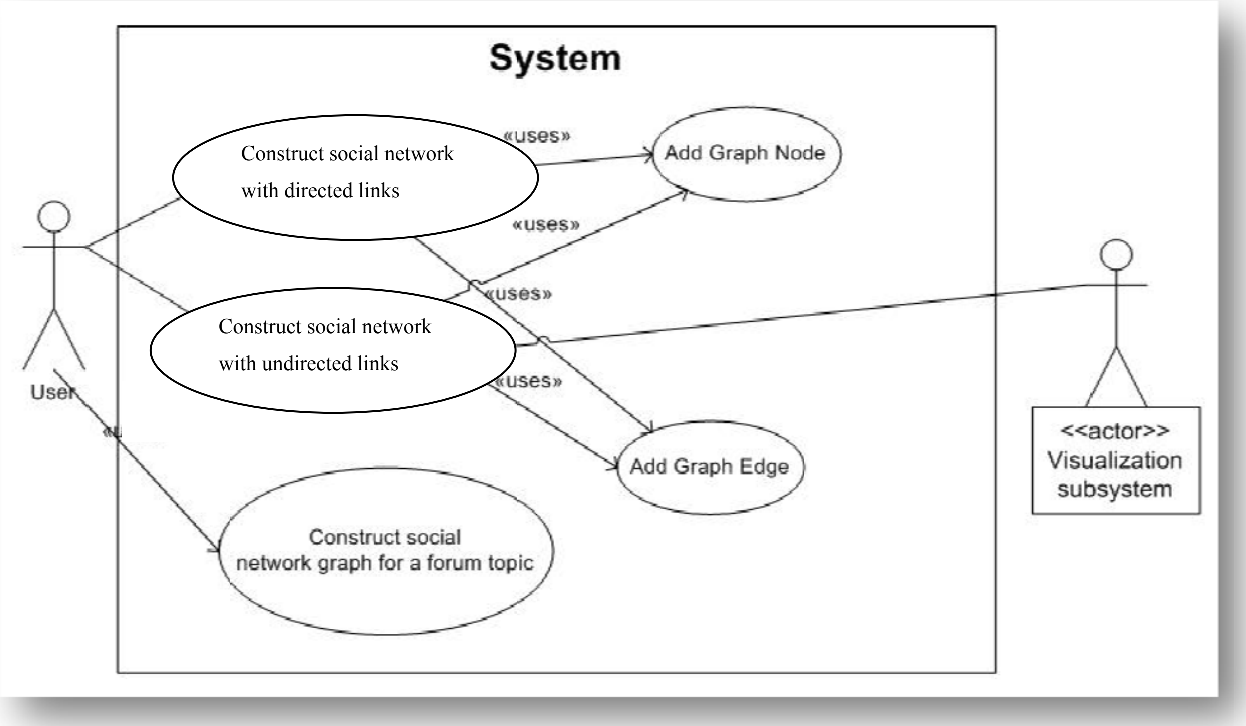

System
Construct social network with directed links
Construct social network with undirected links
«uses»
«uses»
«uses»
«uses»
Add Graph Node
Add Graph Edge
Construct social network graph for a forum topic
User
<<actor>> Visualization subsystem

Figure 7 – Use case diagrams for the Social Network package

The social network graph can use directional or non directional relations. Both graphs are used to calculate the network metrics but only the undirected graph is used by the visualization subsystem. Social network graphs for specific topics within a forum are also generated to allow the comparison of the different social networks at different times of the year.

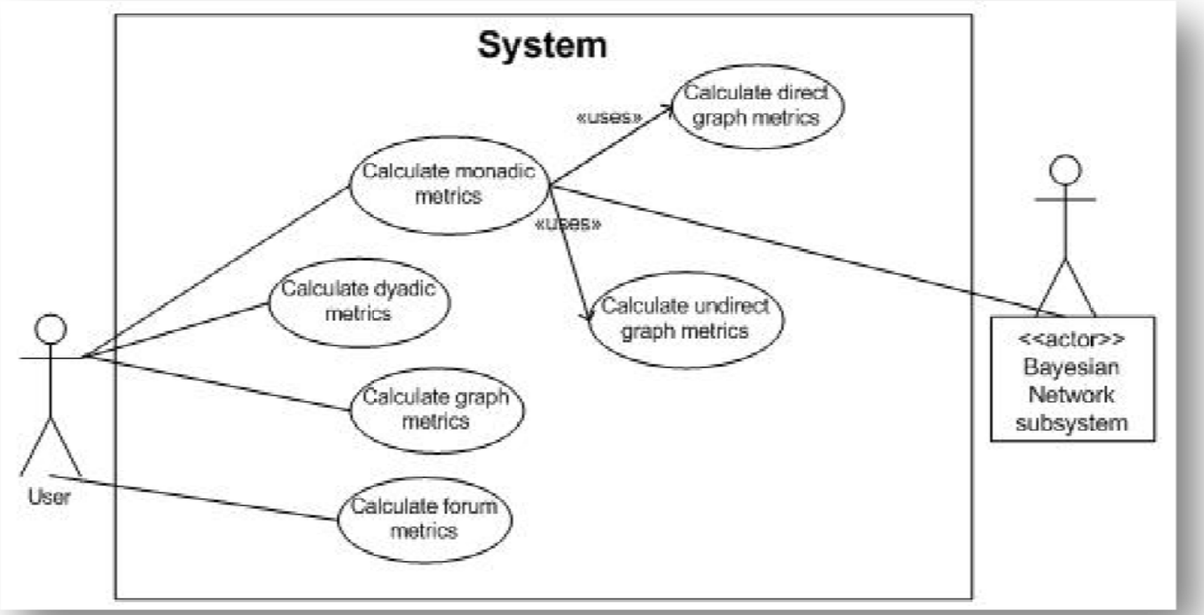

System
Calculate monadic metrics
«uses»
«uses»
Calculate direct graph metrics
Calculate undirect graph metrics
Calculate dyadic metrics
Calculate graph metrics
Calculate forum metrics
User
<<actor>> Bayesian Network subsystem

Figure 8 - Use case diagrams for the Social Network Metrics package

The monadic metrics are calculated for both graph orientations and are the input for the Bayesian network. The graph metrics values are dependent on the values of the monadic metrics calculation.

3.3.4 Sequence Diagrams

The sequence diagrams were created to support use cases by specifying the behaviour of each use case. They show the communication and collaboration between system objects. The behaviour of the Metrics subsystem can be seen from the diagrams describing the Calculation of Monadic Metrics, figure 9, and Calculation of Forum Metrics, figure 10.

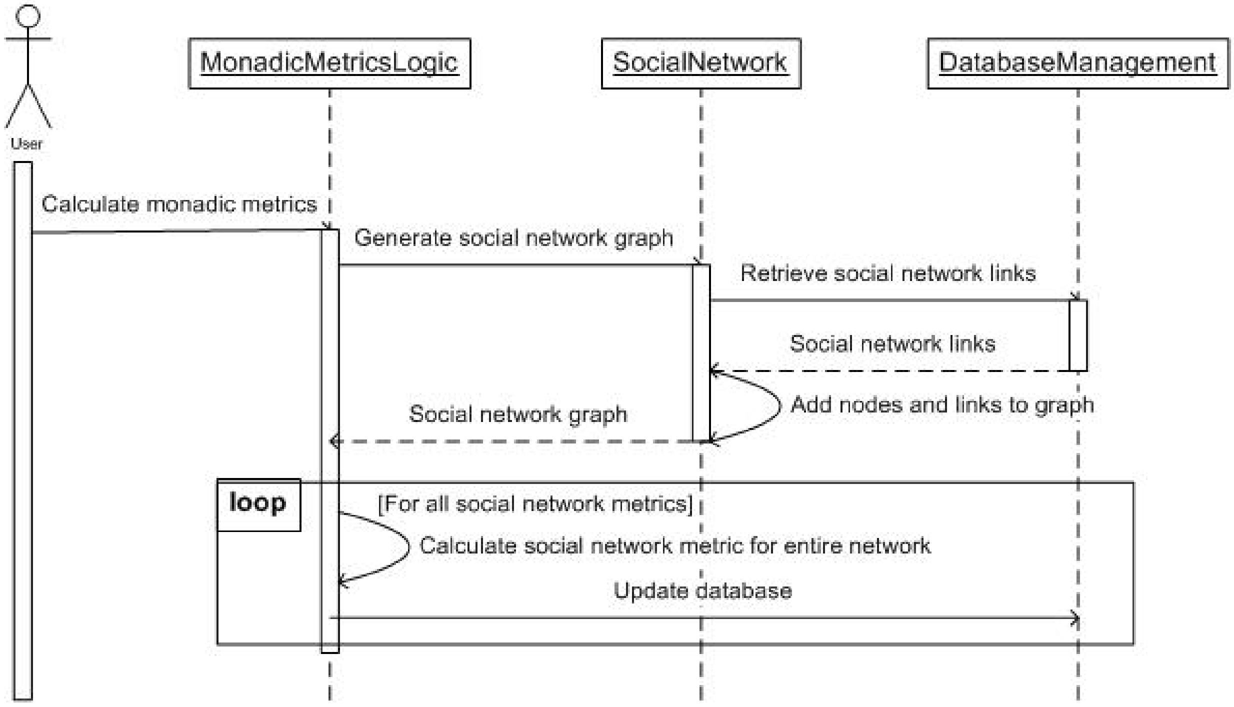

User
MonadicMetricsLogic
SocialNetwork
DatabaseManagement
Calculate monadic metrics
Generate social network graph
Retrieve social network links
Social network links
Add nodes and links to graph
Social network graph
loop
[For all social network metrics]
Calculate social network metric for entire network
Update database

Figure 9 – Sequence diagram for the interaction *Calculate Monadic Metrics*

To perform the calculation of the monadic metrics, the MonadicMetricsLogic class is used. This class contains the methods to run each of the metric algorithms. When the MonadicMetricsLogic class is called to calculate the metrics, it calls the SocialNetwork class to generate the network graph. The SocialNetowrk class reads the links from the database and generates the social network graph requested by the MonadicMetricsLogic. This graph can be directed or undirected depending on the request. When the graph is created, it is sent to the MonadicMetricsLogic that will start running the metrics algorithms on it. A metric is calculated to every graph node and then the values are stored to the database. This process is repeated for each metric algorithm.

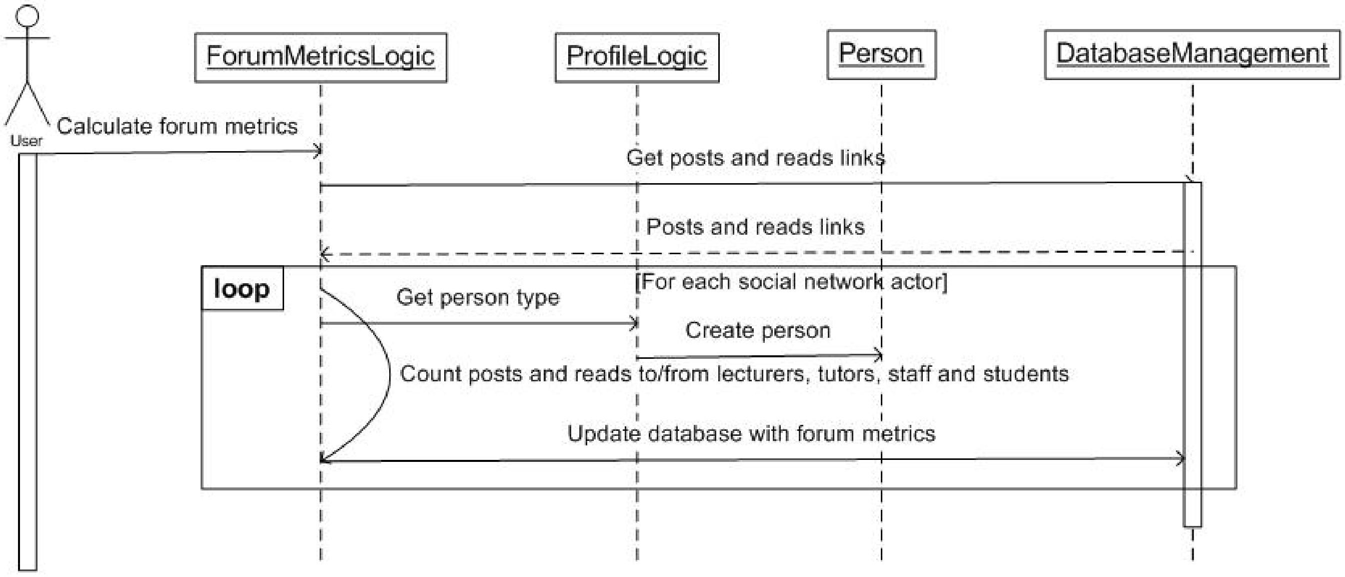

ForumMetricsLogic
ProfileLogic
Person
DatabaseManagement
User
Calculate forum metrics
Get posts and reads links
Posts and reads links
loop
[For each social network actor]
Get person type
Create person
Count posts and reads to/from lecturers, tutors, staff and students
Update database with forum metrics

Figure 10 – Sequence diagram for the interaction *Calculate Forum Metrics*

The algorithms to calculate the forum metrics are all in the ForumMetricsLogic class. For this reason, when the Calculate Forum Metrics use case is executed, a ForumMetricsLogic object is created. This object reads the social network links from the database containing both the posts and the reads. The metrics are calculated and stored in the database for each actor within the network. The calculation process involves the creation of a ProfileLogic object that will return the person type in the link. A person can be a student, tutor, lecturer or administrative staff.

3.3.5 Activity Diagrams

In order to model the detail of how some particular operations were carried out, activity diagrams were developed. The use cases to create the sender-receiver relationships, figure 11, and to create the social network, figure 12, are described in detail in activity diagrams.

Activity Diagram 1
To create a sender-receiver relation, first the forum tables of both posts and reads must be read into memory. Then the posts and the reads are analysed in parallel as they are stored in different tables. The processing is the same for both tables with exception to the operation "Get ID of Message Read" in the reads where for each reading record the ID of the message read is deduced from a long expression. When both tables have a sender and a receiver for every link, they are combined in a third table and the duplicates removed.

Activity Diagram 2
The process of creating a social network graph, as can be seen from the diagram, has three main loops. The first loop is to get the number of actors in the social network. This value is obtained by analysing every social network link. Once we have the value for the number of actors, a graph is created. The second and third loops add the vertexes and edges to the graph. This process is the same for both the undirected and directed graph generation.

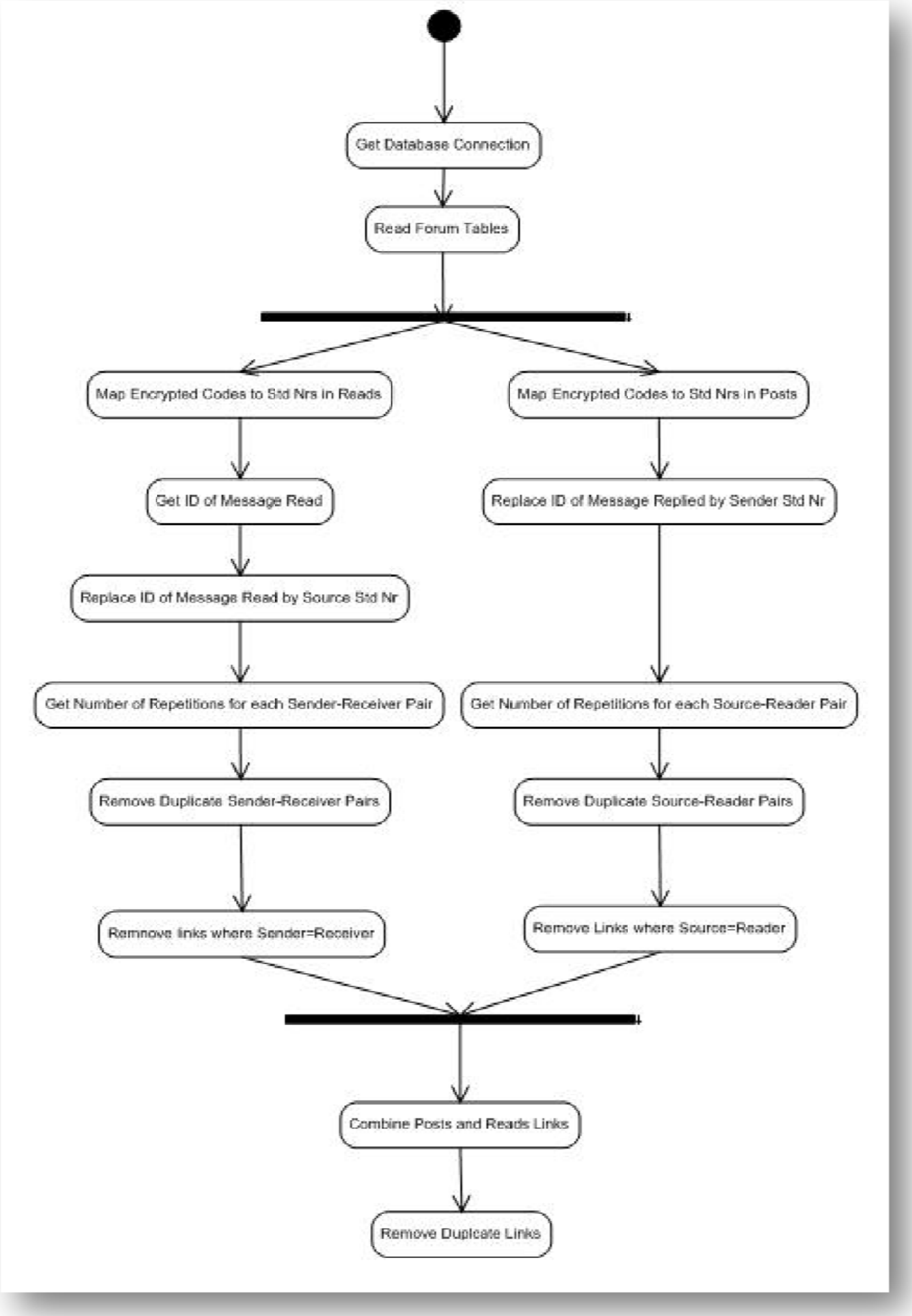

Get Database Connection
Read Forum Tables
Map Encrypted Codes to Std Nrs in Reads
Map Encrypted Codes to Std Nrs in Posts
Get ID of Message Read
Replace ID of Message Replied by Sender Std Nr
Replace ID of Message Read by Source Std Nr
Get Number of Repetitions for each Sender-Receiver Pair
Get Number of Repetitions for each Source-Reader Pair
Remove Duplicate Sender-Receiver Pairs
Remove Duplicate Source-Reader Pairs
Remnove links where Sender=Receiver
Remove Links where Source=Reader
Combine Posts and Reads Links
Remove Duplicate Links

Figure 11 – Activity diagram for the use case *Create sender-receiver relationships*

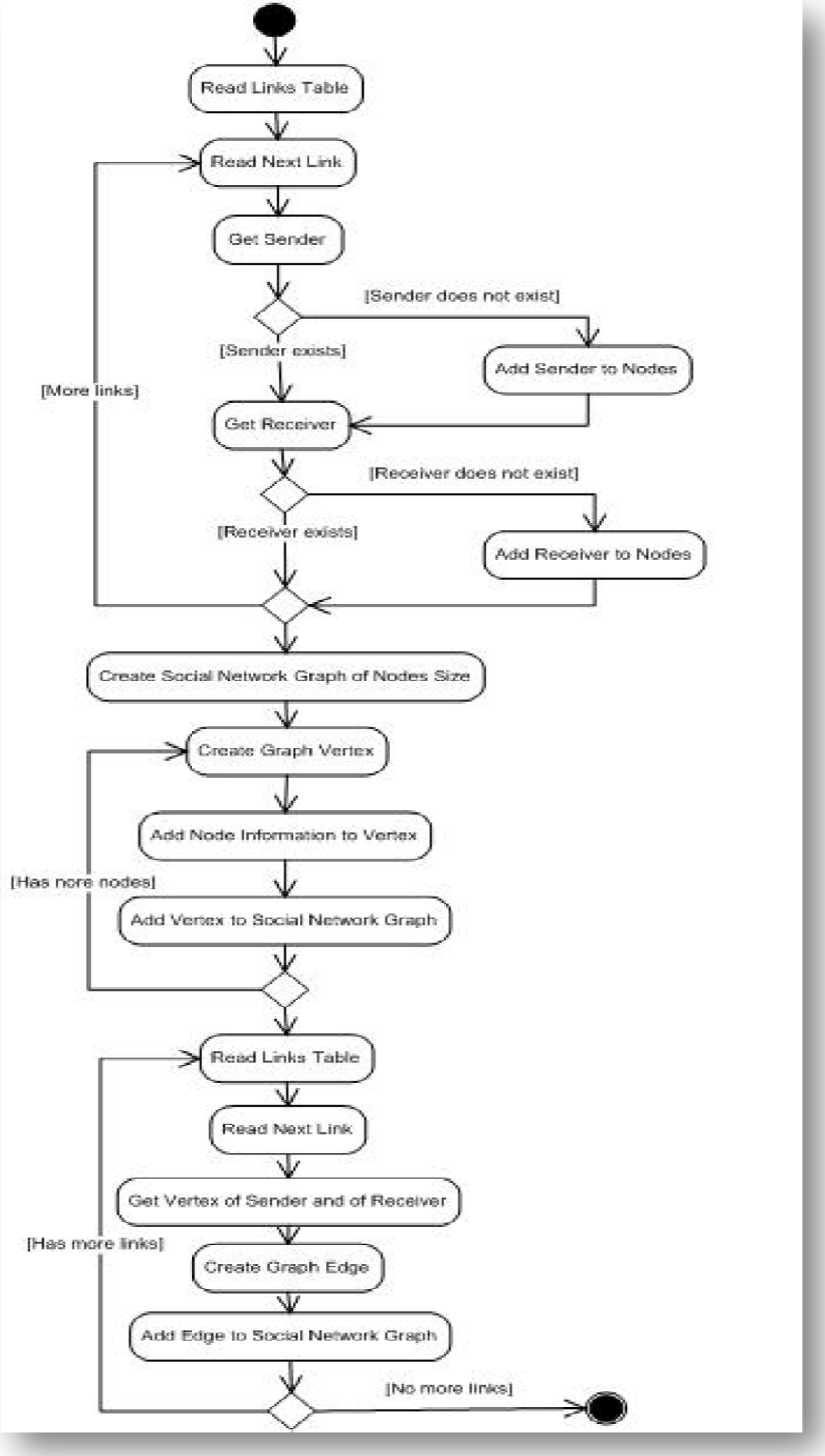


Figure 12 – Activity diagram for the use case *Generate Social Network Graph*

3.3.6 Class Diagrams

Class diagrams were developed to depict the classes within the subsystem's implementation. The implementation classes were identified using the UCDs and SDs presented in sections 3.3.3 and 3.3.4, respectively. The subsystem high level class diagram is provided in figure 13.

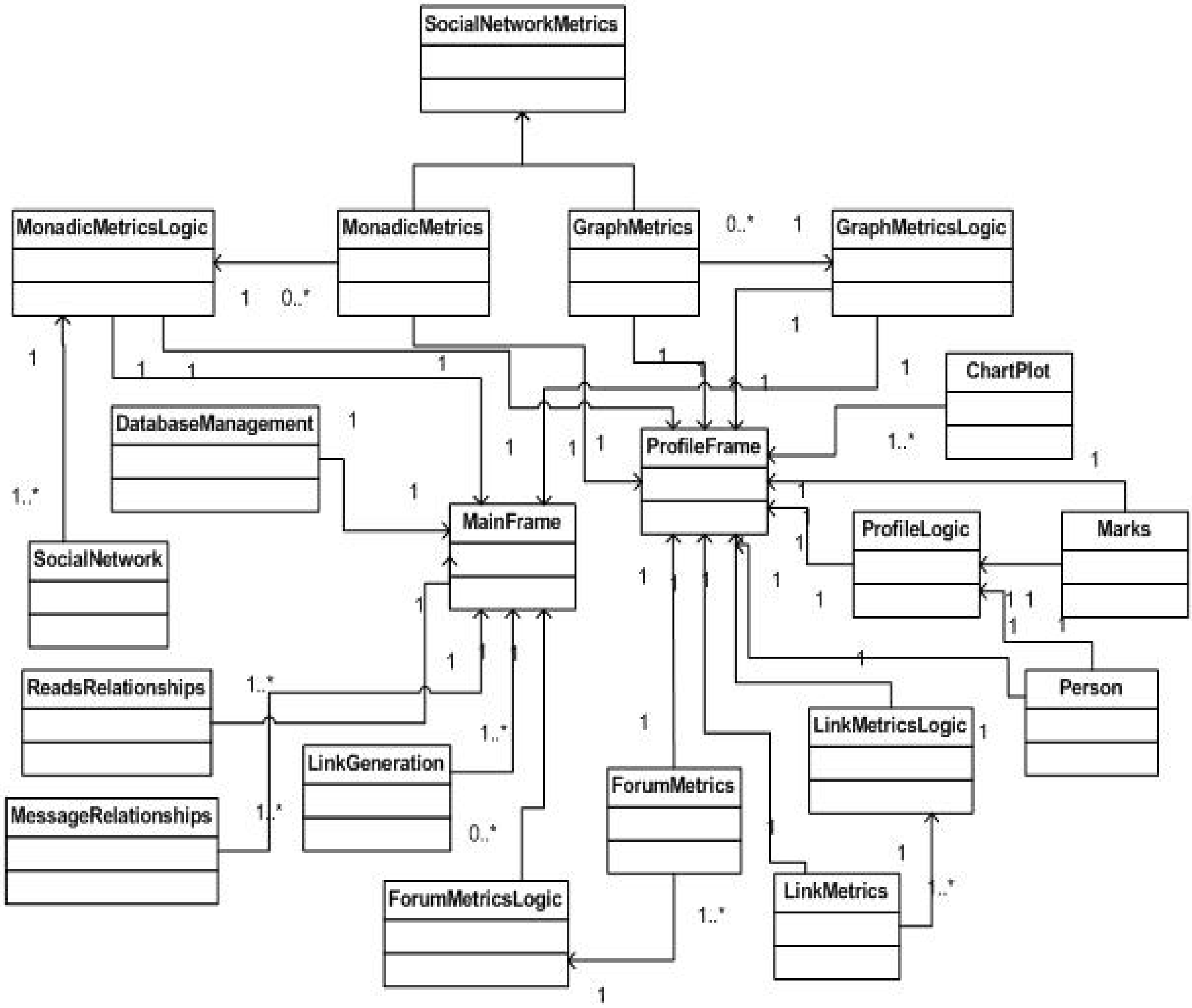


Figure 13 – High level class diagram of the social network subsystem

Database Management Class
This class is a boundary class and it is the interface to the database. It is responsible for creating a database connection and has internal methods to build an object database connection (ODBC) with the database server. This class builds a connection to the database server during class initialization (calling the constructor).

Main Frame Class
This class is a boundary class responsible for the interaction between the user and the application. It act as the driver class of the application. Every single operation that the user needs to perform regarding the data loading, the social network creation or the metrics calculation is done through this class.

Reads Relationships Class
The loading of the forum reads is done in this class. Given the table with the forum reads, it is responsible for creating a sender-receiver relationship, counting the link repetitions and removing the duplicate links.

Message Relationships Class
This class has the same behaviour as the "Reads Relationships" with the difference that it operates with the forum message posts instead of the forum readings.

Link Generation Class
Once the reads and the posts are mined, they need to be combined and stored in a format that can be read by the social network graph. These operations are executed in this class that creates the links to feed the directed graph and undirected graph as required.

Social Network Class
This class is responsible for the creation of the social network graph, using directional relations and non-directional relations. The methods to create the social network graph, read the links generated by the "Link Generation". Depending on the option, the links read are directed or undirected.

Person Class
This is a entity class representing an actor (e.g. student, tutor, lecturer) in the social network. It keeps all the personal details that any person has like name, age, gender or citizenship.

Profile Logic Class
This control class operates on a Person object. Every piece of information required from a person is retrieved through this class.

Marks Class
Every student in the system has four different marks values represented in this entity class.

Social Network Metrics Class
This is an abstract class representing the metrics within the social network. The class has attributes that are common to both the Monadic Metrics class and the Graph Metrics class namely degree, closeness and betweenness.

Monadic Metrics Class
This entity class represents the monadic metrics that every social network actor has and was derived from the Social Network Metrics class. The class has thirteen attributes, describing the actor's participation in the network, including metrics such as eccentricity, pagerank and HITS.

Monadic Metrics Logic Class
This is a control class responsible for the calculation of the social network metrics. The metric algorithms were all implemented in this class for both a directed graph and a undirected graph. Before calculating the metrics, a method calls the Social Network to provide the social network graph. Then, each algorithm called, operates on the entire network and stores the values to the database. A method to retrieve a student's Monadic Metrics object given the student identification was also implemented.

Graph Metrics Class
This is a entity class representing the metrics of a social network graph and was derived from the Social Network Metrics class.. The system operates on different social networks and the overall behaviour of the network can be observed by the graph metrics. These metrics include the graph density, diameter or overall degree.

Graph Metrics Logic Class
This is class is responsible for calculating the graph metrics. The algorithms to calculate metrics such as density and diameter are all implemented in this class. The class also works in direct relation with the Monadic Metrics Logic as many graph metrics are dependent an the values of the Monadic Metrics. In opposition to the monadic metrics, here the metrics are calculated for just one entity at a time, therefore, the database storage only occurs when all the metrics have been calculated. A method to retrieve a Graph Metrics object given the graph identification was also implemented.

Link Metrics Class
The social network links are measured and given a weight in order to enhance the most important relations between network actors. This entity class keeps the values that define every link behaviour. These values include the number of messages sent and read between the sender-receiver pair as well as the link weight obtained from the first two values.

Link Metrics Logic Class
This control class is responsible for the calculation of the number of reads and number of messages sent between the pair sender-receiver. The class has a method that reads the tables generated by the Reads Relationships and the Message Relationships. These tables keep the number of repetitions for each of the read links and post links. The class then weights the read links differently from the message posts links and from these weights calculates a unique link weight value.

Forum Metrics Class
The forum metrics for each student are represented by this entity class. The class has sixteen attributes representing metrics such as posts and reads from lecturers, tutors, students, total number of messages receiver or total number of messages read.

Forum Metrics Logic Class
The calculation of each forum metric is performed within this class. The class has a method that loops through all the social network links calculating for each student the forum metrics. A method to retrieve a Forum Metrics object given the student identification was also implemented.

Profile Frame Class
This is a boundary class that allows the user to query a specific student within the network. Given the student identification, the student personal details, marks, forum metrics and social network metrics are displayed. The class also displays some charts to facilitate the interpretation of the metrics.

Chart Plot Class
This class is responsible for creating the charts required in the Profile Frame. Given the student identification, the class has methods to generate charts comparing the students' forum metrics, marks and social network metrics. The class generates both pie and line charts. Once created, the charts are stored to a directory from where they will be read by the Profile Frame.

The SDs generated in section 3.3.4 describes the behaviour of the Calculate Monadic Metrics use case. A class diagram was also created in order to show which the classes interact. Figure 14 models the Calculate Monadic Metrics process.

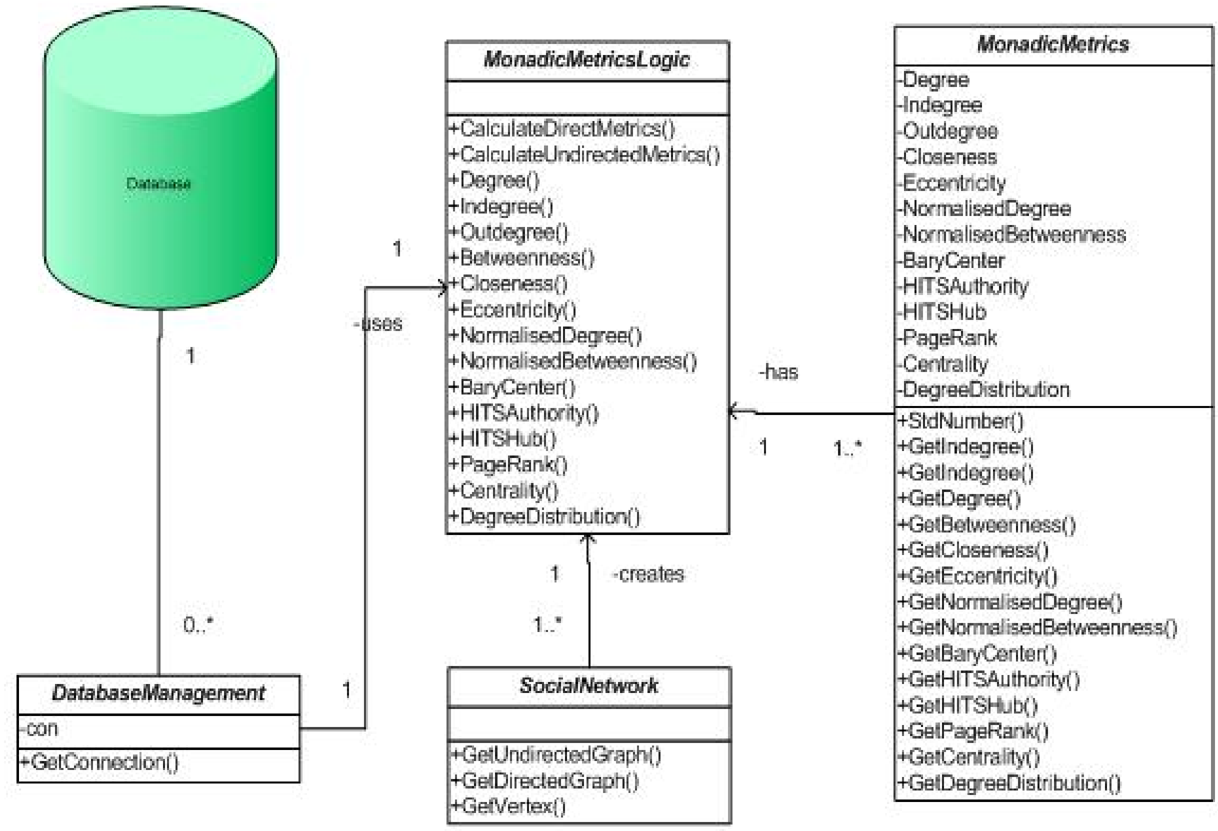


Figure 14 – Class diagram for the Calculate Monadic Metrics use case, with some attributes and operation

As can be seen, the calculation of the Monadic Metrics involves the communication between four different classes. The Database Management created the database connection that will be used by the Monadic Metrics Logic. The last one creates a social network graph to be able to generate the required metrics.

3.3.7 Interfaces

In order to perform the various operations involved in the Social Network subsystem, two main interfaces were created, the MainFrame and the ProfileFrame. The MainFrame is an interface used by the developer to execute operations in the data loading, the social network generation or the metrics calculation phase. To execute any of these operations only one input is required, the name of the social network under investigation. Once all the metrics are calculated, they must be available in a suitable format to be understood by the final user. For this reason, the ProfileFrame was created to show the different metrics calculated for a unique student. This interface is called from the MainFrame and it requires two parameters, the student number and the social network under study. The frame then shows a student's personal details, marks, monadic metrics and forum metrics for that social network. Charts are also generated to compare a student's marks with their forum participation and social network metrics.

4. Implementation

The Social Network subsystem was programmed in Java. This language was chosen primarily for its portability, flexibility and robustness. Furthermore, there are plenty of open source libraries that can be used to perform mathematical and statistical operations. The Java programming language can also be used as an integral part of SQL, crucial for a system that interacts constantly with the database. However, possibly the biggest motivation for using Java over other languages is that there exists an open source library, JUNG, that implements many of the graph metric algorithms that were mentioned in the background chapter.

4.1 Resources Required

The subsystem was built using Java JDK 1.6 running with NetBeans 6.0 as the integrated development environment. A number of external libraries were used in the system namely:

JUNG (http://jung.sourceforge.net/download.html)

The Java Universal Network/Graph Framework--is a software library that provides a common and extendible language for the modelling, analysis, and visualization of data that can be represented as a graph or network. Various graph algorithms from JUNG were used in the calculation of social network metrics.

Commons Collection (http://commons.apache.org/collections/)

Commons Collections contain implementations, enhancements and utilities that complement the Java Collections Framework. It has many powerful data structures that accelerate development of most significant Java applications. It serves as support for the JUNG library.

JFreeChart (http://www.jfree.org/jfreechart/)

JFreeChart is an open source Java chart library that allows the creation of quality charts. It was used to compare the metrics for a student.

JCommon (http://www.jfree.org/jcommon/index.php)

The library contains the common classes, which provide some global utility functionality for both GUI and non-GUI applications. It is a collection of useful classes used by JFreeChart.

4.2 Global perspective of execution

As explained in the design chapter, the subsystem is divided in three phases. Appendix B provides a summary of the information flow between components. Because the outputs of each phase are stored in the database, the order of execution is flexible as long as the values for the previous phase have already been computed. The Data Loading provides the structured data to the Social Network Generation that makes a graph available to Social Network metrics calculation. The main idea is that whenever a phase finishes their computation, the outputs namely structured data, social network graph and social network metrics are stored to the database.

4.3 Database Communication

Although the three phases provide to a large extent different services, there is a degree of commonly shared functionality. The ability to connect and authenticate with the local MySQL database is an example of one such requirement. In order to communicate with the database, the program needs a Java Database Connectivity (JDBC) connection with the database. For this project, the database used was the MySQL 5.0. Before executing any database statement, the program needs to set up a connection with the database. The format of the JDBC URL is “jdbc:mysql://<database URL” where database URL is the database location. Below, we can see the code used to set up the JDBC connection.

```
Class.forName("com.mysql.jdbc.Driver") newInstance();
String url = "jdbc:mysql://localhost:3306/projdb";
Connection con =
DriverManager.getConnection(url,username,password);
Statement stmt = con.createStatement();
```

First, the program initializes a new instance of the JDBC driver. Then the URL of the database is defined. Using the DriverManager, a connection to the database is created. To create the connection, the database URL, user name and password are required. The last two are to verify to the database server application. Finally, a statement object is created that is used for executing SQL statements.

4.4 Data Loading

The data loading section has three main responsibilities:

1. Read any new dataset and remove the information that is not relevant to the final system. The data used in the system was provided by Vula and by PeopleSoft.

2. Convert the data into a structured format. This involves having the sender and receiver for each link defined

3. Safeguard the identity of the people being investigated by applying confidentiality algorithms

4.4.1 Read and clean raw data

In order to read the data provided by the different sources, the data was first converted to csv format. The Vula forums data were already in this format but the students’ records obtained from PeopleSoft and the students’ marks where in Microsoft Excel format. The system then reads the entire data to the database by using a **LOAD DATA INFILE** SQL command. A description of the tables provided by each data source is shown below.

Table Name	Description	Data Source
Msgstructure	Stores the information regarding the posts and replies for each forum participant. Each message record contained the date and time, the sender, the title, the message replied to and the message topic.	Vula Administration
Events	Stores every event occurred within the specified course site (PSY1001W 2007). Every click for a resource, a message posted or a message read can be seen in this table. Each event record contained the date and time, the user, the event type and the event description.	Vula Administration

Topics	This table shows to which topic each message was posted. Each record contained the topic ID and the topic name.	Vula Administration
Student Codes	Maps encrypted codes representing a forum participant in the tables above, with their UCT code. Each record contained a student's encrypted code and the respective student number.	Vula Administration
Marks	Gives the year performance of each student in the PSY1001W 2007 course. For each student, the marks for every assignment and test were given.	Vula PSY1001W 2007 site
Student Records	Contain the personal details of each student in the PSY1001W 2007 course. The students' matric marks and final year marks for the PSY1001W course were also given here. Each record contained a student's name, surname, citizenship, ethnicity, matric school, matric mark and final year mark.	PeopleSoft

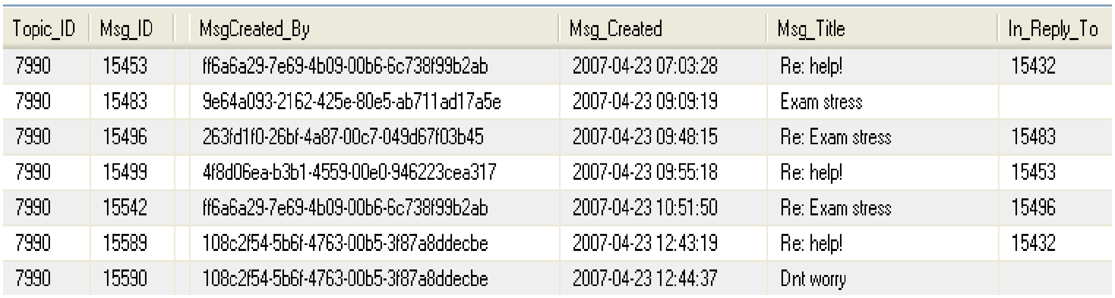
Table 1 – The data provided with the respective description and origin

Once in the database, the data could then be investigated and the irrelevant information removed. The main operations performed were the:

- Creation of a table containing just the forum readings
- Removal of the data that did not lie in the period of investigation
- Mapping of each forum reading and post to the respective topic

4.4.2 Structure the data

To facilitate the creation of the social network, the links stored in the database must be represented by the ID of the sender and the receiver. This operation is performed for both the posts and readings tables. The process of analysing messages from a forum is not the same as emails for instance. When analysing emails, for every message we always have a sender and a receiver. However, when using forums, we do have a sender but there is no receiver until someone reads the message. For this reason, only the replies are considered in the message posts table. Below, we can see the initial format of the forum posts table.



Topic_ID	Msg_ID	MsgCreated_By	Msg_Created	Msg_Title	In_Reply_To
7990	15453	ff6a6a29-7e69-4b09-00b6-6c738f99b2ab	2007-04-23 07:03:28	Re: help!	15432
7990	15483	9e64a093-2162-425e-80e5-ab711ad17a5e	2007-04-23 09:09:19	Exam stress	
7990	15496	263fd1f0-26bf-4a87-00c7-049d67f03b45	2007-04-23 09:48:15	Re: Exam stress	15483
7990	15499	4f8d06ea-b3b1-4559-00e0-946223cea317	2007-04-23 09:55:18	Re: help!	15453
7990	15542	ff6a6a29-7e69-4b09-00b6-6c738f99b2ab	2007-04-23 10:51:50	Re: Exam stress	15496
7990	15589	108c2f54-5b6f-4763-00b5-3f87a8ddecbe	2007-04-23 12:43:19	Re: help!	15432
7990	15590	108c2f54-5b6f-4763-00b5-3f87a8ddecbe	2007-04-23 12:44:37	Dnt worry	

Figure 15 – The initial format of the forum posts records

First, the UCT number replaced the code defining each person by using the table provided by Vula. Then, a table was created to store both the message posts and reads links. This table has the following schema:

DirectedLinks(Sender, Receiver, LinkType, Count)

For the reading table, it is considered the sender the one who reads a message and the receiver the one who posted the message. This assumption was made based on the metrics definition from the background chapter where a node that has many links going in has more importance than one that has many links going out. Therefore, someone that has his/her posts read many times has more value than one that just read many posts. The LinkType value just shows whether a link is from a reading or from a message reply. The number of reads and replies for each node pair is then calculated and stored in the count field while the link repetitions are deleted. A similar table, named UndirectedLinks, was also created were a link from A to B is considered the same as a link from B to A. This table only stores the sender and receiver and is used to create the undirected graph while the DirectedLinks is used to create the directed graph. In addition to the links table, the marks and profile tables for each student are also finalised in this phase. The full database structure is provided in Appendix C. Many of the relations share the StdNumber attribute as most of the tables are describing a student in some way.

4.4.3 Confidentiality

To protect the identification of the social network participants to the final user, mask names were created for each person. These mask names were created using the set of names from the network. Once we have the gender of each network participant, included in the records from PeopleSoft, a proper name can be given to each person according to their gender. A person's first name was randomly chosen from the set of people with the same gender. The surnames were randomly chosen from the entire set as they are not influenced by a person's gender.

4.5 Social Network Generation

In order to generate the social network, the respective links must be already formed. A method was created to read the table containing the social network links, create the social network graph and populate the graph. Once we have the links, the number of unique nodes in the network is calculated. A sender or receiver represents a node.

The system uses JUNG, defined in the beginning of the chapter, as a framework on which applications and tools for manipulating graph and network data can be built. The process of generating and populating a social network graph can be divided in two parts, the vertices insertion and the edges insertion. First, to create a new graph, the following code is executed:

```
UndirectedSparseGraph g = new UndirectedSparseGraph();
```

The code above creates a new undirected graph. Once a graph has been created, vertices are also created and added to this graph:

```
Vertex v;
for(int i= 0;i<temp.size();i++)
{
  v= new UndirectedSparseVertex();
  v.addUserDatum("StdNumber",temp.get(i),UserData.SHARED);
          g.addVertex(v);
}
```

First, a new Vertex object is created and then a loop runs for each node in the social network. The variable “temp” is a vector created when counting the unique social network nodes and each position keeps the student number of a node. Therefore, if the social network size is 300, the vector “temp” will store 300 unique student numbers. For each social network node, a new vertex is created and then populated with the node student number. At the end, the vertex is added to the undirected graph.
The last step in the graph population is edge insertion:

```
    UndirectedSparseEdge tempedge;
    while(rs.next())
    {
       V1 = GetVertex(g,rs.getString("Sender"));
       V2 = GetVertex(g,rs.getString("Receiver"));
       tempedge= new UndirectedSparseEdge(v1,v2);
      tempedge.setUserDatum("ID",rs.getInt("ID"),UserData.SHARED);
       g.addEdge(tempedge);
    }
```

In the edges insertion, a new edge object is created and then a loop is executed for each social network link. The variable “rs” is a data structure storing the links table values. The GetVertex method, takes a student number as a parameter and returns the corresponding vertex object. So, v1 and v2 are populated with the vertices of the sender and receiver respectively. A new edge is then created between the two vertices and added to the graph. When the graph vertices and edges are inserted, the method returns the graph. The process described above, shows how an undirected graph is created. A method to create a directed graph was also implemented that reads the table containing the directed links and just replaces UndirectedSparseGraph, UndirectedSparseVertex and UndirectedSparseEdge for DirectedSparseGraph, DirectedSparseVertex and DirectedSparseEdge respectively.
The social network graph can also be created for different networks. This is specified in the Social Network class constructor that takes the name of the graph to be generated as a parameter.
The social network being investigated by the system is the PSY1001W course for 2007. This course has four main assessments throughout the year namely essay 1 (middle of first semester), June exam (end of first semester), essay 2 (middle of second semester) and final exam (end of year). Each of these assessments has a main topic within the forum from which a smaller network can be created. Students’ marks for each of the four assessments were also provided. Therefore, in addition to the main network representing the interactions throughout the whole year, four smaller networks were created representing each of the assessment’s forum topics

4.6 Social Network Metrics

The social network metrics implemented for the system can be divided in four different groups as described in the design chapter.

4.6.1 Monadic Metrics

The monadic metrics are the metrics that are going to be used by the Bayesian network and they describe the behaviour of a node individually. All the metrics discussed in the background chapter were implemented, many supported by JUNG libraries. These algorithms run one at a time on the entire graph and when each finishes processing the network, the values obtained are sent to the database. The algorithms, require a graph object, therefore, the social network class discussed above is called to return a directed or undirected graph depending on the metrics being calculated. The code implemented to calculate some of the metrics can be seen below.

```
UnweightedShortestPath usp = new UnweightedShortestPath(g);
for(int i=0;i<usg.numVertices();i++)
{
    for(int j=0;j<usg.numVertices();j++)
    {
        if(usp.getDistance(ind.getVertex(i),
ind.getVertex(j))==null)
        {}
        else
        {
           V1= ind.getVertex(i);
           V2= ind.getVertex(j);
           count= count + usp.getDistance(v1, v2);
         }
     }
     closeness = 1/count;
     if(count>0)
          M[i]=closeness;
     else
           M[i]=0;

   closeness=0;
   count=0;
}
```

The code above implements the closeness centrality metric that shows how close an actor is to all the other actors in the network. First, an object to compute the shortest path distance for the graph is created. Then, the two nested loops calculate for each graph node, the sum of the distances to all other nodes. The inverse of the value is then calculated since centrality is inversely proportional to distance as discussed in the background chapter.
Some metrics were also calculated using JUNG algorithms. Below is an example where the degree distribution is implemented.

```
DegreeDistributionRanker ranker = new DegreeDistributionRanker(g);
ranker.setRemoveRankScoresOnFinalize(false);
ranker.evaluate();
for(int i=0;i<usg.numVertices();i++)
{
    M[i]=ranker.getRankScore(ind.getVertex(i));
}
```

First a degree distribution object is created passing the graph object as parameter. The second line is telling the object not to remove the metric values from the nodes once the nodes have been computed. The evaluate() performs the metric calculation but does not return any value. Then, the loop goes through each node and returns the correspondent metric value. The other metrics calculated using JUNG libraries were the betweenness centrality, bary center, pagerank and HITS.

The entire set of metrics was calculated for both a directed graph and undirected graph, with the exception of pagerank that is only defined for a directed graph. This was done for each of the five social networks discussed above.

4.6.2 Graph Metrics

The graph metrics require the monadic metrics to have been calculated previously. Each of the metrics discussed in the background chapter was implemented for the five social network graphs investigated by the system. Besides the number of nodes and number of edges, the other metrics calculated for each graph were the diameter, the density and the three centrality metrics. The implementation of the density and the group degree centrality is shown below.

```
double density=0;
int div= Size()*(Size()-1);
while(rs.next())
{
     density=density+rs.getDouble("Degree");
}
return density/div;
```

```
int maxdegree=0;
int div = (Size()-1)*(Size()-2);
while(rs.next())
{
     if(rs.getInt("Degree")>maxdegree)
     {  maxdegree=rs.getInt("Degree");}
 }
 rs.beforeFirst();
 int degree=0;
 while(rs.next())
 {
      degree= degree+(maxdegree-rs.getInt("Degree"));
 }
 return degree/div;
```

The implementation of the three centrality metrics is similar just varying the value for the denominator. The diameter is just the maximum eccentricity value in the graph.

4.6.3 Dyadic Metrics

As the system keeps track of the number of messages sent between two nodes A and B and the number of times that A read B’s posts, a unique weight was calculated to measure the link importance. Both the number of messages and number of reads were calculated in the data loading phase. Because the forum replies illustrate a stronger relationship between two people, the system gave a single reply the weight of five reads. This value is a parameter defined by the user and can be changed whenever the user wishes. So,

Link Weight A→B = # of B’s Posts Read by A + 5*(# of Msgs sent from A to B)

To normalise the weight values, each link weight is divided by the average of the link weights. The weight above is defined for all directed links. When undirected links are used, the weight is just the sum of the weight from A to B and from B to A.

4.6.4 Forum Metrics

The forum metrics give extra information to the final user regarding the behaviour of each student within the forum. For each student the following values are calculated:

- Number of replies sent
  Number of times the student replied to forum posts. A student can reply more than once to a unique forum post.

- Number of replies received
  Number of times the student's posts were replied.

- Number of posts read
  Number of other people's posts read by the student

- Number of times that his/her posts were read
  Number of times that other people read the student's posts

- Number of replies to lecturers or tutors
  Number of times the student replied to messages posted by lecturers or tutors

- Number of replies received from lecturers or tutors
  Number of times that the lecturers or tutors replied to the student's posts

- Number of lecturers' or tutors' posts read
  Number of times that the student read messages posted by lecturers or tutors

- Number of times lecturers or tutors read his/her message
  Number of times that lecturers or tutors read the student's posts

4.7 Metrics Presentation

As discussed in the Design chapter, an interface class was implemented representing the profile of a student. The class displays a student's personal information, marks and relevant metrics. To provide the user a more friendly and efficient way to view the metrics, charts were implemented showing different social network information. Two examples are shown in figure 16 and figure 17. The first one shows whose forum postings a specific student reads and the second one compares a student's forum participation with the class average participation for each of the four social networks generated.

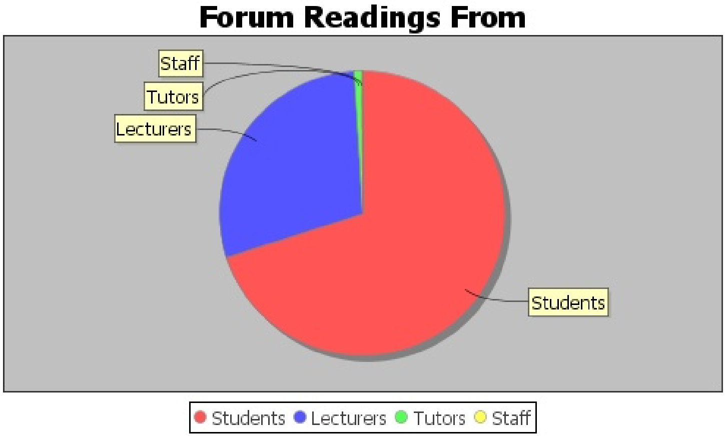


Figure 16 – A pie chart showing whose forum postings a specific student reads

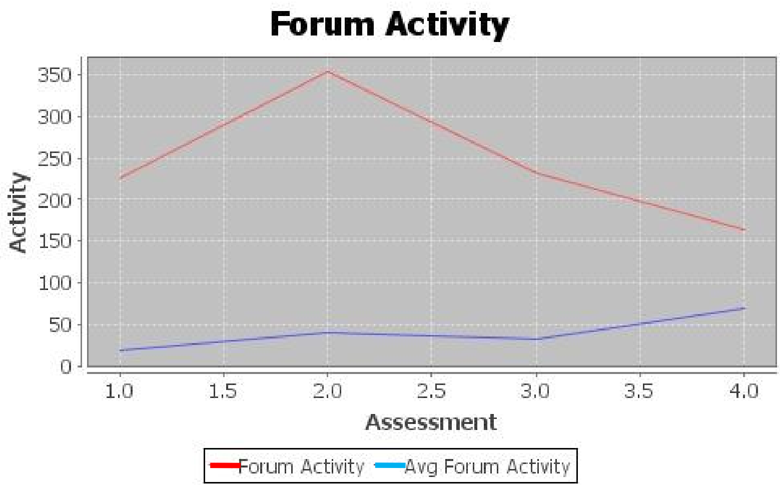


Figure 17 – A line chart comparing a student's forum participation with the class average

The charts are implemented in a class called ChartPlot that uses the JFreeChart library to draw the charts. The chart methods use data stored in the database and output the charts to the /resource/ directory. The code below shows the implementation of the pie chart in figure 16.

```
ForumMetricsLogic fml = new ForumMetricsLogic(con,graph);
ForumMetrics fm = fml.DBRetrieval(StdNumber);

int streads = fm.ReaderTimes-fm.LSReads-fm.TSReads-fm.StaStdReads;

DefaultPieDataset pieDataset = new DefaultPieDataset();
pieDataset.setValue("Students", new Integer(streads));
pieDataset.setValue("Lecturers", new Integer(fm.LSReads));
pieDataset.setValue("Tutors", new Integer(fm.TSReads));
pieDataset.setValue("Staff", new Integer(fm.StaStdReads));
JFreeChart chart = ChartFactory.createPieChart(
                   "Forum Readings From",
                    pieDataset,
                    true,
                    true,
                    false);

ChartUtilities.saveChartAsJPEG(new File("/resources/chart4.jpg"), chart,
500, 300);
```

The method first creates a ForumMetricsLogic object that has the method to retrieve a student's metric from the database. The student number of the stuent to be plotted is sent as a method parameter. The metrics are retrieved and stored in a metrics object. A dataset is then created and populated with the values that will be displayed in the chart. A chart objected is created and the dataset passed as parameter. Lastly, the chart object is saved in JPEG format to the /resources/ directory.

5. Integration

As discussed in the Design chapter, the three parts of the SONET system communicate through the database. In the Social Network subsystem, the main outputs are the social network that is going to be used by the visualization subsystem and the social network metrics that are going to feed the Bayesian network. Figure 18 provides a summary of the information flow between the components of the Social Network subsystem (Data Loading, Social Network and Metrics).

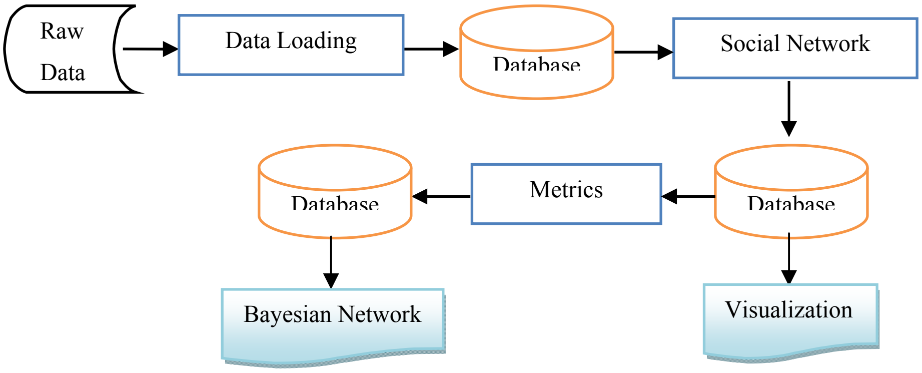


Figure 18 – The cycle of the Social Network subsystem

As can be seen, after each operation, the outputs are stored in the database. The Bayesian network and the visualization read the metrics and the social network respectively from the database.

Visualization

The social network is stored in a table called “UndirectedLinks”. This table stores all the links within the social network. Each link contains the sender, the receiver and the social network being investigated, it can be seen in Appendix C. The table has the following schema:

UndirectedLinks(<u>Sender, Receiver</u>, <u>Topic</u>)

The visualization also uses the Person, Marks and MonadicMetrics classes to show a student’s profile and social network position.

Bayesian Network

The Bayesian network uses the social network metrics to profile students and predict students’ performance. The metrics to be used by the Bayesian network are stored in a table called MonadicMetrics. This table schema is shown below:

MonadicMetrics(<u>StdNumber, </u>degree,betweenness, closeness, eccentricity, centrality, normaldegree, normalbetweenness, degreedistr, pagerank, barycenter, HITSAut, HitsHub)

This table can be seen in Appendix D.

6. Results and Testing

6.1 Social Network Results

As referred in the previous chapters, the SONET system was implemented for the PSY1001W 2007 site. The site was composed by students, tutors, lecturers and administrative staff. Table 2 below shows the proportion of each group within the site.

Group	Number of participants	%
Students	448	94,3
Tutors	13	2,8
Lecturers	12	2,5
Administrative Staff	2	0.4

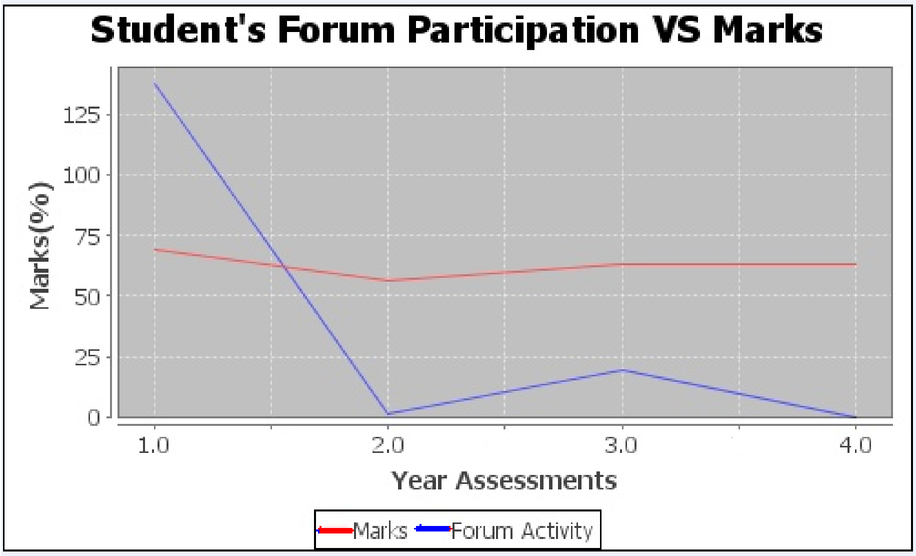
Table 2 – Proportion of each group in the PSY1001W site

Within the site main forum, 69 different sub forums were available. In addition to the main forum, 4 other sub forums were investigated. These 4 sub forums relate to the main course assessments namely Essay 1, June Exam, Essay 2 and Final Exam. The social network size for each of the networks created is shown in Table 3 below.

Social Network	Number of nodes
Full Network	475
Essay 1	175
June Exam	248
Essay 2	71
Final Exam	77

Table 3 – Different social networks and their sizes

These 4 forums were selected because they were large enough to create a social network ad perform the respective analysis. As stated above, these 4 forums correspond to the course 4 main assessments. Therefore, a comparison between a student's social network participation and the mark can be done. The graph below shows, for a specific student, how the marks and the forum participation change across the different networks.



Year Assessments
1 – Essay 1
2 – June Exam
3 – Essay 2
4 – Final Exam

Figure 19 – Marks and forum participation change across different networks

Considering that Essay 1 and June Exam are in the first semester and Essay 2 and Final Exam are in the second semester, it is clear that the forum participation was much more intense during the first six months of the year.

The full network was investigated in order to see who the most participative students are. The subset of students that has degree centrality above average was selected. Then, the following graph was plotted:

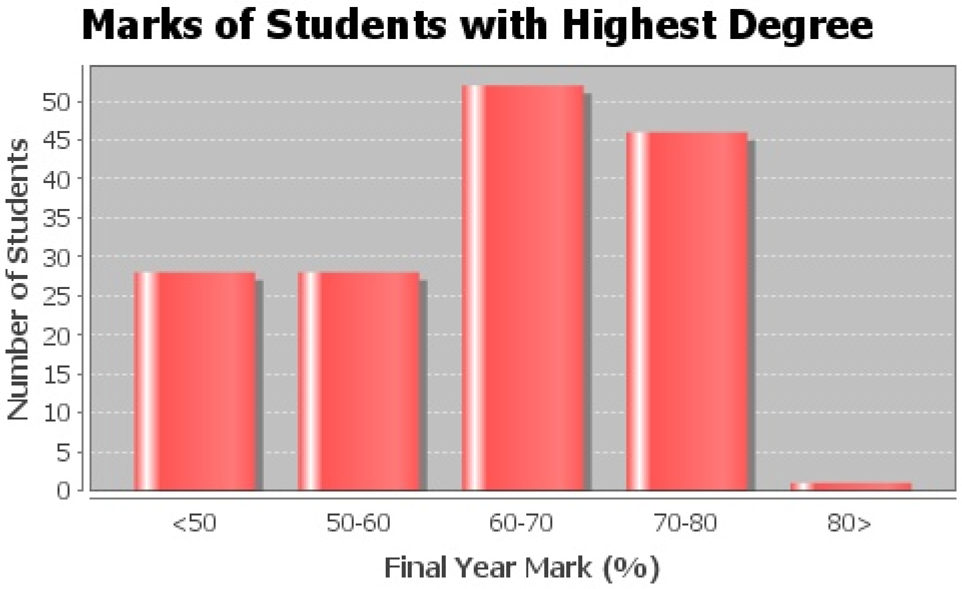


Figure 20 – Mark range of students with highest degree centrality

The graph shows that the students that have final year marks between 60% and 70% are the ones connected to most people within the network (degree centrality).

The graph above shows the number of connections that a student has, whether they are from reads or replies. To see whose students participated more actively in the forum with replies and not with reads, the following graph was plotted:

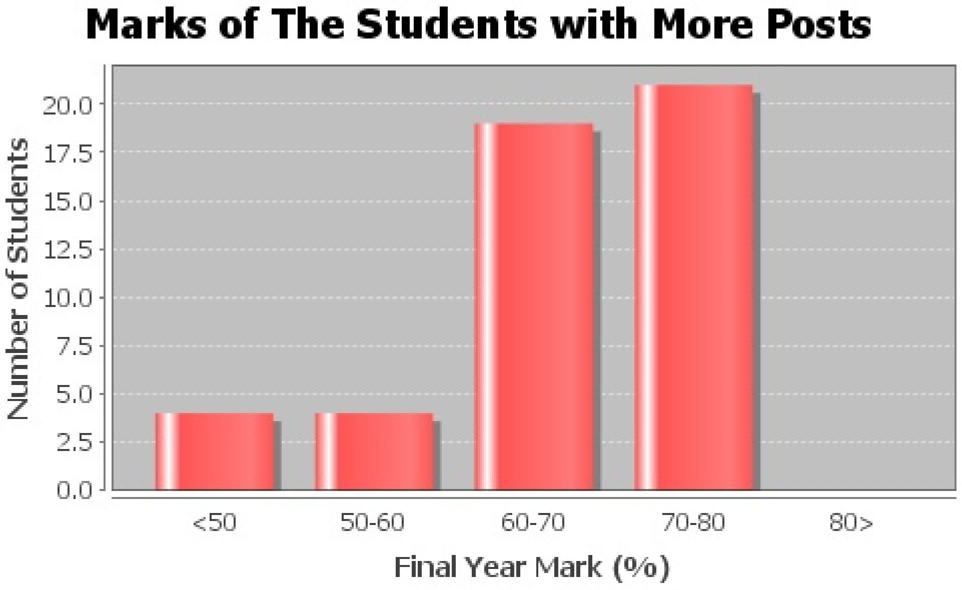


Figure 21 – Mark range of students with more reply posts

As can be seen, the gap between the intervals below 60% and above 60% increased comparing to the degree centrality graph. This graph also shows that the most active students in terms of message postings are the ones lying in the interval of 70% to 80%.

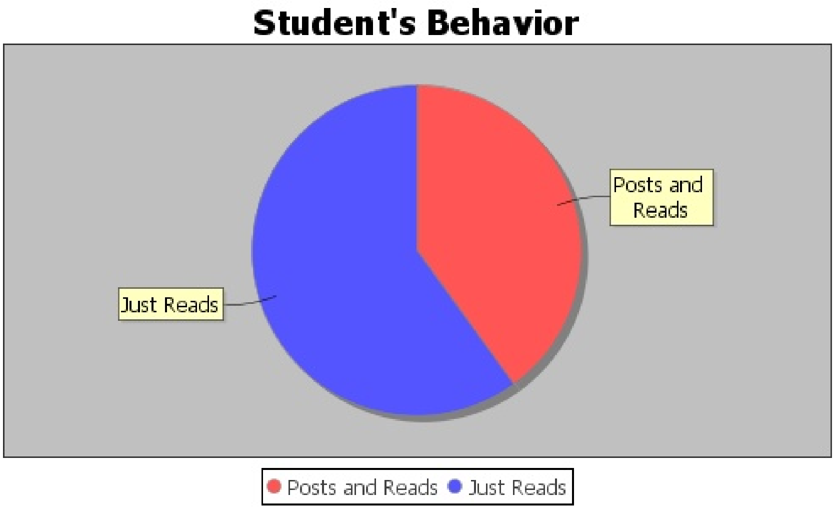


Figure 22 – Posts and reads analysis

Figure 22 above, shows that in a network of 475 participants, only 190, 40%, post messages while the others 285, 60%, just read. It was also possible to see that a single participant, the main lecturer, posted 27% of the total forum posts and three students, without the lecturer, posted 28% of the total forum posts. When looking to just the reads, we can see that six students make up 26% of the forum readings.

As no performance predictions could be made based on the forum's participation, a new hypotheses was tested. Even though forum participation can't predict future marks, it can still be an indicator of the student's performance. This was tested by comparing the matric and the final year results of the most participative students. Each mark was normalized with the class average in order to provide a valid comparison meaning that both the matric marks and the final marks were divided by the respective class average. The results can be seen from the graph below.

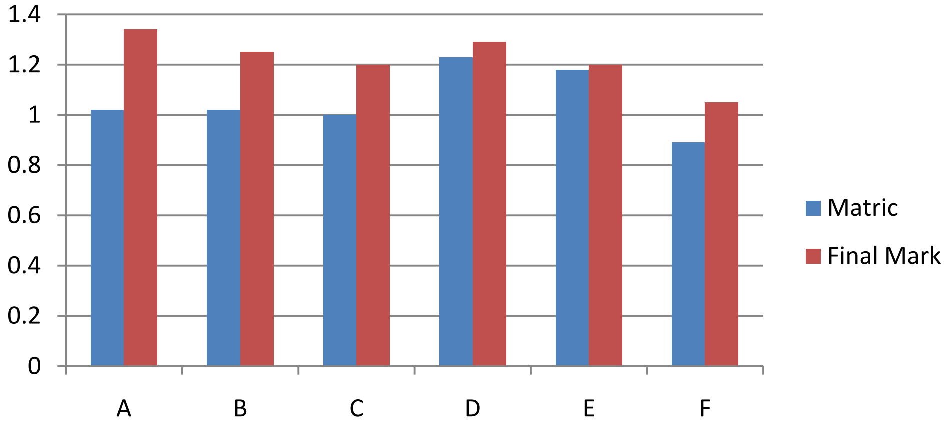


Figure 23– Comparison between matric marks and final marks for students with high forum participation

This graph gives us two main pieces of information. First, it is clear that all the students with high forum participation improved their marks. This can be seen from the difference between the red bars and the blue bars. Second, knowing that the class average mark is at 1, we can see that the students also improved in comparison to the class.

6.2 Metrics Testing

The only test required for this subsystem was the correctness of the metrics calculated. Because the metrics were calculated for both directed and undirected graphs, two tests were performed to measure the accuracy of the results obtained.

6.2.1 Undirected Graph Metrics

To test the undirected graph metrics, a dataset from Orgnet.com was used. Orgnet.com provides social network analysis software & services for organizations, communities, and their consultants. Clients such as IBM, TRW, Google, Northrop Grumman use Orgnet.com software and services to map and measure networks in organizations, communities, and other complex human systems. Figure 17 shows the graph provided by orgnet.com.

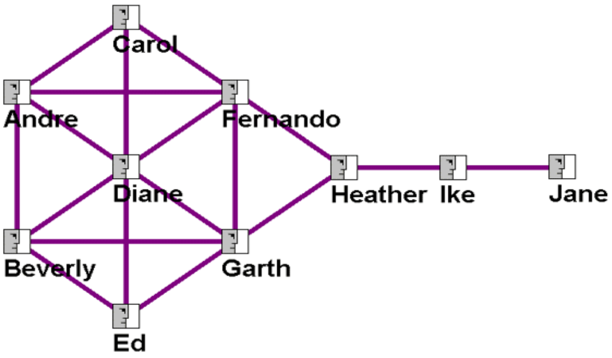


Figure 24 – Orgnet.com social network

To test the system, a social network graph was created using the diagram shown above and the centrality metrics calculated for each of the network's participants. The Orgnet graph consisted of 10 nodes and 18 links. The centrality values given by orgnet.com and the values obtained by the SONET system are shown are shown in table 4. The values displayed, are the normalized values for each of the metrics.

	Orgnet Degree	SONET Degree	Orgnet Betweenness	SONET Betweenness	Orgnet Closeness	SONET Closeness
Diane	0.667	0.667	0.102	0.102	0.6000	0.6000
Fernando	0.556	0.556	0.231	0.231	0.643	0.643
Garth	0.556	0.556	0.231	0.231	0.643	0.643
Andre	0.444	0.444	0.023	0.023	0.529	0.529
Beverly	0.444	0.444	0.023	0.023	0.529	0.529
Carol	0.333	0.333	0.000	0.000	0.500	0.500
Ed	0.333	0.333	0.000	0.000	0.500	0.500
Heather	0.333	0.333	0.389	0.389	0.600	0.600
Ike	0.222	0.222	0.222	0.222	0.429	0.429
Jane	0.111	0.111	0.000	0.000	0.310	0.310

Table 4 – Centrality metrics calculated by Orgnet.com for a undirected graph

As can be seen from table 4, the values obtained by SONET are the same as the ones given by Orgnet.com. Therefore, the metrics calculated for an undirected graph can be considered valid.

6.2.2 Directed Graph Metrics

The metrics for a directed graph were tested by using a dataset available at the Social Network Analysis department in the University of Essex [20]. Figure 25 shows the graph provided in [20].

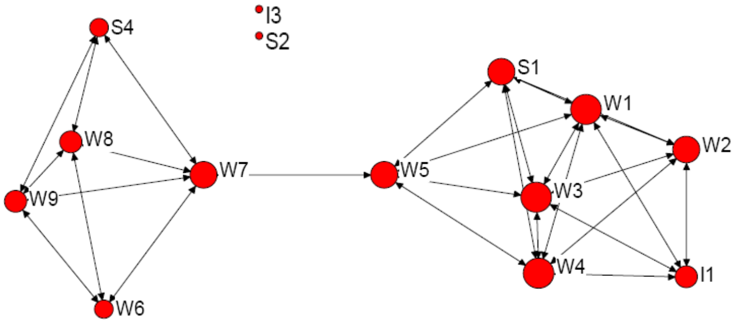


Figure 25 – University of Essex social network

In opposition to the graph provided by Orgnet.com, this graph has directed links and from the fourteen nodes represented, two are not connected to the main graph. The graph above was created in the SONET system and the centrality metrics calculated. The centrality values given in [20] and the values obtained by the SONET system are shown below.

	Borgatti Degree	SONET Degree	Borgatti Betweenness	SONET Betweenness	Borgatti Closeness	SONET Closeness
I1	4	4	0	0	27	27
I3	0	0	0	0	0	0
W1	6	6	6.82	6.82	20	20
W2	5	5	0.45	0.45	26	26
W3	6	6	6.82	6.82	20	20
W4	6	6	6.82	6.82	20	20
W5	5	5	54.55	54.55	17	17
W6	3	3	0	0	27	27
W7	5	5	51.52	51.52	19	19
W8	4	4	0.61	0.61	26	26
W9	4	4	0.61	0.61	26	26
S1	5	5	2.73	2.73	21	21
S2	0	0	0	0	0	0
S4	3	3	0	0	27	27

Table 5 – Centrality metrics calculated from the University of Essex for a directed graph

The values available in [20] were not in a normalized format. For this reason, the values computed by the SONET system were outputted in the same format. From table 5, we can see that for all the centrality metrics, the values obtained are the same as the ones from the original data source.

The other metrics produced by the SONET system, namely the web link metrics and the graph metrics were tested manually. A small graph was created and applying the formulas given in the background chapter, the values for the different metrics were proved for correctness.

7. Evaluation

7.1 Social Network Subsystem

The social network subsystem produces two main outputs namely the social network and the social network metrics that needed to be evaluated. The success or failure of a social network generated can be seen from the visualization subsystem that visualizes the social network. If the visualization is able to show the social network with all the nodes and links, then the generation of the social network can be considered a success. The metrics calculated were also tested for correctness. Two different data sources were used and the results obtained prove that the metrics calculated are valid as can be seen in the previous chapter. The results illustrated in the previous chapter also show that the majority of students with high forum participation have a good scholar progress as they lie in the interval 60%-80%. We can also see that for the site under investigation, the forum participation is focused on the same 6 to 8 participants being the main lecturer the most active. The social network analysis also proved to be an effective technique to identify the main actors within a network. The fact that the main social network actors are lecturers and students with positive progress also shows that these main actors truly represent knowledge hubs.

7.2 SONET System

The SONET system allows the identification of knowledge hubs within a social network, profiles students based on their network position and on their marks and finally visualizes the social network highlighting the key actors based on their centrality metrics. For these reasons, the SONET system is a success. In this project, it was learnt that all the results obtained are valid as long as they are correct and as long as they reflect the reality, even when they go against our wishes. We experienced this as we were investigating whether there was a correlation between forum's participation and final marks. SONET shows that for the site under investigation, no strong correlation was found between the two parameters meaning that the forum participation does not influence the final marks. Therefore, another test was implemented to test if students that use forums heavily tend to improve their Matric marks. The results showed that the students with high forum participation got a better performance in comparison to the others. Another successful aspect is the fact that the system was proposed and will be presented in a conference with the topic, Teaching with Technology. The SONET system is also being seen as a useful tool to all lecturers who teach using discussion forums and a possible plug in to Vula sites.

8. Conclusion

8.1 Success of the Social Network subsystem

In conclusion, the social network subsystem can be considered a success. All the functional requirements were met and both the Bayesian Network and Visualization were satisfied with the inputs provided by the system. Not just the social network metrics were implemented successfully but also forum metrics showing a student's participation in the forum were generated. These metrics can also be visualised by the final user in the form of charts.

8.2 Social Network Analysis

The social network analysis proved to be an effective technique to identify the key players in a social network. The metrics generated reflect accurately the importance and position of each participant within the social network forum.

8.3 Vula Forums

From the forum analysis, we saw that the main participant within the network was the course main lecturer who contributed with approximately a quarter of the total forum activity. Only a third of the class posted messages in the forum while the other two thirds just reads.

8.4 Summary of project

Even though significant relations between a student's forum participation and the marks not have been found, overall the project can be considered a success. Not just each of the subsystems met the desired requirements but the system as a whole performs the tasks in an integrated manner. The SONET system identifies the knowledge hubs within a network, compares the forum participation with performance, profiles students based on their network participation and their personal details and visualizes the social network. SONET proved to be a useful system and it can be used in future as a tool to aid and enhance the flow of knowledge across a social network.

9. Future Work

9.1 Text Mining

Actually the SONET system creates relationships between students based on the messages read and replied without looking to the message content. To strengthen the relationships between the students, text mining and content analysis could be performed on the posts. This would also extend the users' profiles with the type of content that they post and read.

9.2 Vula Customization

The SONET system could easily be incorporated into Vula as an additional feature available for the lecturers. The actual system is already automated meaning that it would just need to be adapted to suit the new users' requirements. The tool would display the course social network and highlight, based on the lecturer preferences, the most important actors within the network. As all the Vula sites are stored in the same format as the PSY1001W, used by the SONET system, the data loading process would not be a problem. Batch files can be ran overnight to convert the daily forum activity to a social network format and calculate the social network metrics. In addition to the social network analysis, the tool could also recommend some measures to be taken by lecturers to improve the progress of the weakest students. The formation of groups combining strong and weak students is an example of a recommendation that could be given by the system. After N years of Vula data, different studies can also be performed. Three years of data would already allow one to investigate the behavior of a student throughout his/her degree. Courses within the same degree can also be analyzed simultaneously to investigate if the students have the same behavior in different courses. The behavior of a social network throughout the years, from first year up to the last year can also be studied.

9.3 SONET

Once the SONET system for learning environments was a success and produced valid results, it can be adjusted to new environments. SONET can investigate the flow of knowledge and identify the knowledge hubs within a business context. Here, the main user would be the person responsible for the business activity and would be interested in identifying the company's knowledge hubs as well be able to predict workers' performance. The profiling of people within a business context would differ from the profiling in learning environments. Parameters such as the number of years worked in the company or the salary would play an important role in defining a person's profile.

10.References

[1] Carley, K. Dynamic Network Analysis forthcoming in the Summary of the NRC workshop on Social Network Modeling and Analysis, R. Available from http://www.stiet.si.umich.edu/researchseminar/Winter%202003/DNA.pdf (2003); accessed 10 July 2008.

[2] Cooke, R. Link Prediction and link detection in sequences of large social networks using temporal and local metrics, *Master Theses CS06*. Department of Computer Science, University of Cape Town, 2006.

[3] Dekker, A. Visualization of Social Networks using CAVALIER, *Australian Symposium on Information Visualization*, Sydney, December 2001.

[4] Gibbons, A. Algorithmic Graph Theory. Cambridge University Press, Cambridge, 1985.

[5] Hanneman, R. Introduction to Social Network Methods.
Available from http://faculty.ucr.edu/~hanneman/nettext/networks.zip (2001); accessed 10 July 2008.

[6] Haythornthwaite, C. Social Network Methods and Measures for Examining E-learning: University Illinois at Urbana-Champaign. Available from http://www.lis.uiuc.edu/haythorn (2005); accessed 10 July 2008.

[7] Huang, Z., Li, X., and Chen, H. 2005, Link Prediction Approach to Collaborative Filtering. In Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries, (June 7 – June 11).

[8] Krackhardt, D. Graph Theoretical Dimensions of Informal Organizations. In Computational Organization Theory, K.M. Carley and M.J. Prietula, Eds. Lawrence Erlbaum Associates, Hillsdale, NJ, 1994, 89-111.

[9] Saltz, J., Hiltz, S., and Turoff, M. Student Social Graphs: Visualizing a Student's Online Social Network. ACM, New York, NY, 2004.

[10] Wasserman, S. and Faust, K. *S*ocial Network Analysis: Methods and Applications. Cambridge University Press, Cambridge, 1994.

[11] Wasserman, S. and Scott, J. Models and Methods in Social Network Analysis. Cambridge University Press, Cambridge, 2005.

[12] Wasserman, S. and Galaskiewicz, J. Advances in Social Network Analysis. Sage, London, 1994.

[13] Trudeau, R.J. Introduction to Graph Theory. Dover Publications, New York, 1993.

[14] Laumann, E.O., Marsden, P.V., and Prensky, D. The boundary specification problem in network analysis, George Mason University Press, 1989.

[15] Harary, F., Norman, R.Z., and Cartwright, D. Structural Models: An Introduction to the Theory of Directed Graphs. Wiley, New York, 1965.

[16] Ng, A.Y., Zheng, A. X., and Jordan, M.I. Link Analysis, Eigenvectors and Stability. In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, Seattle, 2001, pp. 903-910.

[17] Kleinberg J. Authoritative sources in a hyperlinked environment. Journal of the ACM. 46, (1999), 604-632.

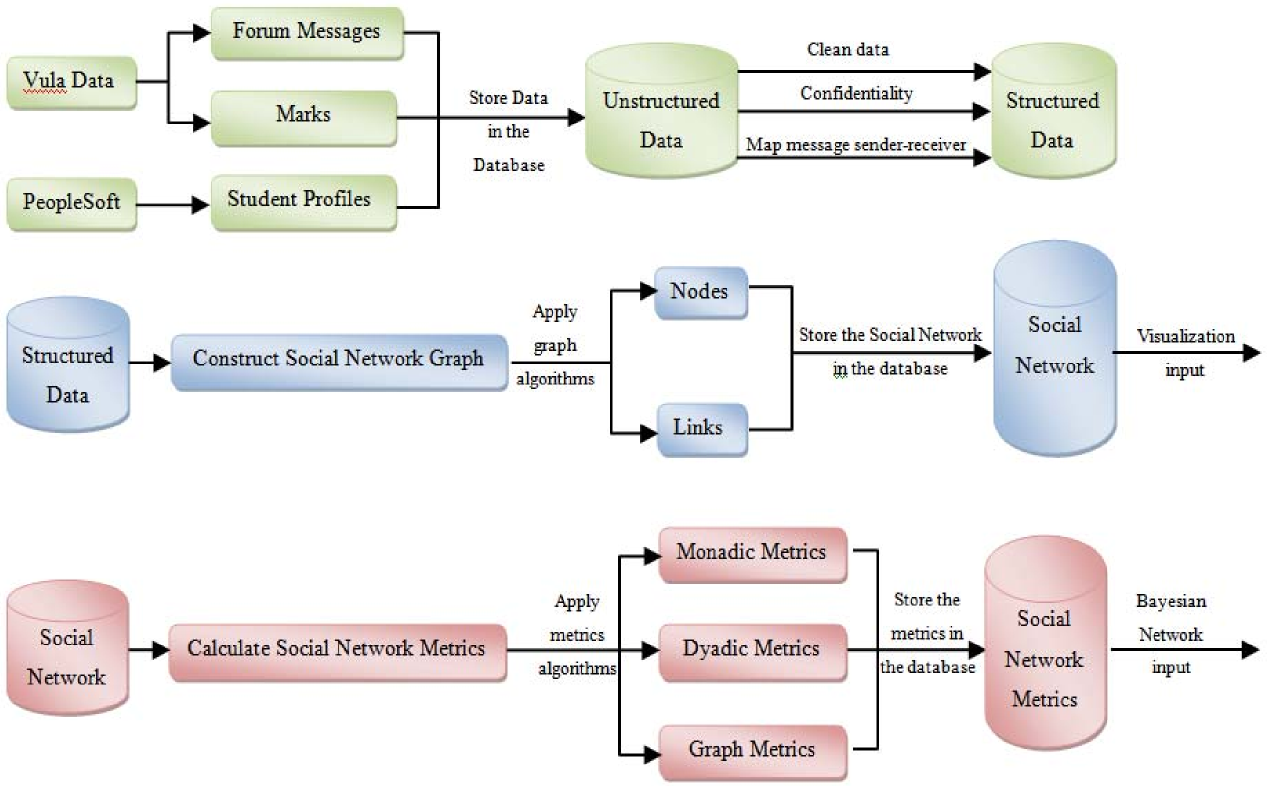
[18] Brin, S., and Page, L. The anatomy of a large-scale hypertextual (web) search engine. In Proceedings of the Seventh International Conference on the World Wide Web, 1998.

[19] Schneiderman, B., Perera, A. Balancing Systematic and Flexible Exploration of Social Networks. Available from http://www.hcil.cs.umd.edu/trs/2006-25/2006-25.pdf (2006); accessed July 10 2008.

[20] Borgatti, S. 1H Social Network Analysis. Available from http://www.analytictech.com/essex/schedule.htm (2007); accessed 10 July 2008.

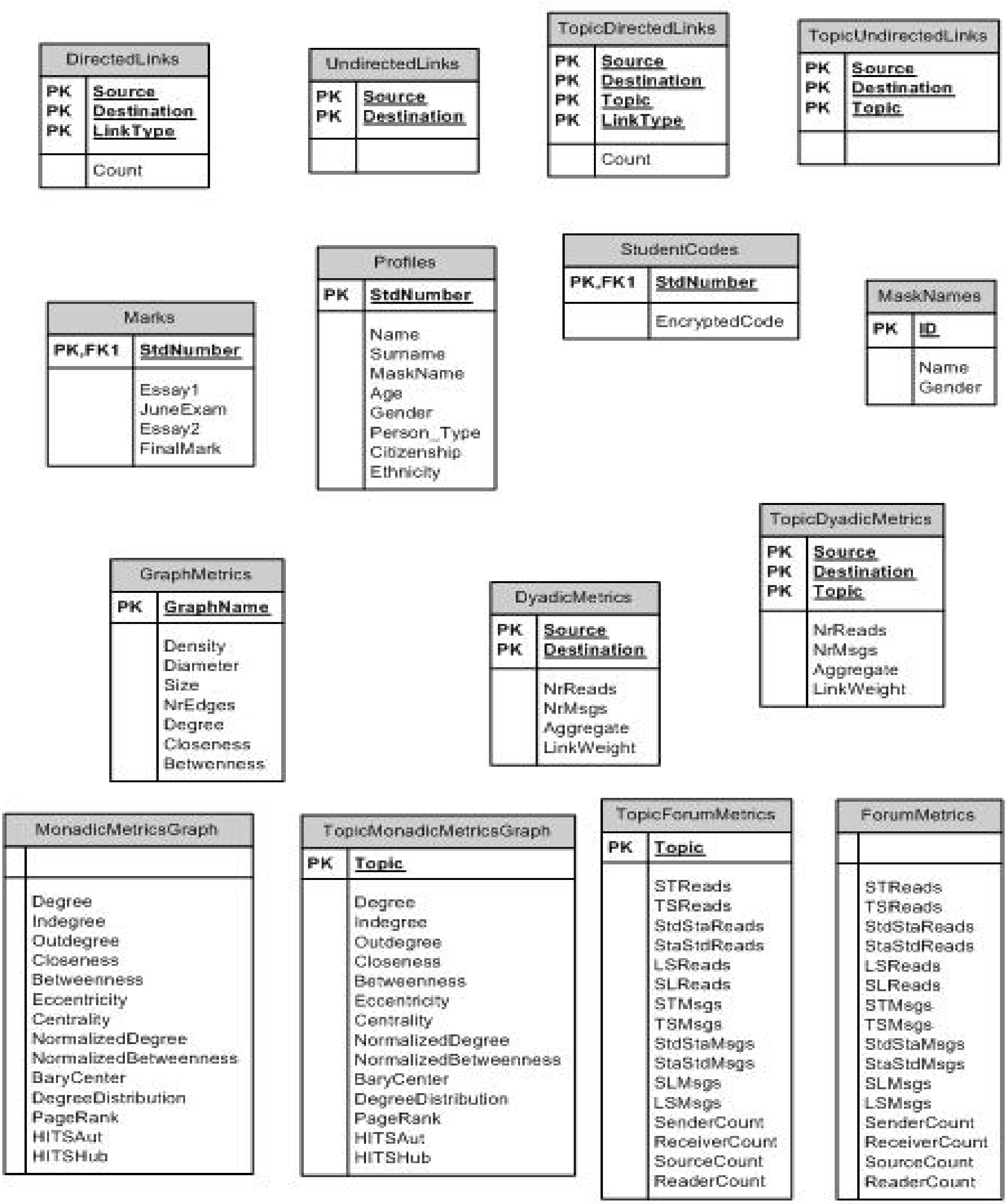
APPENDIX A

Social Network Subsystem Design



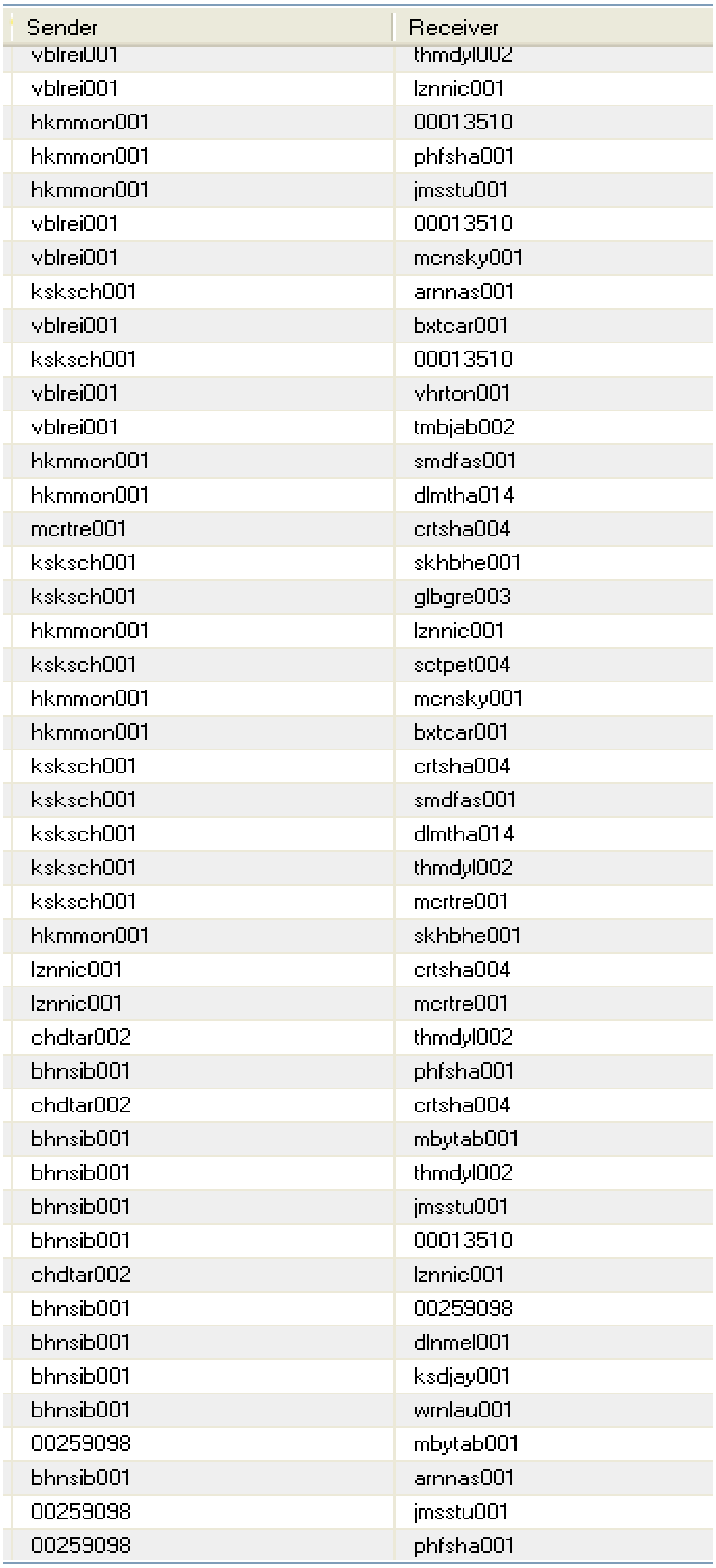
APPENDIX B

MySQL Entity Relationship (ER) Model



APPENDIX C

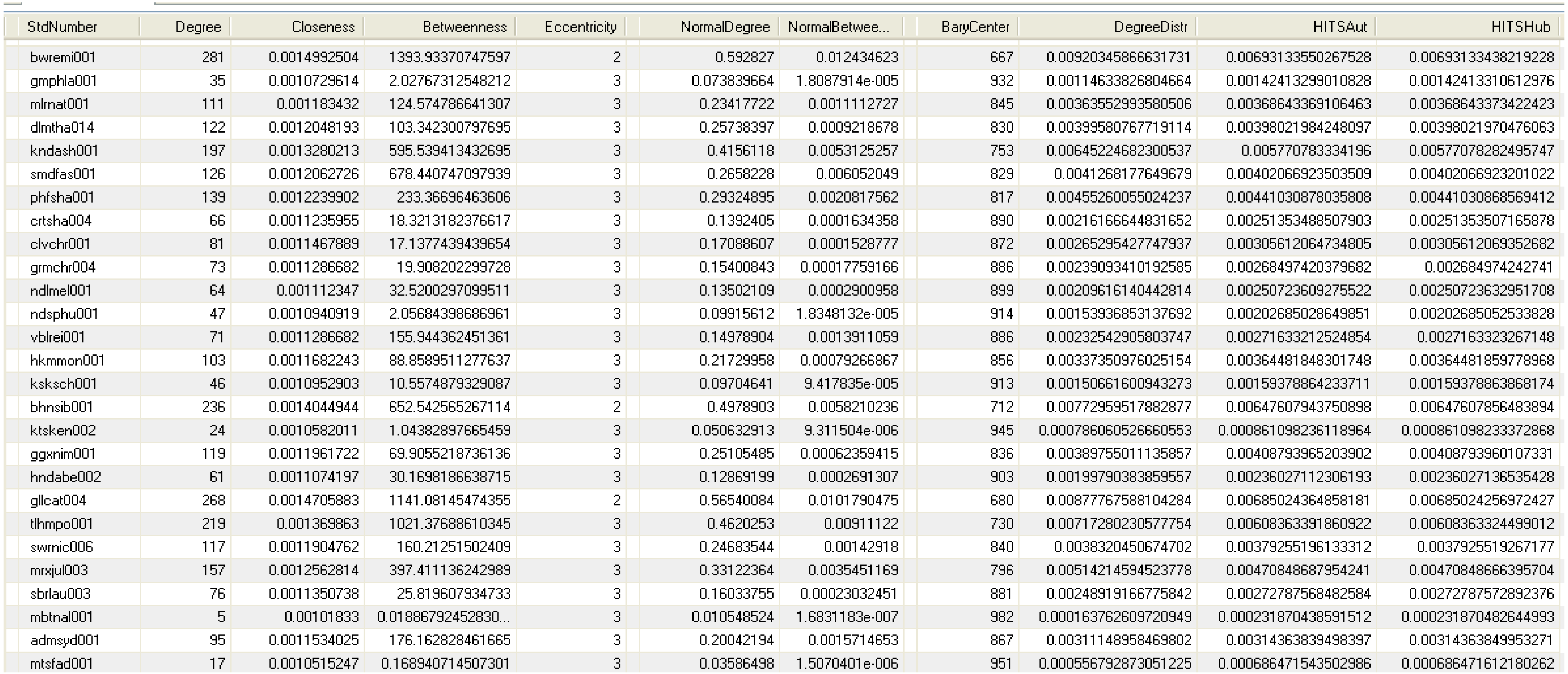
Social Network Database Representation



Sender	Receiver
vblrei001	thmdyl002
vblrei001	lznnic001
hkmmon001	00013510
hkmmon001	phfsha001
hkmmon001	jmsstu001
vblrei001	00013510
vblrei001	mcnsky001
ksksch001	arnnas001
vblrei001	bxtcar001
ksksch001	00013510
vblrei001	vhrton001
vblrei001	tmbjab002
hkmmon001	smdfas001
hkmmon001	dlmtha014
mcrtre001	crtsha004
ksksch001	skhbhe001
ksksch001	glbgre003
hkmmon001	lznnic001
ksksch001	sctpet004
hkmmon001	mcnsky001
hkmmon001	bxtcar001
ksksch001	crtsha004
ksksch001	smdfas001
ksksch001	dlmtha014
ksksch001	thmdyl002
ksksch001	mcrtre001
hkmmon001	skhbhe001
lznnic001	crtsha004
lznnic001	mcrtre001
chdtar002	thmdyl002
bhnsib001	phfsha001
chdtar002	crtsha004
bhnsib001	mbytab001
bhnsib001	thmdyl002
bhnsib001	jmsstu001
bhnsib001	00013510
chdtar002	lznnic001
bhnsib001	00259098
bhnsib001	dlnmel001
bhnsib001	ksdjay001
bhnsib001	wrnlau001
00259098	mbytab001
bhnsib001	arnnas001
00259098	jmsstu001
00259098	phfsha001

APPENDIX D

Social Network Metrics



StdNumber	Degree	Closeness	Betweenness	Eccentricity	NormalDegree	NormalBetwee...	BaryCenter	DegreeDistr	HITSAut	HITSHub
bwremi001	281	0.0014992504	1393.93370747597	2	0.592827	0.012434623	667	0.00920345866631731	0.00693133550267528	0.00693133438219228
gmphla001	35	0.0010729614	2.02767312548212	3	0.073839664	1.8087914e-005	932	0.00114633826804664	0.00142413299010828	0.00142413310612976
mlrnat001	111	0.001183432	124.574786641307	3	0.23417722	0.0011112727	845	0.00363552993580506	0.00368643369106463	0.00368643373422423
dlmtha014	122	0.0012048193	103.342300797695	3	0.25738397	0.0009218678	830	0.00399580767719114	0.00398021984248097	0.00398021970476063
kndash001	197	0.0013280213	595.539413432695	3	0.4156118	0.0053125257	753	0.00645224682300537	0.005770783334196	0.00577078282495747
smdfas001	126	0.0012062726	678.440747097939	3	0.2658228	0.006052049	829	0.0041268177649679	0.00402066923503509	0.00402066923201022
phfsha001	139	0.0012239902	233.36696463606	3	0.29324895	0.0020817562	817	0.00455260055024237	0.00441030878035808	0.00441030868569412
crtsha004	66	0.0011235955	18.3213182376617	3	0.1392405	0.0001634358	890	0.00216166644831652	0.00251353488507903	0.00251353507165878
clvchr001	81	0.0011467889	17.1377439439654	3	0.17088607	0.0001528777	872	0.00265295427747937	0.00305612064734805	0.00305612069352682
grmchr004	73	0.0011286682	19.908202299728	3	0.15400843	0.00017759166	886	0.00239093410192585	0.00268497420379682	0.002684974242741
ndlmel001	64	0.001112347	32.5200297099511	3	0.13502109	0.0002900958	899	0.00209616140442814	0.00250723609275522	0.00250723632951708
ndsphu001	47	0.0010940919	2.05684398686961	3	0.09915612	1.8348132e-005	914	0.00153936853137692	0.00202685028649851	0.00202685052533828
vblrei001	71	0.0011286682	155.944362451361	3	0.14978904	0.0013911059	886	0.00232542905803747	0.00271633212524854	0.0027163323267148
hkmmon001	103	0.0011682243	88.8589511277637	3	0.21729958	0.00079266867	856	0.00337350976025154	0.00364481848301748	0.00364481859778968
ksksch001	46	0.0010952903	10.5574879329087	3	0.09704641	9.417835e-005	913	0.00150661600943273	0.00159378864233711	0.00159378863868174
bhnsib001	236	0.0014044944	652.542565267114	2	0.4978903	0.0058210236	712	0.00772959517882877	0.00647607943750898	0.00647607856483894
ktsken002	24	0.0010582011	1.04382897665459	3	0.050632913	9.311504e-006	945	0.000786060526660553	0.000861098236118964	0.000861098233372868
ggxnim001	119	0.0011961722	69.9055218736136	3	0.25105485	0.00062359415	836	0.00389755011135857	0.00408793965203902	0.00408793960107331
hndabe002	61	0.0011074197	30.1698186638715	3	0.12869199	0.0002691307	903	0.00199790383859557	0.00236027112306193	0.00236027136535428
gllcat004	268	0.0014705883	1141.08145474355	2	0.56540084	0.0101790475	680	0.00877767588104284	0.00685024364858181	0.00685024256972427
tlhmpo001	219	0.001369863	1021.37688610345	3	0.4620253	0.00911122	730	0.00717280230577754	0.00608363391860922	0.00608363324499012
swrnic006	117	0.0011904762	160.21251502409	3	0.24683544	0.00142918	840	0.0038320450674702	0.00379255196133312	0.0037925519267177
mrxjul003	157	0.0012562814	397.411136242989	3	0.33122364	0.0035451169	796	0.00514214594523778	0.00470848687954241	0.00470848666395704
sbrlau003	76	0.0011350738	25.819607934733	3	0.16033755	0.00023032451	881	0.00248919166775842	0.00272787568482584	0.00272787572892376
mbtnal001	5	0.00101833	0.01886792452830...	3	0.010548524	1.6831183e-007	982	0.000163762609720949	0.000231870438591512	0.000231870482644993
admsyd001	95	0.0011534025	176.162828461665	3	0.20042194	0.0015714653	867	0.00311148958469802	0.00314363839498397	0.00314363849953271
mtsfad001	17	0.0010515247	0.168940714507301	3	0.03586498	1.5070401e-006	951	0.000556792873051225	0.000686471543502986	0.000686471612180262