

ABSTRACT

Title of Dissertation: LULLABI: KNOWLEDGE VISUALIZATION NETWORK

Clarence Heron Cupid, D.Eng. May 2019

Dissertation Chair: LeeRoy Bronner, Ph.D.

Industrial and Systems Engineering Department

Network science is an ever evolving field. As the research continues to evolve, new gaps are opened, and tools are created to help fill these gaps. There are a variety of network analysis tools available for research and they cover a wide range of use cases. However, there are three gaps to be addressed. First, there seems to be little discussion that network links can carry as much information (attributes) as nodes, and the important role that this information can play. Second, there is a need for multimedia data within networks, where images, video, sound, and documents can be used to visually enhance the display of information in a network. Third, network partitions do not address the idea of multiple classifications, direct use of continuous data to create clusters, or clusters composed entirely of links. A Knowledge Visualization Network (KVN) is presented as an innovative approach to filling this gap.

LULLABI: KNOWLEDGE VISUALIZATION NETWORK

By

Clarence Heron Cupid

A Dissertation Submitted in Partial Fulfillment

of the Requirements for the Degree

Doctor of Engineering

MORGAN STATE UNIVERSITY

May 2019

LULLABI: KNOWLEDGE VISUALIZATION NETWORK

By

Clarence Heron Cupid

has been approved

February 2019

DISSERTATION COMMITTEE APPROVAL:

_____, Chair

LeeRoy Bronner, Ph.D.

Guangming Chen, Ph.D.

Gregory Ramsey, Ph.D.

Gregory Wilkins, Ph.D.

Table of Contents

List of Tables	vii
List of Figures	vii
1 Chapter 1: Introduction.....	1
1.1 Networks.....	1
1.2 Network Science Overview	3
1.2.1 Network Visualization	5
1.2.2 Network Partitions	6
1.3 Problem Scope	7
1.4 Problem Description.....	7
1.4.1 Frist problem: Little emphasis on storage of attributes in links	7
1.4.2 Second Problem: No storage of multimedia data.....	8
1.4.3 Third Problem: Limited use of attributes for network partitions	9
1.5 Objectives.....	12
2 Chapter 2: Background & Literature Review	13
2.1 Network Analysis Tools	13
2.2 Tulip	14
2.2.1 Use Cases	16
2.2.2 Discussion.....	17
2.3 Graphviz	17
2.3.1 Uses Cases	18
2.3.2 Discussion.....	20
2.4 Gephi.....	20
2.4.1 Discussion.....	22
2.5 Cytoscape.....	23

2.5.1	Use Cases	25
2.5.2	Discussion.....	27
2.6	Pajek.....	27
2.6.1	Use Cases	29
2.6.2	Discussion.....	32
2.7	SocNetV.....	33
2.7.1	Use Cases	34
2.7.2	Discussion.....	35
2.8	Meerkat.....	35
2.8.1	Use Case	36
2.8.2	Discussion.....	37
2.9	Network Work Bench	38
2.9.1	Discussion.....	39
3	Chapter 3: Methodology	41
3.1	Overview	41
3.2	JAD Session.....	42
3.2.1	Problem Definition	43
3.2.2	Problem Description.....	43
3.1.3.	Conceptual Model	44
3.1.4.	Use Case Model.....	44
3.3	Analysis Model (OOAD).....	46
3.3.1	Class Diagram	47
3.3.2	Object Diagrams	48
3.3.3	Sequence Diagram.....	49
3.3.4	State Diagram	50
3.3.5	Activity Diagram	51

3.4	Design Model	52
3.5	Implementation Model	52
3.6	Test Model	52
3.7	System Evaluation	53
3.8	Proposed Application Modules & User Integration.....	53
3.9	Conceptual GUI	54
3.10	KVN Example Usage	56
3.11	Morgan Community Mile Example.....	56
3.11.1	Structure	56
3.11.2	Small Public	57
3.11.3	Wide Public	57
3.11.4	Researchers	58
3.11.5	KVN.....	58
3.12	Morgan State University Undergrad Senior Research Project.....	59
4	Chapter 4: Results	61
4.1	KVN Conceptual Design.....	61
4.1.1	Entity Component Model (ECM)	61
4.1.2	Attribute Component, Processing, and Visualization Model	62
4.1.3	Network Link Model	63
4.1.4	Network Database Model	64
4.1.5	Network Filter Model	65
4.2	KVN Use Case Model.....	66
4.3	KVN Class Model	68
4.4	KVN Application	69
4.5	Example Network.....	70
4.5.1	Link Attributes	71

4.5.2	Multimedia	72
4.5.3	Network Filter	73
4.6	KVN Use Cases.....	81
4.7	Deliverables.....	81
4.7.1	Conceptual Model	81
4.7.2	Logical Model	82
4.7.3	Physical Model	82
4.8	Future Work	82
5	References	84
	Appendix A: Morgan Community Mile.....	90
	Appendix B: MSU Undergrad Senior Project Research	97

List of Tables

Table 1: Family Relationships.....	5
Table 2: ABC School	10
Table 3: ABC School Network Partition Classifications	10
Table 4: ABC School Partition Criteria	11
Table 5: ABC School Relationship Classification	12
Table 6: Typical JAD Agenda (Shelly & Rosenblatt, 2012).	43
Table 7: MCM Use Case Contrast.....	57
Table 8: Use Case Scenario Activate Filter	67

List of Figures

Figure 1: High Level Systems Diagram (HLSD)	3
Figure 2: Network Graph.....	6
Figure 3: Tulip 5 application window.	15
Figure 4: Gephi application window.....	21
Figure 5: Gephi Drag & Drop Filters	23
Figure 6: Cytoscape application window.....	24
Figure 7: Pajek application window.	28
Figure 8: Simplified IVL DB Schema	31
Figure 9: SocNetV application window.	33
Figure 10: Meerkat application window.	36
Figure 11: NWB application window.....	38
Figure 12: Systems Development Life Cycle (SDLC; Tsang et al., 2005).....	42

Figure 13: Use case diagram create bus route (Shelly & Rosenblatt, 2012).	45
Figure 14: Use case scenario add new student (Shelly & Rosenblatt, 2012).	46
Figure 15: Class diagram customer order processing system (Shelly & Rosenblatt, 2012).	47
Figure 16: Dog object diagram (Shelly & Rosenblatt, 2012).	49
Figure 17: Sequence diagram (Shelly & Rosenblatt, 2012).	50
Figure 18: State transition diagram bank account (Shelly & Rosenblatt, 2012).	50
Figure 19: Activity diagram ATM cash withdrawal (Shelly & Rosenblatt, 2012).	51
Figure 20: KVN application integration	54
Figure 21: KVN Application GUI.....	55
Figure 22: KVN Application Example	55
Figure 23: CAP Project.....	60
Figure 24: Entity Component Model (ECM)	62
Figure 25: Attribute Component, Processing & Visualization	63
Figure 26: Network Link Model.....	64
Figure 27: Network Database Model	65
Figure 28: Network Filter Model	66
Figure 29: Network Filter Use Cases	67
Figure 30: KVN Class Diagram	68
Figure 31: KVN Entity Component Model Class Diagram	69
Figure 32: KVN Application Window	70
Figure 33: KVN Application Multiwindow	70
Figure 34: KVN network application with network	71
Figure 35: Attributes on links	72
Figure 36: Selected link with media	72

Figure 37: KVN media popup	73
Figure 38: KVN network filter application map	74
Figure 39: ABC School sex filter.....	74
Figure 40: ABC School sex filter network	75
Figure 41: ABC School teacher filter.....	75
Figure 42: ABC School teacher filter network	76
Figure 43: ABC School sex and teacher filter.....	77
Figure 44: ABC School sex and teacher filter network	77
Figure 45: ABC School sex and teacher with overlap	78
Figure 46: ABC School sex and teacher with overlap applied.....	78
Figure 47: ABC School dynamic criteria filter	79
Figure 48: ABC School dynamic criteria filter network.....	79
Figure 49: ABC School link cluster filter	80
Figure 50: ABC School link cluster filter network	81
Figure A-1: MCM as a network.....	90
Figure A-2: MCM to Staff Network Transitions	91
Figure A-3: MCM Staff.....	92
Figure A-4: MCM Board.....	92
Figure A-5: MCM Partners.....	93
Figure A-6: MCM Projects	94
Figure A-7: MCM Community Network.....	95
Figure A-8: MCM Community Selection	95
Figure A-9: MCM Community Income vs Child Poverty	96

Figure B-1: MSU Student Project Media Database..... 98

Figure B-2: MSU Student Project Excel Import w Database 98

Figure B-3: MSU Student Project Network..... 99

1 Chapter 1: Introduction

1.1 Networks

There are fundamental laws that govern how the universe works. These laws are often modeled using mathematical expressions consisting of variables and operations that describe how those variables interact. There are laws that describe the functions of galaxies down to the smallest atoms. When a high level view is taken of these systems and the models that describe them, it becomes easy to see models of network systems.

Networks are universal because they can be used to describe objects and the interactions between them. From looking at the World Wide Web (WWW), genetic networks, and social systems, even though there are differences in the nature of the entities and their interactions, these networks are complex systems governed by fundamental laws (Barabási, 2012). It does not matter whether the objects are atoms and how they chemically react to each other, brain neurons and how they fire signals to each other, an ecosystem consisting of animals and plants relating how they affect each other, or the universe consisting of stars and how they affect each other. Networks are universal and can be used to represent any phenomena and its interactions in the observable universe.

Network research is an important field because of its universality and its integration into improving humanity. According to Barabasi, the economic giants of our era are no longer carmakers and oil producers, but companies like Cisco, Google, Microsoft, Facebook, Apple, and Twitter, who build, manage, and fuel social networks that are affecting our everyday lives (Barabási, 2012). No single theory can tackle the complexity of network research because the problems are too diverse. However, when the problems are being attacked by several different

companies and several different fields, it is much more likely that network complexity can be addressed.

The National Research Council (NRC) defines network science as “organized knowledge of networks based on the study using the scientific method” (National Research Council [U.S.], 2005). The purpose of the definition was to distinguish it from various technologies that use it. Lewis argues that the definition is not mature enough to be separated from its technological roots which seems to be rooted in the Internet (Lewis, 2009). However, it may be best to separate the definition because different subfields have different working definitions of network science. For network science to mature it must be separated from its technological roots, to allow different subfields to bring different phenomena to the table that may be able to apply to other fields. It is naïve to think that the entirety of network science can be developed using just the Internet alone when networks can be seen everywhere. An important part of the scientific method is the formulation of a hypothesis, which is then tested to determine whether it is true or false. Likewise, it may be best to separate network science from its roots and allow other fields to formulate their own hypothesis and test it to see if there is any interesting phenomena. It can be argued that the separation can cause network science to become fragmented. However, it can also be argued that if the separation of network science from its roots does not occur, it can cause the science to stagnate. For the science to grow, different disciplines must use it in their own way. Whatever is learnt in one discipline could have the potential of being applied to another discipline.

This is shown through network analysis that cuts the boundaries of traditional disciplines consisting of sociologists, anthropologists, mathematicians, economists, political scientists, psychologists, communication scientists, statisticians, ethologists, epidemiologists, computer scientists, market specialists from business schools, and recently, physicists (Freeman,

2004). Network science is complex, and the ever increasing size of networks pose many challenges. Thus, it is important for different disciplines to bring their own version of network science to the table. This increases the body of knowledge with the possibility of tackling network complexity by allowing information to be used across disciplines.

1.2 Network Science Overview

Figure 1 is a High Level Systems Diagram (HLSD) that shows networks as a system and the problem scope of this research. The figure at a minimum shows the important parts of a network that is required for proper visualization. It shows networks, nodes, links, attributes, network visualization, network partitions, and the relationships between them. This figure will be further elaborated on in the following sub sections.

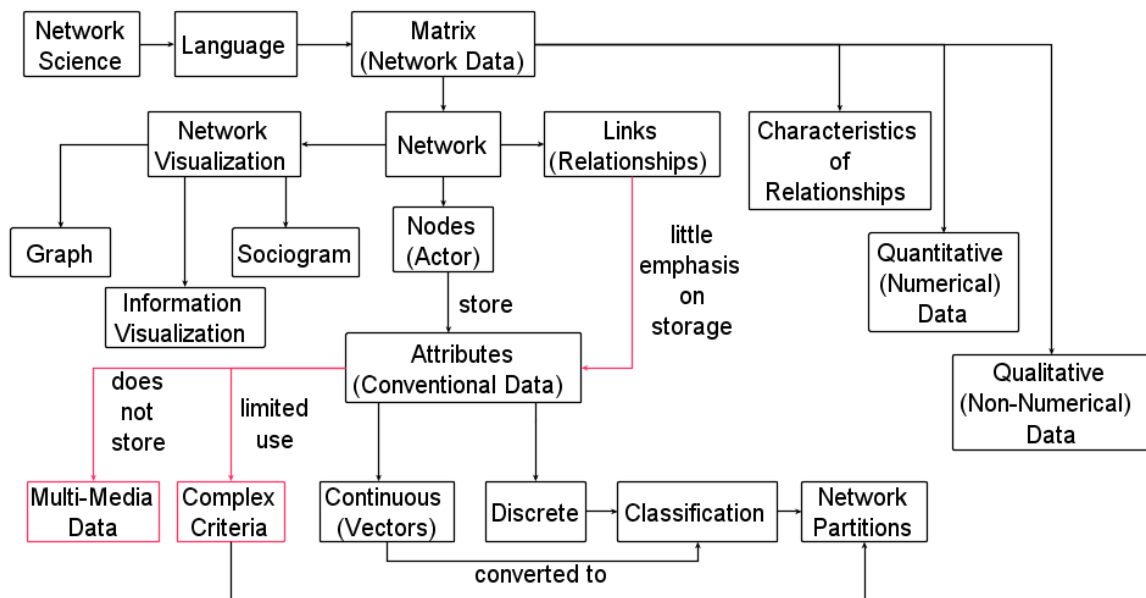


Figure 1: High Level Systems Diagram (HLSD)

Networks have a mathematical definition given by equation (1). $G(t)$ represents the entire network as a function of time consisting of nodes $N(t)$, links $L(t)$, mapping function $f(t)$, and how the network behaves over time. Networks also have a language that allows one to model a problem to bring greater insight and understanding (Hanneman & Riddle, 2005). This

language consists of matrices which are used to model networks as shown in Figure 1. A binary matrix as given in equation (2), models network data by showing the nodes and relationships in a network. Matrices can also be used to characterize relationships using quantitative and qualitative data as shown in Figure 1. A weight matrix as given in equation (3), models network data by showing the strengths of relationships. This can be considered quantitative data. A categorical matrix as given by equation (4), can characterize relationships where the categories can be transformed using a table (Table 1). This can be considered qualitative data. Nodes store attributes, also known as conventional data. These attributes help to describe the node. Attributes can be discrete or continuous and allows for classifying networks using network partitions as shown in Figure 1.

$$G(t) = \{N(t), L(t), f(t) : J(t)\} \quad (1)$$

Where, t = time, simulated or real

N = nodes, also known as vertices or “actors”

L = links, also known as edges

$f = N \times N$ = mapping function that connects node pairs

J = algorithm for describing behavior of nodes and links versus time

$$G = \begin{pmatrix} - & A & B & C \\ A & 1 & 1 & 0 \\ B & 0 & 0 & 1 \\ C & 0 & 0 & 0 \end{pmatrix} \quad (2)$$

$$W = \begin{pmatrix} - & A & B & C \\ A & 0.5 & 0.4 & 0.3 \\ B & 0 & 0 & 0.2 \\ C & 0 & 0 & 0 \end{pmatrix} \quad (3)$$

$$C = \begin{pmatrix} - & Ally & Bill & Cara \\ Ally & 0 & A & S \\ Bill & N & 0 & N \\ Cara & S & A & 0 \end{pmatrix} \quad (4)$$

Table 1: Family Relationships

Key	Value
A	Aunt =1
S	Sister = 2
N	Nephew = 3

1.2.1 Network Visualization

Network visualization is the process of visualizing networks using graphs (Figure 2), sociograms or information visualization as shown in Figure 1. Graphs are just simple drawings of vertices and their edges. Sociograms are graphs that represent structural ties within groups (Nooy, Mrvar, & Batagelj, 2005). Information visualization are novel ways of presenting network information visually to make it easily understood.

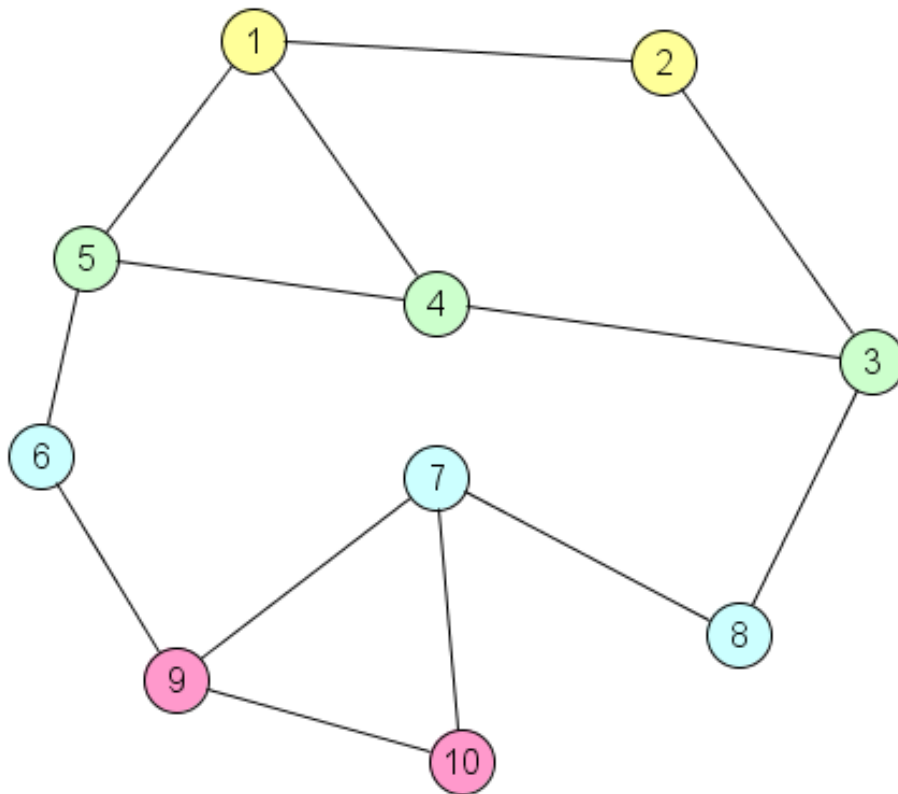


Figure 2: Network Graph

1.2.2 Network Partitions

Network partitions (see different colors in Figure 2) is a form of information visualization that allows the extraction of information to visually enhance large networks. Network partitions reduce networks into a local view, a global view or a contextual view. Local views are the extraction of a subnetwork from a network, global views have all vertices of a class shrunk to one new vertex, and contextual views shrinks all classes in the network except for the one of interest (Nooy et al., 2005). Network partitions are the classification or clustering of vertices, where each vertex belongs to one class or cluster (Nooy et al., 2005). The classes are determined from a discrete attribute of the vertex (ex: Sex, Country, etc.), or from continuous attributes (ex: Age, Time, etc.) that have been converted into some class like structure as shown in Figure 1.

1.3 Problem Scope

Figure 1 outlines the problem scope using red dotted lines. The problem scope is little emphasis on link storing attributes, no attributes that can store multimedia, and limited use of criteria to partition networks. These individual problems will be elaborated on in the problem description.

1.4 Problem Description

Network science involves the analysis of complex networks, where algorithms are used to find patterns to solve real world problems. Various network analysis tools have been developed over the years to aid in analysis by providing unique features. With these tools there is usually heavy emphasis on nodes and their attributes. However, there is little emphasis on the idea that links can carry as much information (attributes) as nodes (Hanneman & Riddle, 2005), and the important role that this information can play. Attributes in a network are either quantitative (numerical/continuous) or qualitative (non numerical/discrete). However, there is a need for multimedia data within networks, where images, video, sound, and documents can be used to visually enhance information in a network. Network partitioning can visually enhance a network; it is the clustering of vertices using a single classification (Nooy et al., 2005). However, network partitions do not address the idea of vertices having multiple classifications, dynamic use of attributes to create clusters, or clusters composed entirely of links.

1.4.1 Frist problem: Little emphasis on storage of attributes in links

There is little emphasis on the use of conventional data for links in network science. To explain why applying conventional data to links makes sense, the Universal Modeling Language (UML) Association class provides the perfect reason. There are times when a simple relationship between two nodes is not enough to fully describe their relationship. This is similar to UML where an association class is used to describe the relationships between two other classes.

Tsang, Lau, and Leung (2005) asserts that sometimes it is necessary to, “describe an association by including additional attributes which do not naturally belong to the objects involved” (Tsang, Lau, & Leung, 2005, p. 28). This happens when the relationship is complex enough to become a class, containing its own attributes, and operations. Tsang et al. (2005) further asserts that when relationships become complex, it is beneficial to contain the information in a separate class to maximize module cohesion.

With respect to networks, links associate two nodes to each other. When the relationship is complex enough such that it needs to be further described, network data could potentially be of no use. However, by allowing links to store conventional data, it allows one to fully describe the relationship between two nodes using numerical, non numerical, and multimedia data.

1.4.2 Second Problem: No storage of multimedia data

There is a need for multimedia data within networks, where images, video, sound, and documents can be used to visually enhance information in a network. Attributes that are quantitative or qualitative are used to describe nodes using textual data, and even links per this research. Multimedia data goes a step beyond and allows for the enhancement of data using images, video, sound, and documents.

For example, suppose a scientist wants to track a super family of meerkats consisting of about 50 or more members. Conventional data would be needed to properly describe each individual family member. Network data would be needed to show the relationships between family members. Multimedia data adds a twist by allowing the storing of images of each family member, sounds that they make, videos of daily activities, and documents written about the research.

1.4.3 Third Problem: Limited use of attributes for network partitions

Traditionally network science uses vertex classification to create network partitions; it does not allow the use of multiple classifications, dynamic filters that use criteria based on attributes, or the clustering of links as partitions. Network partitioning is a classification or clustering of vertices, such that each vertex is assigned to exactly one class or cluster (Nooy et al., 2005). Partitions store discrete properties, which consists of a limited number of classes. For example, sex can be a discrete property with 2 classes which are male and female. Partitions can also store structural properties, where it specifies a vertex position (x,y coordinates) in a network, or an attribute of the vertex that is independent of the network that uniquely describes the vertex. Vectors store continuous properties which take any value in a defined range, which are based around real numbers. For example, the age of students in a classroom can be considered a continuous property. Nooy and colleagues imply that vectors cannot be used to group vertices into classes, so they cannot be used to produce network partitions (Nooy et al., 2005). When using Pajek (Batagelj & Mrvar, 2018), one must first convert the vectors into classes using a frequency distribution, before they can be used to produce network partitions. It would be beneficial if the procedure was automated.

1.4.3.1 Multiple Classifications

Network partitions do not allow the use of multiple classifications, as there are cases where vertices do fall into multiple classes. For example, Table 2 shows students, their sex, and their teachers. From looking at the table, network partitions can be created based on sex or teacher. These 2 classifications would produce a total of 4 partitions with 2 in each network. However, if the two classifications are combined, it is possible to create 4 more partitions (see Table 3).

Table 2: ABC School

Student	Sex (Male/Female)	Teacher (Apple/Peach)
Ally	F	Ms. Peach
Bart	M	Ms. Peach
Carla	F	Ms. Apple
David	M	Ms. Apple
Erica	F	Ms. Peach
Fernard	M	Ms. Apple

Table 3: ABC School Network Partition Classifications

Classification	Category	Color
Sex	Male	Blue
	Female	Pink
Teacher	Ms. Peach	Orange
	Ms. Apple	Red
Sex & Teacher	Male & Ms. Peach	Blue & Orange
	Female & Ms. Peach	Pink & Orange
	Male & Ms. Apple	Blue & Red
	Female & Ms. Apple	Pink & Red

1.4.3.2 Dynamic partitioning using criteria based on attributes

Network partitions are good for showing classes of vertices. However, they do not hold up well when there are several hundred vertices with multiple attributes (some of which are vectors). In this situation, a criteria would best fit that dynamically uses the attributes contained

in the network to partition it. For example, if Table 2 was expanded to 30 students and also list their ages from the 3–6, then Table 4 would list the conditions used to partition the school. It list several criteria that uses sex, teacher, and age to produce network partitions with a color code.

Table 4: ABC School Partition Criteria

Criteria	Color
Sex = F & Teacher = Apple & age > 5	Blue
Sex = F & Teacher = Apple & age <= 5	Brown
Sex = M & Teacher = Apple & age > 5	Red
Sex = M & Teacher = Apple & age <= 5	Green
Sex = F & Teacher = Peach & age > 5	Pink
Sex = F & Teacher = Peach & age <= 5	Purple
Sex = M & Teacher = Peach & age > 5	Orange
Sex = M & Teacher = Peach & age <= 5	Yellow

1.4.3.3 Link clusters or partitions

Network partitioning is the clustering of vertices; there are uses cases where the clustering of links may be preferred. For example, if relationships among the students had a quality consisting of friendly, hostile, or assumed friend, then the relationship quality can be used to classify (see Table 5) the network. Looking at a link cluster of the network would give an idea of the type of environment the students create. If the link clusters contained mostly blue links, then it can be disseminated that the environment is friendly. If the link clusters contained mostly red links, then it can be disseminated that the environment is hostile. If the link clusters contained mostly green links, then it can be disseminated that the environment is unstable, and can sway either way depending on the mood of the students.

Table 5: ABC School Relationship Classification

Classification	Color
Friendly	Blue
Hostile	Red
Assumed Friend	Green

1.5 Objectives

From discussing the problems that exist in current network science, there are objectives that this research would like to meet. The objective of this research is to develop a web application for Knowledge Visualization Networks (KVN) that supports the following:

- Creation and storage of networks to a database.
- Same capabilities between nodes and links.
- Viewing of multimedia data within nodes and links.
- Dynamic filtering of networks that use criteria based on attributes.

There may be questions concerning why a web application was chosen rather than a desktop application. The reason for this is because the web runs on all major desktop operating systems, with a large library of frameworks to make web application development easy. This would allow one to focus on developing the application rather than the development of libraries to support application development.

2 Chapter 2: Background & Literature Review

2.1 Network Analysis Tools

There are a wide variety of network tools that have been developed over the last 20-30 years. They all provide similar features and have been used for various purposes. Features often included by these tools are:

- import/export of various graph formats,
- various layout functions,
- tabulated nodes and links with editing,
- view manipulation,
- network statistics,
- application extension using plugins,
- network filters, and
- dynamic networks.

With such a wide variety of tools available, it is difficult to determine which one best fits a particular problem. That is unless the tool is specifically tailored to a problem. To derive a measure of tools, Shah & Mehta (2017) used a comparative analysis that categorized social network analysis tools based on their basic functionalities, network types, graph layouts, import formats, and graph attributes (Shah & Mehta, 2017). In their comparative analysis they compared Gephi, SocNetV, yEd, and Cytoscape. Of the 4 they concluded that Gephi was the strongest, but more or less they were all similar in their capabilities.

Desale lists the top 30 social network analysis software (Desale, 2015). The list is mostly comprised of cross platform desktop applications. Some of the applications are web based. Not all network tools are GUI (graphical user interface) based. Some of the tools have grown to

become very popular with many years of software engineering and very carefully crafted architectures. Others seem to have stagnated to some degree in which there has not been a new release in several years. It would make sense that the ones that have been carefully engineered to address diverse problems, an architecture designed for extensibility, and an eye toward the future would stand very strong. This certainly seems to be case with Tulip, and there are others that fall in this list. There have also been publications written about the tools by the authors, researchers seeking to perform comparative analysis, and by other professionals who used the tools for network analysis purposes. It is unclear why some of these professionals chose the tools they used. With such a wide variety of tools, there are a vast amount of uses cases that are covered. However, there is still a gap that needs to be filled as outlined in the problem description (see section 1.4). To define this gap, some of the more popular tools will be evaluated. Their features will be discussed along with case studies of how they were used to show that a gap exists.

2.2 Tulip

Tulip is based on 15 years of research and development (Auber et al., 2017). It has a core architecture that allows the exploration of very large graphs, an OpenGL rendering engine, and python integration for application extensions (Auber et al., 2012). Tulip enables the development of algorithms, visual encodings, interaction techniques, data models, and domain specific visualizations. It was primarily designed for scalability where very large graphs could be handled and visualized (Auber et al., 2012). The authors designed a low level data structure to support the creation of nested/overlapped subgraphs, with a property heritage mechanism at the heart of the system that provides coherence and optimal space usage. This provides efficient definition and navigation of graph hierarchies and often provides answers to aid analysts. Tulip provides the storage of an unbounded number of properties or attributes for graph elements

and includes an exhaustive list of network analysis metrics that can be used for visualization purposes.

Tulip uses an OpenGL rendering engine that takes advantage of the GPU (graphics processing unit) for large scale visualizations. The authors implemented a multiscale graph drawing technique that allows the exploration of large networks that do not fit into main memory. The engine allows multilayer rendering for proper scene composition and 3 dimensional information. Various graph drawing computations are done in real time on the GPU, that allows the rendering of complex curves without storing any precomputed geometry. Tulip also includes a python IDE (integrated development environment) for the scripting of new functionality. Figure 3 gives a simple view of Tulip's application window. It shows a vertical toolbar (far left) with the python IDE icon, algorithms (left), table of elements within network (center), and network view (right).

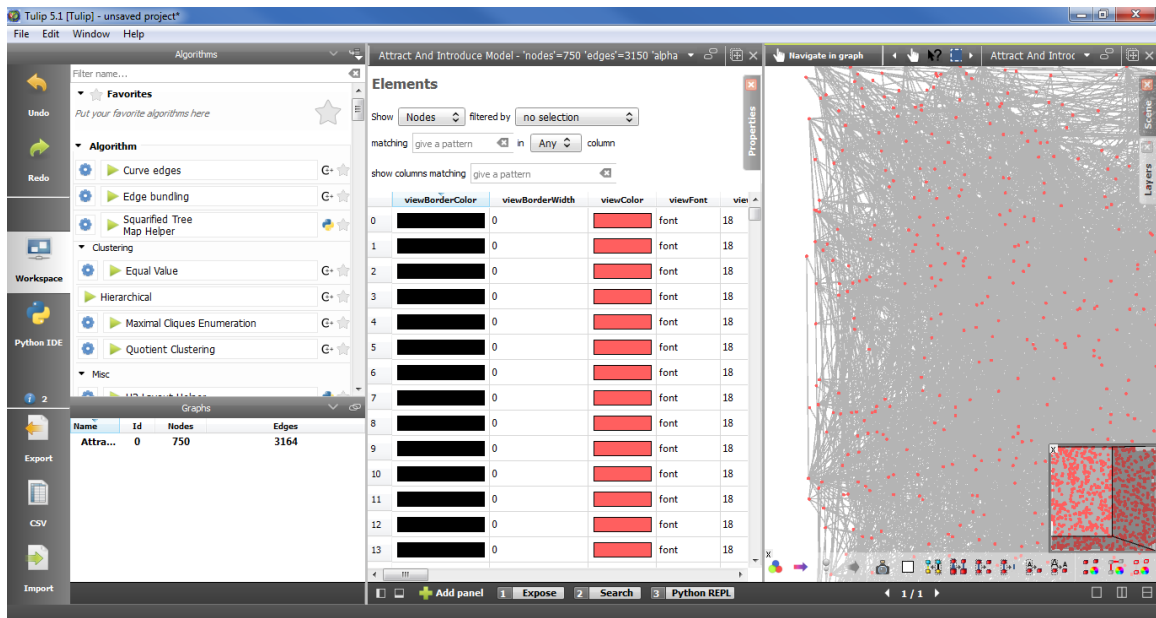


Figure 3: Tulip 5 application window.

Tulip supports a wide variety of features that include:

- plugins to extend program functionality,

- Python IDE integration,
- import from GML (Graph Modeling Language), GraphML, GEXF (Graph Exchange XML Format), Pajek, CSV (comma separated values), and other formats,
- exports to CSV, GML, SVG (scalable vector graphics), and other formats,
- creation of subgraphs from selection,
- wide variety of layout algorithms,
- network statistics,
- viewing/editing/searching of network data in table,
- navigator for large networks, and
- GPU rendering engine.

2.2.1 Use Cases

Tulip has been used for navigation of protein interaction networks, production of automated drawing for secondary RNA (ribonucleic acid) structures, visualization of software reverse engineering graphs, social networks, air passenger traffic, and various other use cases.

The following sections will give an overview of some use cases.

2.2.1.1 *Systryp*

The tsetse fly is a parasite that causes sleeping sickness. Systryp is a perspective constructed using Tulip to help biologist understand the parasite's metabolism (Auber et al., 2012). Biologists used Tulip to implement biology specific algorithms, with improvements through user feedback to devise a proper solution.

2.2.1.2 *Grouse, GrouseFlocks, and TugGraph*

Tulip was also used to research new techniques for efficient rendering of very large graphs, especially in cases where readability is a concern. The Tulip team researched and implemented the Grouse, GrouseFlocks, and TugGraph algorithms (Auber et al., 2012). The

Grouse technique takes a large graph and hierarchy as input to draw parts of the graph on demand. Appropriate drawing algorithms are used based on the topological structure of the graph. GrouseFlocks constructs graph hierarchy based on attribute data and progressively draws them. It uses a technique involving metanodes that could be drawn on demand with Grouse. However, some metanodes are too large to draw on demand. TugGraph was constructed for this side case.

2.2.2 Discussion

Tulip is a powerful and flexible information visualization application. It supports a flexible and robust graph architecture with a GPU supported rendering engine. It has been used for some interesting research, but it does leave more to be desired. For one, Tulip's GUI does not feel user friendly. Its network view looks very cluttered and with a set of tools and no explanation or tooltip for what they do. Navigating within the network view is finicky and needs some improvements. The selection of nodes and links along with the application of data leaves one confounded. Tulip does support filtering of nodes and links within its networks. It looks as if only graphical attributes are supported, but there is not any support for general purpose attributes. If it does support general purpose attributes, there is no easy way of adding them. The output of the filter is not a visual network but rather a list of nodes and links. The list can be used to make selections and possibly other things. However, this is vastly different from the network filtering being addressed in this research. Tulip does not support multiple classifications or dynamic attributes. It is also unclear whether link clusters are supported. Based on this information, it is safe to say that Tulip does not fill the gap being addressed by this research.

2.3 Graphviz

Graphviz (*Graphviz: Graph Visualization Software*, 2014) is a visualization tool with applications in software engineering, networking, databases, knowledge representation, and

bioinformation (Ellson, Gansner, Koutsofios, North, & Woodhull, 2004). It was written in the early 90s and is mostly implemented in the C programming language. The tool can be used via its C library interface, stream based command line tools, and GUI and web browsers. The stream based model is a good fit for applications that need external graph visualization services with a graphical or web interface. It also supports the emission of graph models in the dot language (graph description language) to be loaded in a Graphviz viewer (“The DOT Language,” 2019), or the generation of images in PDF (Portable Document Format) or SVG.

2.3.1 Uses Cases

Graphviz has been used across various disciplines. These include software, biology, the internet, telecommunication services, and even in the development of layout algorithms. These use cases will be discussed to show how Graphviz played a role.

2.3.1.1 *Software*

Graphviz has been successfully used in software engineering, bioinformatics, internet, web structures, and dynamic distribution services. Software engineering gets complicated with large scale software. Software visualization is one way to attack this problem. The industry standard is to use Object Oriented Analysis and Design (OOAD) along with UML for software visualization. Two notable systems that relied on Graphviz are Doxygen (Heesch, 2018) and Bugzilla (“Home :: Bugzilla :: bugzilla.org,” 2019). Doxygen is a software documentation tool and Bugzilla is a bug tracking tool used by the Mozilla Foundation.

There are also other tools that have been used to support software engineering. One earlier tool is Improvise, which is a “system for modeling and analyzing software processes” (E. Gansner & North, 1997). It is specifically stated as an interactive multimedia process visualization tool. It supported the annotation of nodes and links, using text and multimedia. So the adding of media to support information in networks is not new. However, UML tools are

designed for a specific purpose, which is the architecture and analysis of software components and processes. The industry standard today is Enterprise Architect (EA) which is based on UML 2.0 and covers a wide variety of use cases (*Enterprise Architect*, 2015).

2.3.1.2 *Biology*

Biological structures form natural networks that can be visualized using graphs. PubGene (“PubGene: Connect–Discover–Share,” 2019) is one example that uses Graphviz as a web visualization service. It is a biological database application that describes the co-citation of mouse, rat, and human gene symbols in over 10 million articles published on PubMed (pubmeddev, 2019). The system uses interactive queries to allow the exploration of neighborhoods around genes by standard names. The Protein Interaction Extraction System (PIES) is another bioinformation application using Graphviz (Wong, 2001). It allows a user to explore biomedical abstracts from online sources, extraction of relevant information from the text, and the manipulation of interaction pathways.

2.3.1.3 *Internet*

The World Wide Web Consortium (W3C) defines the semantic web in XML format. The Resource Description Framework (RDF) describes web site contents and naturally give rise to graphs. IsaViz and FRODO RDFSviz are translators that can map RDF into Graphviz diagrams. The Apache2Dot translator is another notable use of Graphviz. It is used for the viewing of links followed by a client within a website (Darxus, 2005).

2.3.1.4 *Telecommunication Services*

Distributed Feature Composition (DFC) is an architecture for specifying the structure and composition of modular communication service features. It solves many of the difficulties in specifying telecommunication services. Graphviz was used as a platform to monitor and validate service protocols and feature setups in real time.

2.3.1.5 Stress Majorization Layout Algorithm

GraphViz was used to implement the drawing of graphs using stress majorization. Gansner, Koren, and North (2005) argue that their technique offers advantages over processes like Newton-Raphson or gradient descent; they guarantee “monotonic decrease of the energy value, improved robustness against local minima, and shorter running times” (E. R. Gansner, Koren, & North, 2005). The authors compared their method against the method used by Kamada and Kawai (Kamada & Kawai, 1989). Based on their findings, the authors concluded that their method had better layout quality of networks, better convergence, and a faster running time. Thus, they believed that their method should replace the Kamada and Kawai technique.

2.3.2 Discussion

Graphviz has been used in software engineering, biology, the internet, and telecommunications. It has even been used to develop an improved layout algorithm using stress majorization. Graphviz alone is an interface based on a stream model. For some researchers this is sufficient. However, for researches looking for a GUI based network tool, the one provided by Graphviz does not feel sufficient. Currently, Graphviz may be behind the times in contrast to other applications that provide feature rich GUIs. Based on this analysis, Graphviz does not meet any of the objectives of this research as it does not provide a sufficient platform to be properly evaluated. Thus, it does not fill the gap that this research is addressing.

2.4 Gephi

Gephi (*Gephi: The Open Graph Viz Platform*, 2017) is a network exploration and manipulation software, built using a plugin architecture for expandability (Bastian, Heymann, & Jacomy, 2009). It was built around a core set of requirements defined by the authors; high quality layout algorithms, data filtering, clustering, statistics, and annotation, all wrapped with in a flexible, scalable, and user friendly application. Gephi uses a 3D render engine that leverages

the power of the GPU and takes advantage of CPUs with multiple cores. It allows for personalized node designs, where nodes can be a photo rather than a shape, with labels from any attribute associated with the nodes. Gephi also allows for data filtering to filter nodes and edges using a criteria. Filtering is chained where the output of one filter becomes the input to another filter. One of the most interesting feature of Gephi, is its support for dynamic networks that change over time. A user can control the network time using a slider, to observe what the network looked like at some specific time. Figure 4 gives a view of Gephi's application window. It shows the appearance and layout options (left), network view (center), and filters with various other options (right).

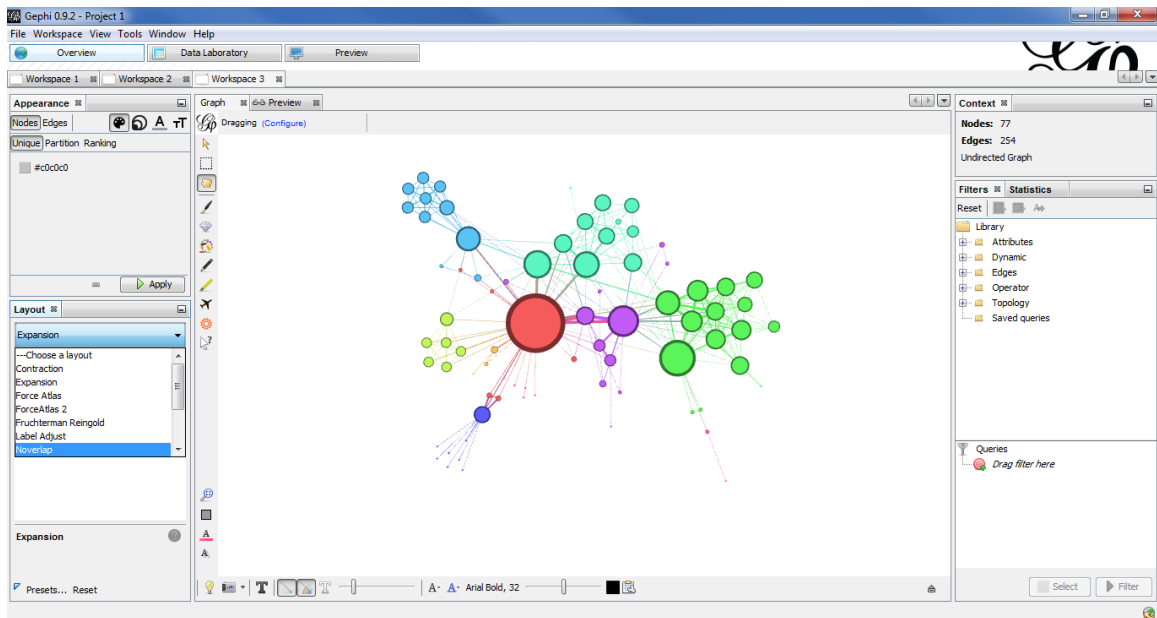


Figure 4: Gephi application window.

Gephi supports a wide variety of features that include:

- plugins that extend functionality,
- import networks from CSV,
- import networks from SQL (structured query language) databases (MySQL, SQLite, PostgreSQL, SQL Server),

- export networks to PDF, PNG (portable network graphics), SVG, GEXF, GML, GraphML, CSV, and other formats,
- a table view that allows editing of network data (nodes, links),
- readily available drag and drop filters that can have their parameters configured,
- export of filtered data to a column or network,
- network statistics,
- dynamic networks with a timeline that can be played back over time,
- GPU rendering engine.

2.4.1 Discussion

Gephi addresses two of the three problem definitions in this research. First, Gephi does place a greater emphasis on the kind of data stored in links (see section 1.4.1). There is no difference between the data that nodes and links can store. Second, Gephi supports an advanced way of network filtering (see section 1.4.3) using attributes and even network statistics. However, there are some minor differences in contrast to KVN. Gephi uses drag and drop filters (see Figure 5) to create node and link clusters. However, links clusters still display their connected nodes. Its filtering mechanism supports attributes, network statistics, and Boolean operations. Filtering in KVN plans on the use of criteria based on attributes and Boolean operations to support multiple classifications, continuous vectors, and links clusters. Gephi however does not support multimedia data in its networks. So while Gephi fills some of the gaps, it does not fill all the gaps being addressed by this research.

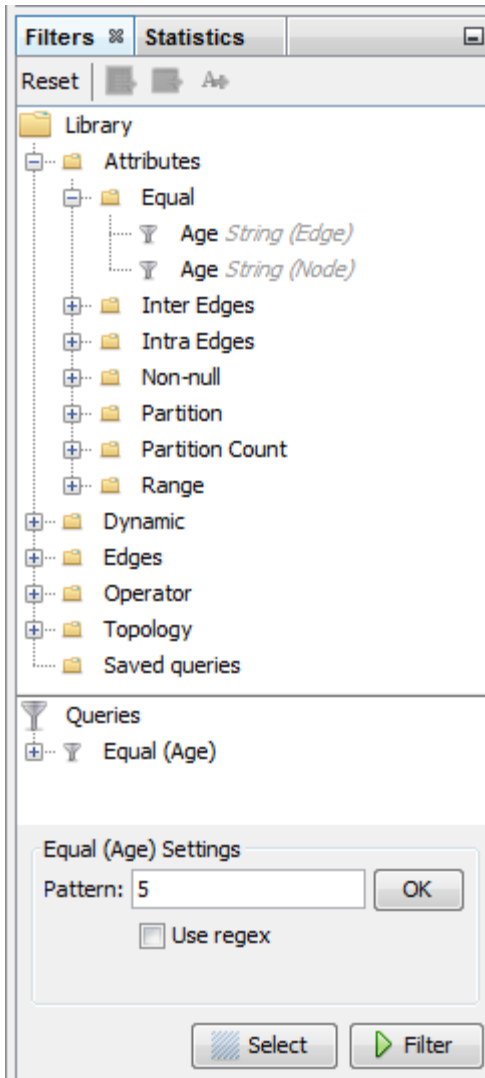


Figure 5: Gephi Drag & Drop Filters

2.5 Cytoscape

Cytoscape is an open source software platform for the visualization of molecular interaction networks and biological pathways, along with the integration of annotations, gene expression profiles, and other state data into the networks (Shannon et al., 2003). Originally designed for biological research, it is now a general platform for complex network analysis and visualization. Its core distribution provides a basic set of features for data integration, analysis, visualization, and an “Apps” architecture to extend its capabilities using the Cytoscape App Store. The apps architecture allows for the application of new functionality across its entire

domain. Apps are developed using Cytoscape's open API (application programming interface) based on Java. Figure 6 gives a view of Cytoscape showing its toolbar (top), control panel (left), network view (right), navigator (far right), and table panel (bottom).

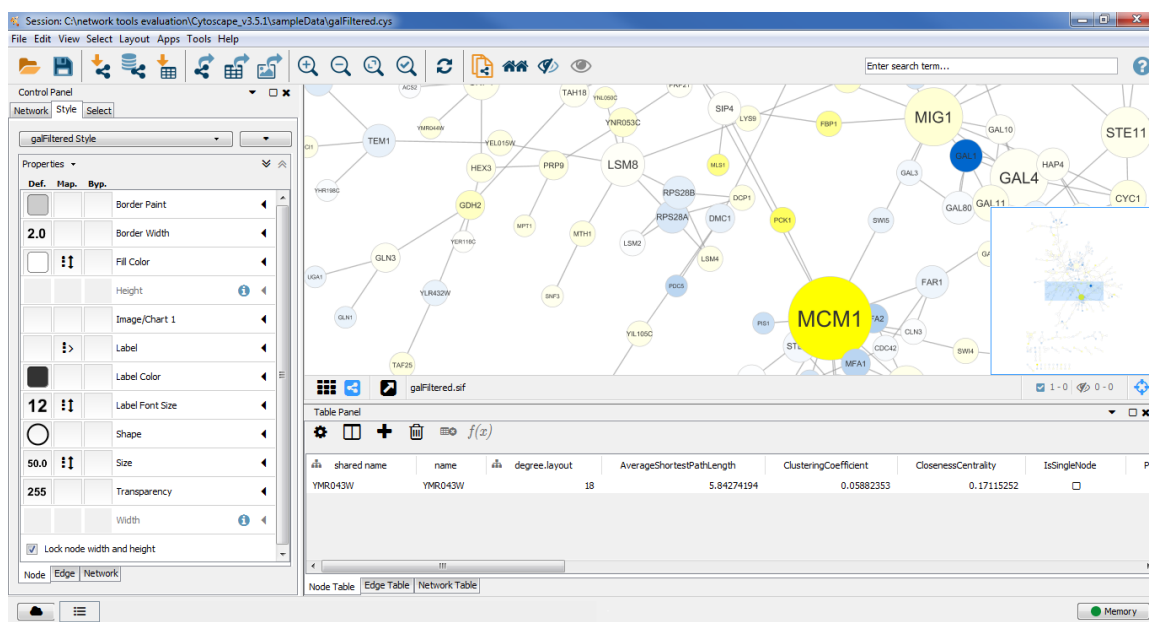


Figure 6: Cytoscape application window.

Cytoscape supports a wide variety of features that include:

- Apps architecture for expandability,
- import of networks via XGMML (eXtensible Graph Markup and Modeling Language), GraphML, CSV, and other formats,
- export of networks via GraphML, XGMML, PSI-MI level 1 & 2, CVS, JPG (Joint Photographic Experts Group), PDF, PNG, Postscript, SVG, and other formats,
- database integration,
- application of graph styles that affect entire network, that can use attributes,
- viewing of network data (nodes, links) as tables that allow the addition of custom data,
- custom functions in tables,
- a navigator for panning (good for large networks),

- a large selection of network layout options,
- a variety of network statistics,
- the extraction of subnetworks,
- the merging of networks and tables,
- a wide variety of custom apps that extend the functionality of the application,
- filtering of tables using table data, and filter chaining.

2.5.1 Use Cases

Cytoscape has been used in publications by researchers to analyze molecular structures. Some of these publications were the analysis of the Chemokine Receptor CCR1, Alzheimer's disease (AD), and the BK Virus. These uses cases will be discussed to show how Cytoscape played a role in their research.

2.5.1.1 *Chemokine Receptor CCR1*

Liang and colleagues identified the chemokine receptor CCR1 in mast cell-derived exosomes, where Cytoscape was used to plot protein-protein interaction (PPI) networks of differentially expressed genes (DEG; Liang et al., 2018). From the PPI network, CCR1 was screened as a hub gene enriched in cytokine from mast cell-derived genomes. Mast cells are important to the immune system, and mast cell-derived exomes that carry ribonucleic acids play an important role in the immune system regulation. The aim of the study was to “identify vital DEGs and pathways that are important for the function of mast cell-derived exosomes” (Liang et al., 2018).

The authors used a microarray dataset from the Gene Expression Omnibus (GEO) database. They performed gene ontology annotation and pathway enrichment analyses, and constructed protein-protein interaction (PPI) networks to identify DEGs. To identify hub genes, the Cluster One plugin for Cytoscape was used (Nepusz, Yu, & Paccanaro, 2012). Their results

provided insight into the roles that mast cell exosomal RNAs played in immunoreactions and disease.

2.5.1.2 MicroRNA and Gene Expression in Alzheimer's disease

In their study on Alzheimer's disease (AD), Moradifard, Hoseinbeyki, Ganji, and Minucheir objectives are the identification of microRNAs (miRNAs) involved in AD and their target genes, the determination of the most important miRNAs, genes, and their pathways in AD (Moradifard, Hoseinbeyki, Ganji, & Minucheir, 2018). The authors investigated and identified AD related DEGs and miRNA, miRNA-mRNA interactions, and signaling pathways. For proper diagnosis and treatment of AD, understanding its molecular mechanisms is required. The authors were able to obtain 179 differentially expressed miRNAs (DEmiRs) and 1404 DEGs using a comprehensive meta-analysis. They identified miRNA-30a-5p and miRNA-335 as the most regulatory miRNAs in AD. Cytoscape played an important role in the visualization of gene interaction and clusters, along with the visualization of subnetworks based on miRNA-30a-5p and miRNA-335.

2.5.1.3 BK Virus (BKV)

Jia and colleagues objective was to identify the PPI networks and implicated pathways of BK virus nephropathy (BKVN) via bioinformatics methods (Jia et al., 2018). Their process involved the identification of DEGs between stable and BKVN renal-transplantation recipients, creation and analysis of PPI networks, and the identification of key genes for better insight into the pathogenesis of BKVN. They used the microarray data GSE75693 of 30 patients with stable kidney transplantation and 15 with BKVN. This data was analyzed to identify DEGs. The authors used gene ontology (GO) functional enrichment analysis, the Kyoto Encyclopedia of Genes, and Genomes (KEG) pathway analysis to investigate the molecular function, biological process, cellular components, and pathways of the DEGS. Cytoscape was used to view the PPI networks

of DEGS. From their analysis CXCL100, EGF, and STAT1 were identified as hub genes which make them significant with respect to BKVN.

2.5.2 Discussion

Cytoscape was used for the construction of biological networks, and the identification of their important genes. This allowed researchers to identify key genes in viruses like Alzheimer's disease and the BK Virus. Cytoscape is now a general purpose tool with the goal of reaching a wider audience. It supports advanced ways of data importation through databases, data visualization using tables, and does not lack in the data visualization arena. Cytoscape aims to support symmetric attribute types across its data, whereas KVN aims to support unique attribute types across its data. Cytoscape does not support the application of multimedia in its networks. Cytoscape supports graph styles that uses attributes to render its network. However, it does not support the filtering of networks as outline in the problem description (1.4). Based on these evaluations, Cytoscape does not meet any of the objectives of this research and thus still leaves the gap open.

2.6 Pajek

Pajek is a cross platform network analysis and visualization program (Batagelj & Mrvar, 2018), able to handle very large networks containing on the order of one billion vertices (Mrvar & Batagelj, 2016). Technically the limit is tied to the amount of available CPU memory. Pajek has been available for over 20 years, as development began in 1996. It is free for noncommercial use and has been used by universities and commercial companies. According to Mrvar and Batagelj (2016), Pajek has been cited more 6000 times in Google Scholar. It also has a monograph entitled "Exploratory Social Network Analysis with Pajek," published by Cambridge University Press in 2005 (Nooy et al., 2005). It has also been ported to Japan and China. Like many general purpose network analysis and visualization tools, Pajek supports the analysis of

social networks; citation, and co-authorship networks; protein–protein interaction networks; transportation networks; archaeological networks; and the use of Pajek as a general network visualization tool (Mrvar & Batagelj, 2016). Figure 7 gives a view of Pajek. It shows the main application in the background with its six classes of data; networks, partitions, vectors, permutations, clusters, and hierarchies. In the foreground is the network view contains the drawing of the network along with any loaded data from the main application.

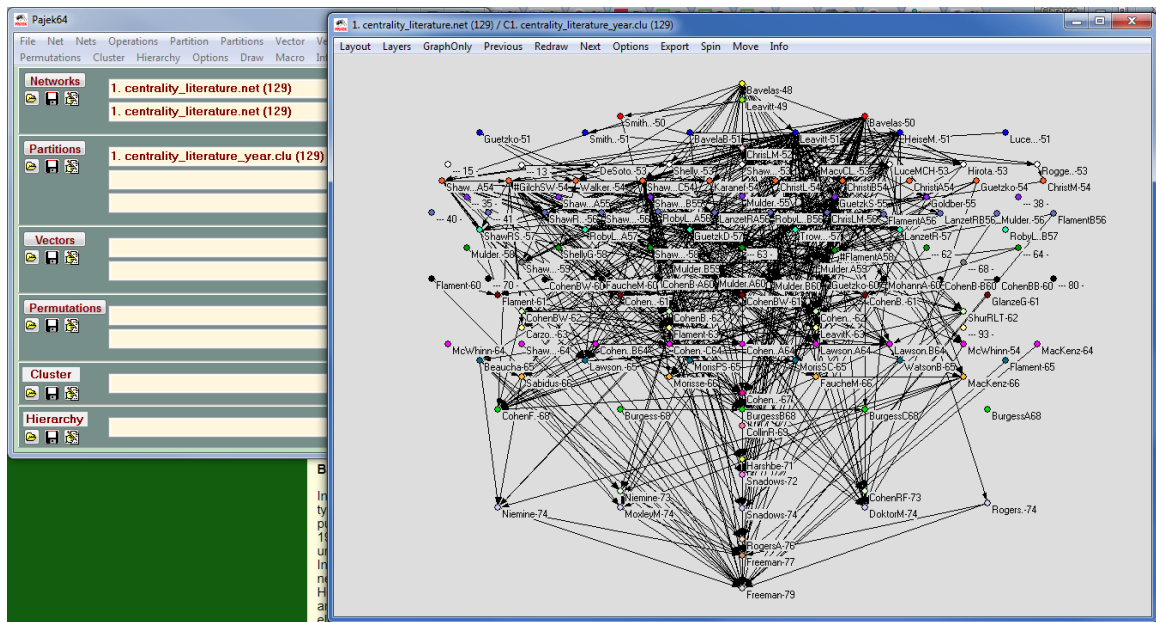


Figure 7: Pajek application window.

Pajek supports a wide variety of features that fall into all the categories of its main application window which include:

- network extraction from other data classes,
- network operations including (but not limited to) transformation, extraction, CPM (critical path method), path between two vertices, etc.,
- partition extraction from other data classes,
- partition operations such as adding, subtracting, merging, fusing, intersection, etc.,
- vector extraction from other data classes,

- vector operations such as addition, subtraction, multiplication, division, linear regression, etc.,
- permutation extraction from other data classes and fusing,
- cluster creation from partitions,
- hierarchy extraction from other data classes, and
- network analysis using R or SPSS.

2.6.1 Use Cases

Pajek is a widely used tool across various disciplines. Three use cases that involved the use of Pajek includes the interlinkage and organization of scholarly knowledge, and in the construction trade. While these uses cases may not be prominent ways in which Pajek is used, they will be discussed to show how Pajek played a role.

2.6.1.1 *Making Sense of Mankind's Scholarly Knowledge and Expertise: . . .*

In the year 2006, there were over half a billion records of scholarly knowledge indexed on the web. However, most of the data are difficult to traverse by citation, coauthor or other linkages. Reference desk librarians are good at traversing scholarly data. However, search engines do so in a poor manner, because they can give a local neighborhood view of citation links, but not a global view of how everything is interlinked. Börner (2007) sought to address the problem of collecting, representing, and interlinking scholarly knowledge, along with organizing scholarly knowledge (Börner, 2007). The author made use of Semantic Association Network (SAN) to improve collection, representation, and interlinkage of this information. Three different approaches were compared to tackle this problem, which involved expert consultation via questionnaires, application of bibliometric techniques and tools to citation data, and the analysis of personal bibliography files. Pajek was used in the construction of collaboration,

invitation networks, and coauthor networks. The networks used node shape, node size, node fill and outline color, and node color to distinguish several attributes of node and link data.

2.6.1.2 *InfoVis Lab (IVL)*

Neiryck and Borner (2007) implemented a database management system for leaders and members of scholarly units (Neiryck & Borner, 2007). The objective of the study was the development of a system with the capabilities to organize, store, and access information for research purposes. Their work is based on semantic association networks to organize and interlink people, projects, teaching, publications, presentations, calls, events, data sets, software, hardware, and funding. Their database schema was designed to handle the linkage of media items to any entity in their networks. Network visualization is handled by querying the database and using Pajek to visualize the network.

2.6.1.2.1 Discussion

Interestingly IVL comes the closest to the objectives of this dissertation. However, there are several things that need to be pointed out. First, the IVL uses a very complex database schema. From looking at the figure (Figure 8), there are 14 different entity types used by the IVL which require 14 different tables each. For any relationship between entities, a bridge table is constructed specifically for that relationship. As Neiryck and Borner (2007) states:

So-called bridge tables store the relationships between different data records, e.g. brdgAuthorsEditors links tblPeople and tblPublications or brdgTeamCollabs links tblPeople and tblProjects. In most cases, bridge tables also store additional attributes. For example, brdgTeamCollabs stores the time during which a link was active, e.g., how long a person was associated with a certain project and in what capacity (team lead, team member, external collaborator). (p. 126)

It can be argued that the database schema could have been made a little less complex by reducing the number of bridge tables required. The querying of such a database using those bridge tables could become complicated given the nature of how SQL syntax needs to be constructed. The IVL was designed for a specific purpose and may very well serve the needs for the management of research information given small teams. However, its design is not general purpose, where it would be able to tackle real world problems outside of its domain. Second, the media table used to store files is only capable of storing photos and text files. The publication is about 11 years old, so it could have been updated. However, the handling of media needs to go beyond just storing photos and text files, such as music, video, documents, and any other media that can be generated from data needs to be a part of this process.

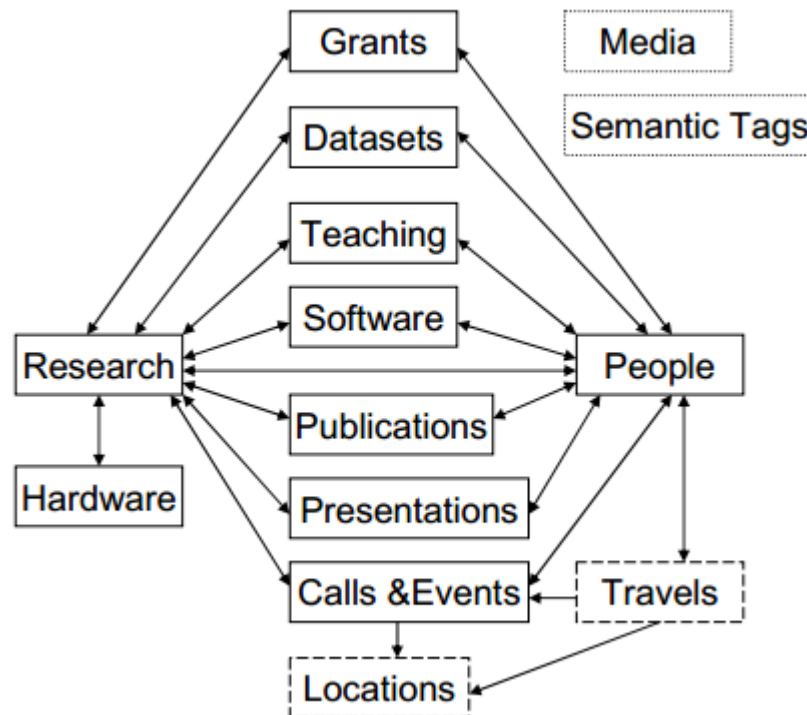


Figure 8: Simplified IVL DB Schema

2.6.1.3 *Using Pajek and Centrality Analysis to Identify a Social Network of Construction Trades*

Wambeke, Liu, and Hsiang, set out to accomplish two objectives in their research; the first was to “identify the organizational social network of trades existing among a construction project,” and the second was to “analyze the social network to identify the key trades within the network” (Wambeke, Liu, & Hsiang, 2012). The authors hoped to reveal the network arrangement and key trades, so project managers can potentially use them to influence other members in the network. For this research, the authors used a \$50 million construction project with 43 trades (a.k.a subcontractors) studied over a period of 28 weeks. The project itself was the conversion of a 150,000 sq ft. warehouse into a data center and white space computer lab. Pajek was used to generate 14 social networks where degree and eigenvector centrality were analyzed. This reflected the distribution of relationships through the network and the identification of key trades. The authors were able to find a social network with in the construction project and how it worked by analyzing how the various trades interacted with each other.

2.6.2 Discussion

The fact that Pajek can handle on the order of one billion vertices is impressive, with respect to other network applications. It has been used in a vast number of network research projects, which is also reflected in the number of times it has been cited in Google Scholar. However, Pajek can use some improvements in its capabilities of network partitions as outlined in the problem description (see section 1.4.3). Its network partitioning does not support multiple classifications, continuous vectors or links clusters. It also does not support storage of multimedia data in it networks. Based on these evaluations, Pajek does not meet the objectives of this research and still leaves the gap open.

2.7 SocNetV

Social Network Visualizer (SocNetV) is a free user friendly cross platform network analysis and visualization software application (Kalamaras, 2018). It allows for the creation of networks using a few mouse clicks, random network generation, the import of networks using various popular file formats, or the creation of networks by crawling connected webpages using the internet. SocNetV also supports a variety of layout algorithms, the calculation and analysis of various network metrics, and dynamic network models. Dynamic networks is a very interesting feature in SocNetV, as one can view a network at various time steps to see how the network changed over time. Figure 9 gives a view of SonNetV's application window. It shows the toolbar (top), control panel (left), network view (center), statistics panel (right), and timeline slider (bottom).

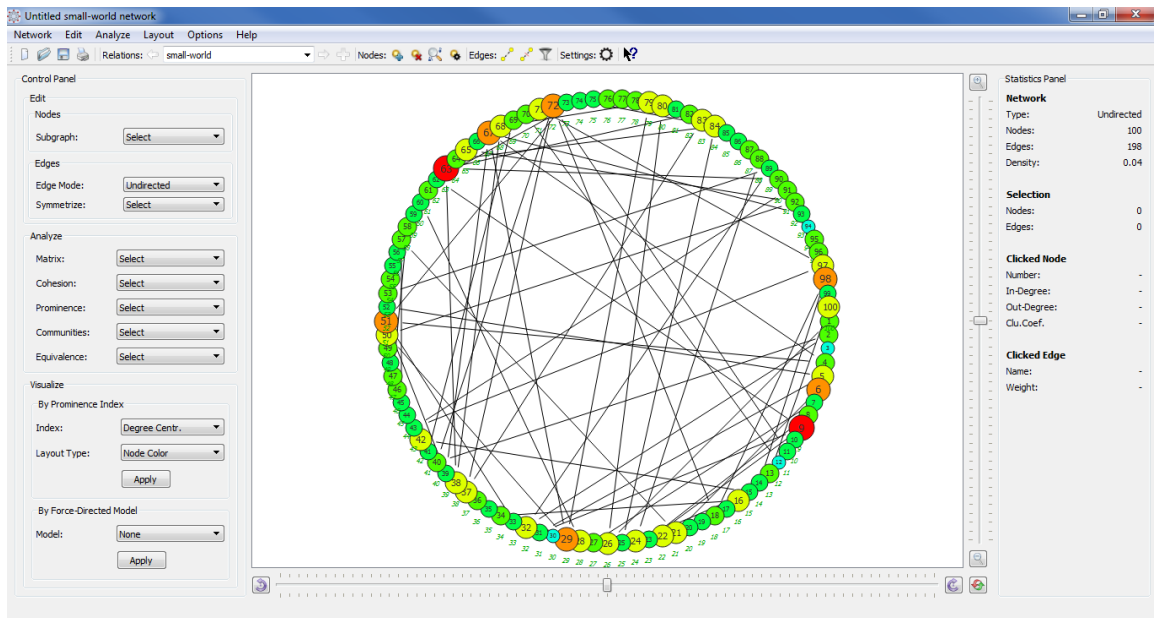


Figure 9: SocNetV application window.

SocNetV provides the following features:

- import networks via GML, GraphML, Pajek, CSV, and other formats,
- export networks via BMP (bitmap), PNG, PDF, Pajek,

- web crawler that can build networks from links,
- provides a variety of network statistics,
- provides a variety of network layouts, and
- dynamic networks.

2.7.1 Use Cases

SocNetV has been used in a conceptual study for the mining, constructing, and retrieval of information. Social networks contain a lot of information that can be used to construct other forms of information. A use case will be discussed to show how the information can be constructed and how SocNetV can play a role.

2.7.1.1 *Using social network analysis for Information Retrieval . . .*

The aim of Arora, Kanjilal, and Varshney's research was to "illustrate the use of Social Network Analysis (SNA) to map information retrieval and knowledge flow in teams with member of diverse areas" (Arora, Kanjilal, & Varshney, 2010). Their process involves the integration of information retrieval and SNA techniques for mining, constructing, and retrieval of information in the network for different tasks. They hoped to demonstrate that information gathered from SNA can be useful predictors for information and knowledge flow in teams.

Text analysis and the use of keyword or tags are different ways of extracting information from text sources. The problem is that in many cases, it is often not known who the user is and whether the user is relevant. From a network perspective, the user can act as a hub if large amounts of information originate from the user or passes through the user. Using SNA can reveal if topics are hot because of relevant people, or if they are small talk by bystanders. The authors conducted a conceptual study to look for socially enriched information about mobile phones with a newsgroup. Information from the mobile phones would be analyzed, the group structured would be evaluated, and finally they both are combined to gain knowledge. SocNetV

would be used as a part of their study to help understand the relationship between the data, the networks, and the combination of both. It is important to note that this study is conceptual and there is no evidence that the actual study was performed.

2.7.2 Discussion

SocNetV does have an impressive number of features. One of its most notable features that does not exist in the majority other network analysis and visualization applications is its dynamic network model. This is interesting as most network visualization applications only allow for the viewing of static networks. SocNetV takes this a step further and allows for the viewing of dynamic networks at different time frames. Aside from this, the application of multimedia and advance network filtering is not supported by this tool. Based on these evaluations, SocNetV does not meet the objectives of this research, and does not fill the gap identified in the problem description.

2.8 Meerkat

Meerkat is a tool for visualization and community mining of social networks (Chen et al., 2010). Meerkat's features include interactive network visualization and metrics, network querying and filtering, community detection in networks, and event analysis of dynamic networks. Meerkat filters are based on user defined attributes, predefined attributes naturally contained in the network, or derived attributes based on network statistics. Nested filters are allowed that user conjunctive and disjunctive Boolean operators. Two of the more notable features of Meerkat are automatic community detection and event analysis. By analyzing dynamic networks, Meerkat can detect communities and generate events based on how nodes changed over time. Figure 10 gives a view of Meerkat's application window. It shows the toolbar (top), project information (left), and network view (center). The right panel contains details, layouts, filters, and other functionality.

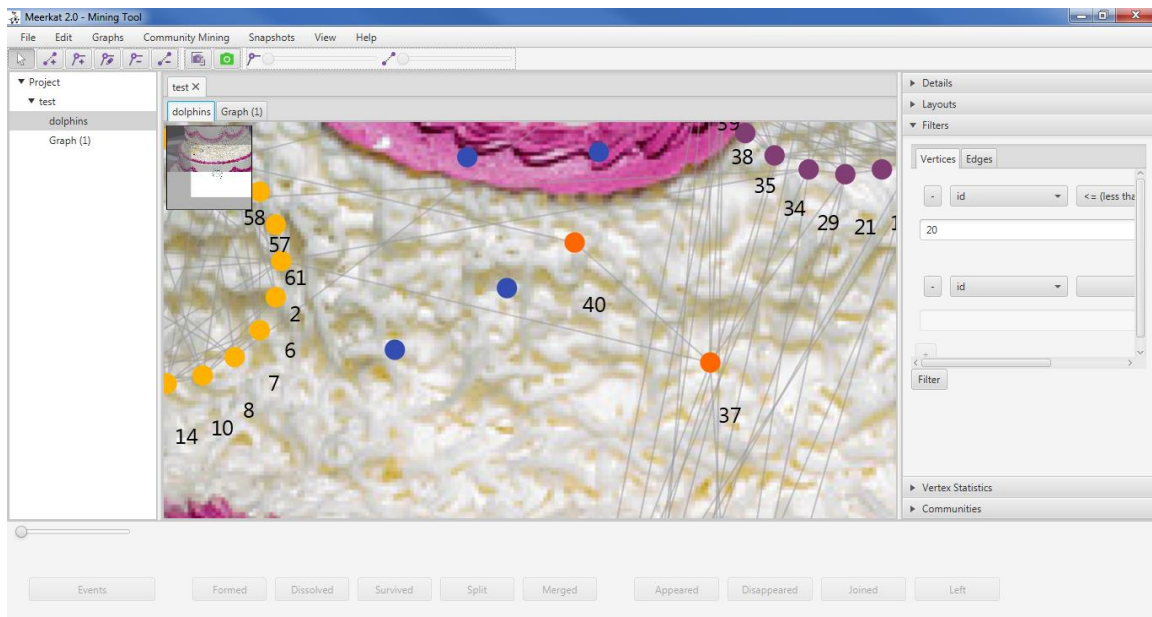


Figure 10: Meerkat application window.

Meerkat provides the following features:

- import networks via CSV, GML, GraphML, and other formats,
- export network via CSV,
- dynamic networks in which changes can be visualized over time,
- different layouts for viewing networks,
- network statistics (limited),
- filtering of network using properties such as edge weight and vertex degree,
- custom filters using attributes in nodes and links, and
- background image.

2.8.1 Use Case

Meerkat has been used in the analysis of dynamic social networks. This includes understanding how networks change over time with respect to community evolution. This use case will be covered to show this important field in network science and how Meerkat played a role.

2.8.1.1 *Analyzing Dynamic Social Networks*

Takaffoli, Sangi, Fagnan and Zaïane, set an objective to develop “an event-based framework to categorize and track how communities evolve in social networks” (Takaffoli, Sangi, Fagnan, & Zaïane, 2010). Traditionally, SNA is a static investigation of independent graphs at different time periods. According to the authors, this method misses the opportunity to capture evolutionary patterns in the network. Their framework can detect community evolution over time, and the tracking of individual nodes involved in the process. This is done by monitoring when communities form, dissolve, and reform to provide insight into dynamic social behavior. There is a total of eleven (11) events involved in their process. Seven (7) events are community specific and four (4) are node specific. For communities they track when they form, dissolve, continue, split, merge, shrink, and reform. For nodes they track when they appear, disappear, join, and leave a community. To test their framework, the authors analyzed an email dataset between employees from Enron Corporation. The dataset spans a period of 15 years, with 80,000 nodes and links over 100,000. Using their framework, they were able to detect and track communities within the Enron email dataset. Their framework was implemented into Meerkat.

2.8.2 Discussion

Meerkat is a nifty network analysis and visualization tool. It is as the forefront in dynamic network research, involving its community detection and event analysis tools. Meerkat filters fall in line with being more flexible and robust over the competition. Their filters allow the use of user defined attributes, predefined network attributes, and derived attributes using network statistics. This falls in line to a high degree of the network partitioning gaps discussed in the problem description. However, Meerkat does not support multimedia data and still leaves this gap open.

2.9 Network Work Bench

Network Work Bench (NWB) is a “network analysis, modeling and visualization toolkit for biomedical, social science, and physics research” (Herr, Huang, Penumarthy, & Börner, 2006). NWB was developed as a part of the cyberinfrastructure (CI) initiative lead by a blue ribbon committee from the National Science Foundation (NSF). The objective was to serve the needs of algorithm developers, application designers, and application users, and is known as the CIShell specification. The specification was built on the Open Services Gateway Initiative (OSGi) specification, which was developed as an easy to use wizard driven plug and play system, for algorithms and datasets implemented as services (Herr et al., 2006). OSGi defines a standardized, component oriented, computing environment for network devices; and has been used from high end servers to embedded mobile devices for several years. Some of the alliance members include IBM, Sun, Intel, Oracle, Motorola, NEC, and others. Figure 11 gives an overview of NWB’s application window. It shows an attribute view (left), network view (center), and various tools (bottom).

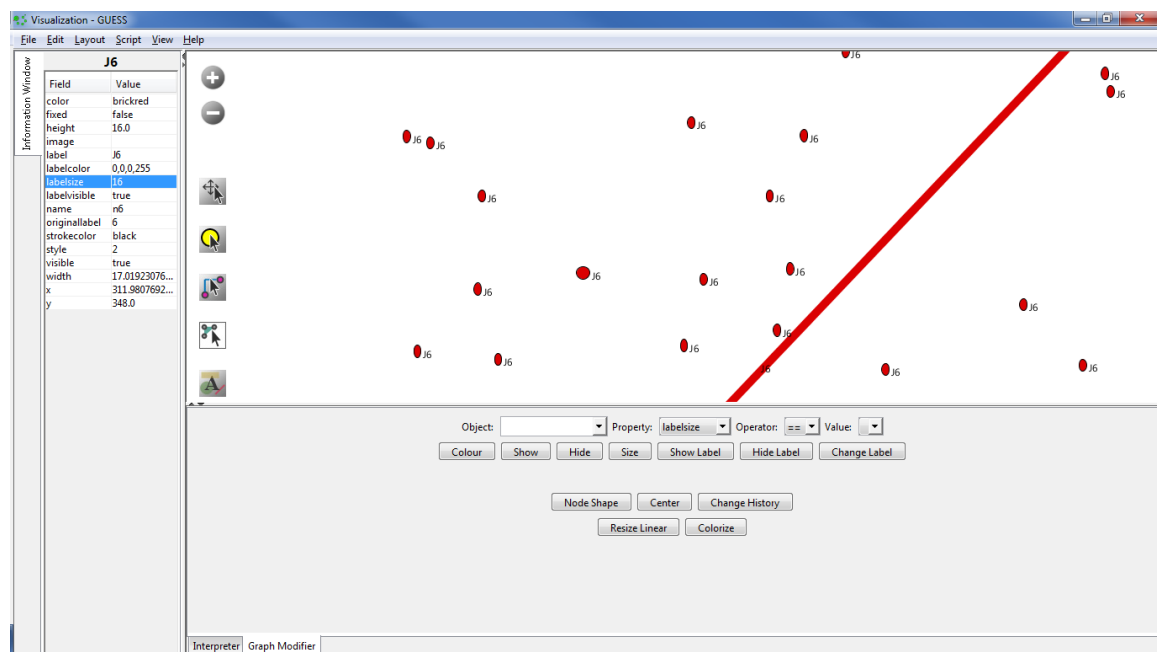


Figure 11: NWB application window.

NWB provides the following features:

- provides a plugin system for extensibility,
- provides a variety of network analysis functions,
- provides a variety of network visualizations that decides how the network is laid out,
- provides functions to transform the network.

According to the authors, computer scientists are not the only ones that develop novel datasets and algorithms. Biologists, physicists, social scientists, etc, also need to develop such datasets and algorithms, some which are specialized to their field. Thus, it is a waste of resources to have algorithms reimplemented. It is better to have high quality algorithms and datasets readily available to be used. Algorithm developers need to quickly disseminate their work, so they are in search for appropriate tools. Researchers and educators are not equipped with the mathematical sophistication and programming knowledge to implement their algorithms. Users need multiple algorithms with user friendly tools that can be customized according to their needs. Thus, the CShell specification was born to address these needs, and NWB is one of the products of a seven year long quest into the development of a 'dream tool' for research in scientometrics and network science (Herr et al., 2006).

2.9.1 Discussion

It was difficult to find any publications on NWB that showed how the tool was used in the field. From evaluating NWB it does not feel like a user friendly tool. However, it does support various graph generation algorithms, preprocessing of data, a large variety of analysis algorithms, visualization and layout algorithms, and algorithms for scientometrics. One notable thing about NWB is the CShell spec in how it brings together data, algorithms, and services. However, that falls more into software design and goes outside the scope of this dissertation. NWB though advanced in its software architecture, does not meet the objectives of this

research. It does not support emphasis on data in links, multimedia or advanced network partitioning. Thus, NWB still leaves the gap open.

3 Chapter 3: Methodology

3.1 Overview

The System Development Life Cycle (SDLC) will be used to address this research problem. SDLC is a structured approach to systems development used to “to plan, analyze, design, implement, and support an information system” (Shelly & Rosenblatt, 2012). Figure 12 lays out the SDLC process group into four parts. The first part consists of the JAD Session being held with Users, Stakeholders, and System Engineers. The second part consists of the requirements model being developed from the problem description through conceptual modeling. The third part consists of the system implementation from analysis and design. The fourth part consist of testing the system and overall evaluation. These all will be addressed in the following sub sections.

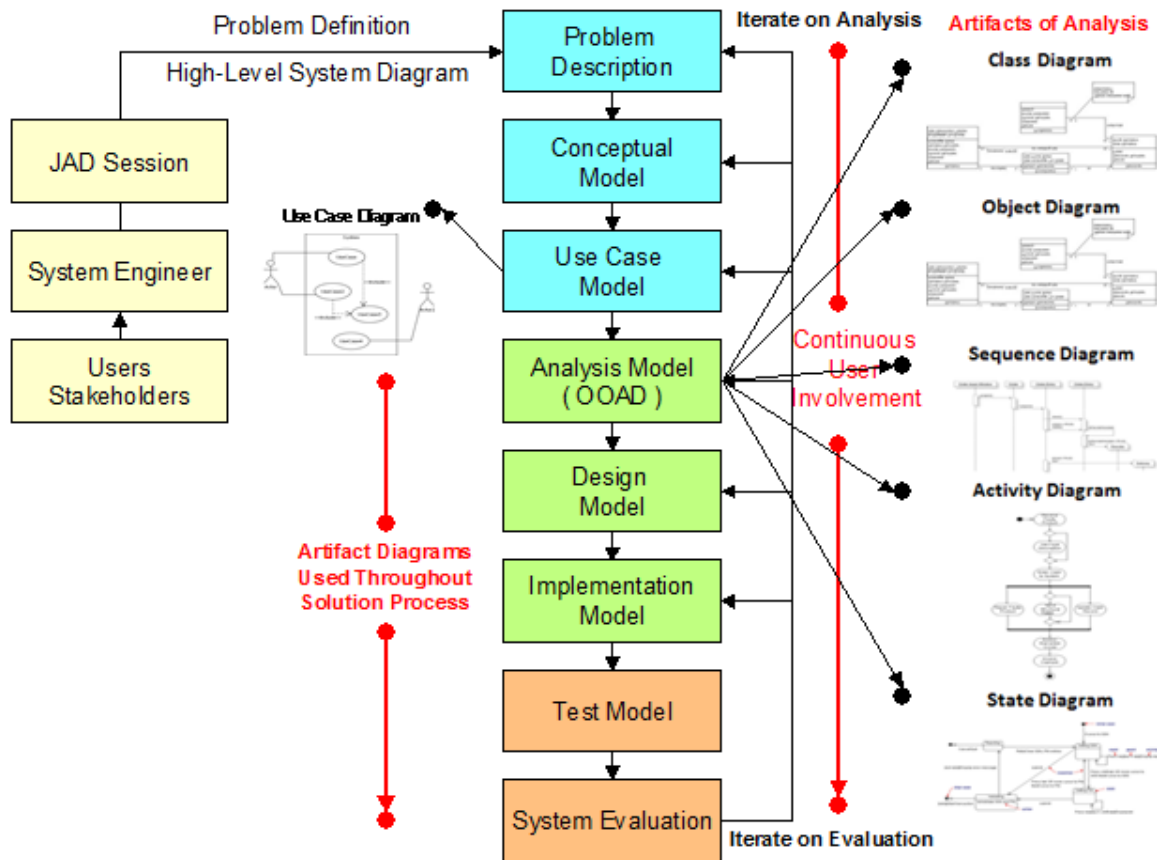


Figure 12: Systems Development Life Cycle (SDLC; Tsang et al., 2005)

3.2 JAD Session

Joint Application Development (JAD) Session is one of the first steps in the SDLC. JAD is a “user-oriented technique for fact-finding and requirements modeling” (Shelly & Rosenblatt, 2012) that involves the Systems Engineer, Users, and Stakeholders of the system. The objective of JAD (see Table 6) is to interact with users and stakeholders to analyze the existing system, develop a problem description, and derive user requirements for the new system.

Table 6: Typical JAD Agenda (Shelly & Rosenblatt, 2012).

Project leader	<ul style="list-style-type: none"> • Introduce all JAD team members • Discuss ground rules, goals, and objectives for the JAD sessions • Explain methods of documentation and use of CASE tools, if any
Top management (sometimes called the project owner or sponsor)	<ul style="list-style-type: none"> • Explain the reason for the project and express top management authorization and support
Project leader	<ul style="list-style-type: none"> • Provide overview of the current system and proposed project scope and constraints • Present outline of specific topics and issues to be investigated
Open discussion session, moderated by project leader	<ul style="list-style-type: none"> • Review the main business processes, tasks, user roles, input, and output • Identify specific areas of agreement or disagreement • Break team into smaller groups to study specific issues and assign group leaders
JAD team members working in smaller group sessions, supported by IT staff	<ul style="list-style-type: none"> • Discuss and document all system requirements • Develop models and prototypes
Group leaders	<ul style="list-style-type: none"> • Report on results and assigned tasks and topics • Present issues that should be addressed by the overall JAD team
Open discussion session, moderated by project leader	<ul style="list-style-type: none"> • Review reports from small group sessions • Reach consensus on main issues • Document all topics
Project leader	<ul style="list-style-type: none"> • Present overall recap of JAD session • Prepare report that will be sent to JAD team members

3.2.1 Problem Definition

During the JAD Session, the problem definition is defined. This is usually done from analyzing the existing system and the user experiences. A High Level Systems Diagram (HLSD) is developed during this session to visualize and support the narrative problem definition. The problem definition is a very short description of the problem such that upper management and persons not necessarily technically aware can have a high level understanding of the particular problem.

3.2.2 Problem Description

After the initial JAD discussions are complete and a high level problem definition has been agreed upon, the system engineering team will formulate a detailed in depth description

of the problem. This description supports the in depth research and analysis moving toward the problem solution. All the critical problem elements will be defined along with the many relationships between these elements. As shown in Figure 12, the user is still heavily involved in the problem description process.

3.1.3. Conceptual Model

The conceptual model is developed from the problem description and is one of the most important problem visuals in the solution development process. In the process of developing the conceptual model all the critical concepts driving the problem and solution description are defined and collaborated with experts in the field. This model is a very detailed visualization of the problem and its initial introduction to the solution concepts. The conceptual model defines all the relationship between these critical problem and solution concepts. Also, there may be very important details to be captured concerning the conceptual object relationships. The conceptual model is the primary input for the system requirements model, which is sometimes referred to as the use case model.

3.1.4. Use Case Model

The use case model is an important part of the SDLC; a system is more likely to meet the needs of a user if the use case model is utilized in the early stages of system development (Tsang et al., 2005). The use case model is made up of two components the use case diagram (Figure 13) and use case scenario (Figure 14). The model serves as a set of system requirements for the new system which are developed from the conceptual model. The use case model is a set of diagrams with additional documentation that shows the requirements of the new system; it includes actors, use cases with lines of communication between actors, and use cases, as well as documentation termed use case scenarios. These scenarios define the initial steps required to fulfill the user requirements.

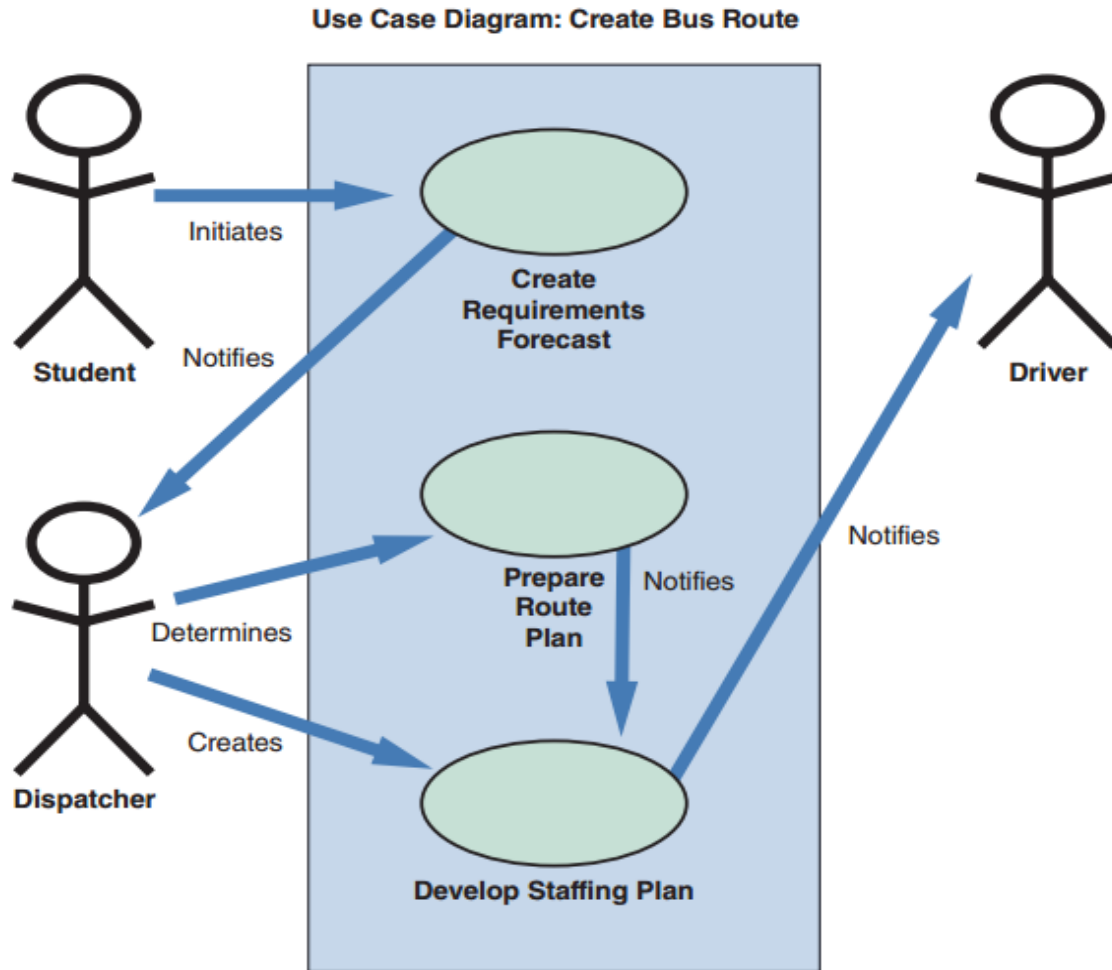


Figure 13: Use case diagram create bus route (Shelly & Rosenblatt, 2012).

3.2.2.1 Use Case Scenarios

Use case scenarios are the sequential steps written in narrative required to define how the user requirements will be fulfilled. They contain preconditions, post conditions, assumptions, and alternative courses of action (Shelly & Rosenblatt, 2012). The scenarios are the primary input for the analysis step in the SDLC methodology. The class diagrams are created directly from the objects, relationships, and logic of the scenarios, as outlined in Figure 15.

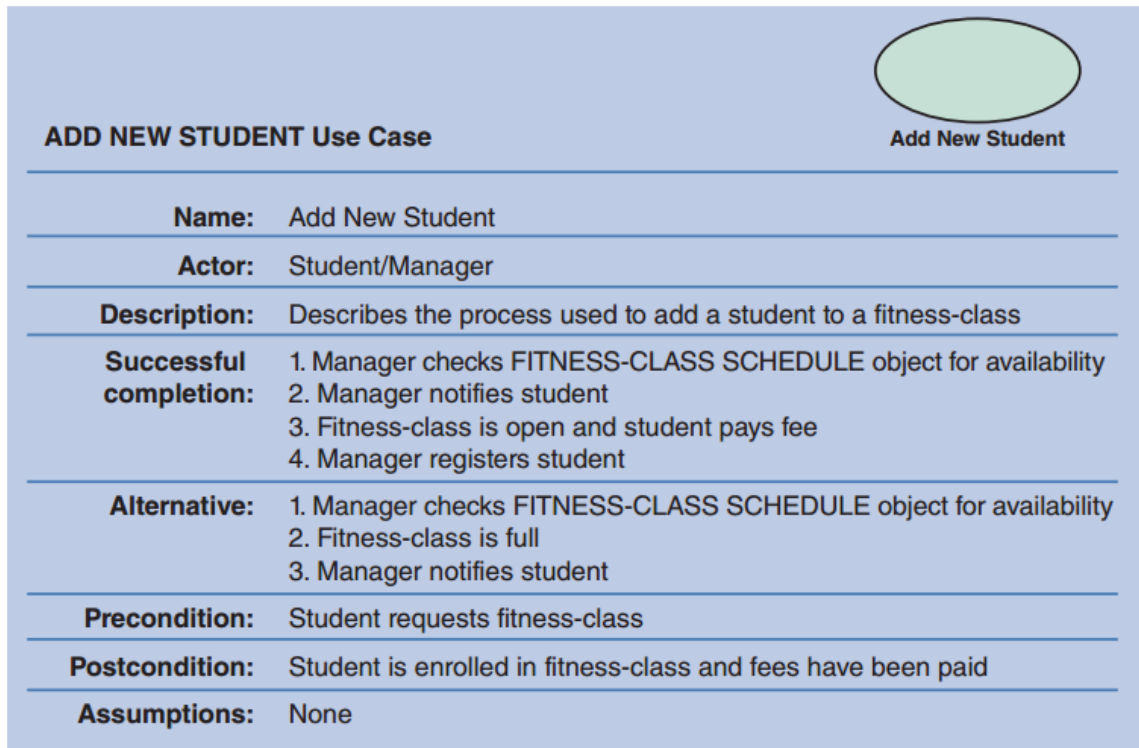


Figure 14: Use case scenario add new student (Shelly & Rosenblatt, 2012).

3.3 Analysis Model (OOAD)

The analysis process is used to analyze the use case model in accordance with the conceptual model to develop a logical system solution. This research will use OOAD. OOAD is used to model and analyze real world business processes, information systems, and operations using objects and interactions between them (Shelly & Rosenblatt, 2012). OOAD is composed of use case models, class diagrams, object diagrams, relationships, sequence diagrams, state diagrams, and activity diagrams, where each serve a specific purpose. Use case models, class diagrams, and object diagrams are static models. Sequence diagrams, state transition diagrams, and activity diagrams are dynamic models that show how objects behave and interact with the system (Shelly & Rosenblatt, 2012).

3.3.1 Class Diagram

A class diagram is a logical model of the system, developed from analyzing the use case scenarios. The objective is to transform entities, data stores, and processes into data structures using structured analysis (Shelly & Rosenblatt, 2012). Class diagrams contains classes, their attributes, and methods essential to the operation of the system. Class diagrams also contains relationships that show how classes relate to each other. They may also include interfaces and enumerations.

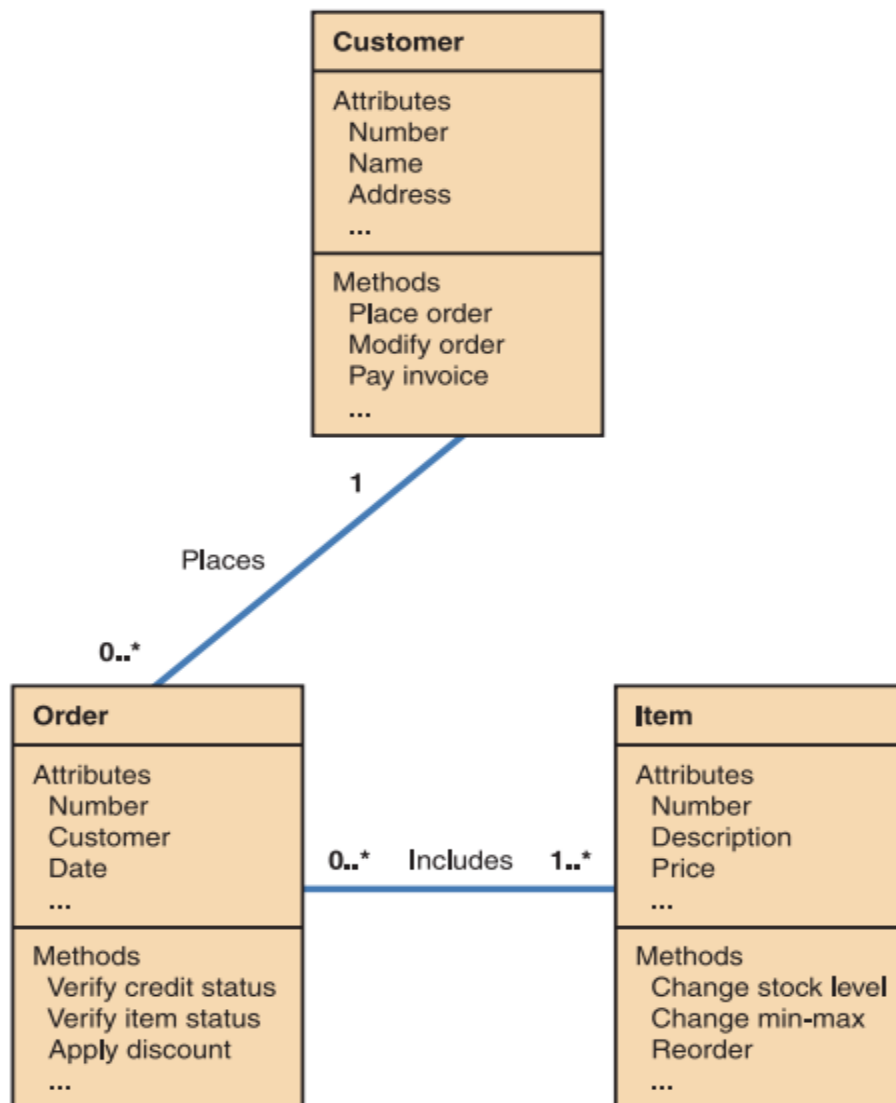


Figure 15: Class diagram customer order processing system (Shelly & Rosenblatt, 2012).

Classes are defined by the rectangular boxes where the name, attributes, and methods are defined. Connecting lines are used to define the relationships between classes. The cardinality between classes are defined using the numbers associating the classes. In Figure 15, the association between classes "Order" and "Item" is "0..*" to "1..*." This means that 0 to many orders can have 1 to many items. Relationships are important in class diagrams as they show how classes communicate and relate to each other. Relationships help to "describe what objects need to know about each other, how objects respond to changes in other objects, and the effects of membership in classes, super classes, and subclasses" (Shelly & Rosenblatt, 2012). Relationships shown in class diagrams are reflected throughout OOAD in other diagrams and shows that the models are uniform.

3.3.2 Object Diagrams

Object diagrams are static views of a class diagram that represent a snapshot in time. An object is a self contained entity with well defined attributes and operations, where attributes define the state of the object and operations define its behavior at a specific state (Tsang et al., 2005). Object diagrams show objects at a specific state and their possible behaviors at that state. Object diagrams are useful in defining instances of use case scenarios to ensure that class diagrams logically make sense.

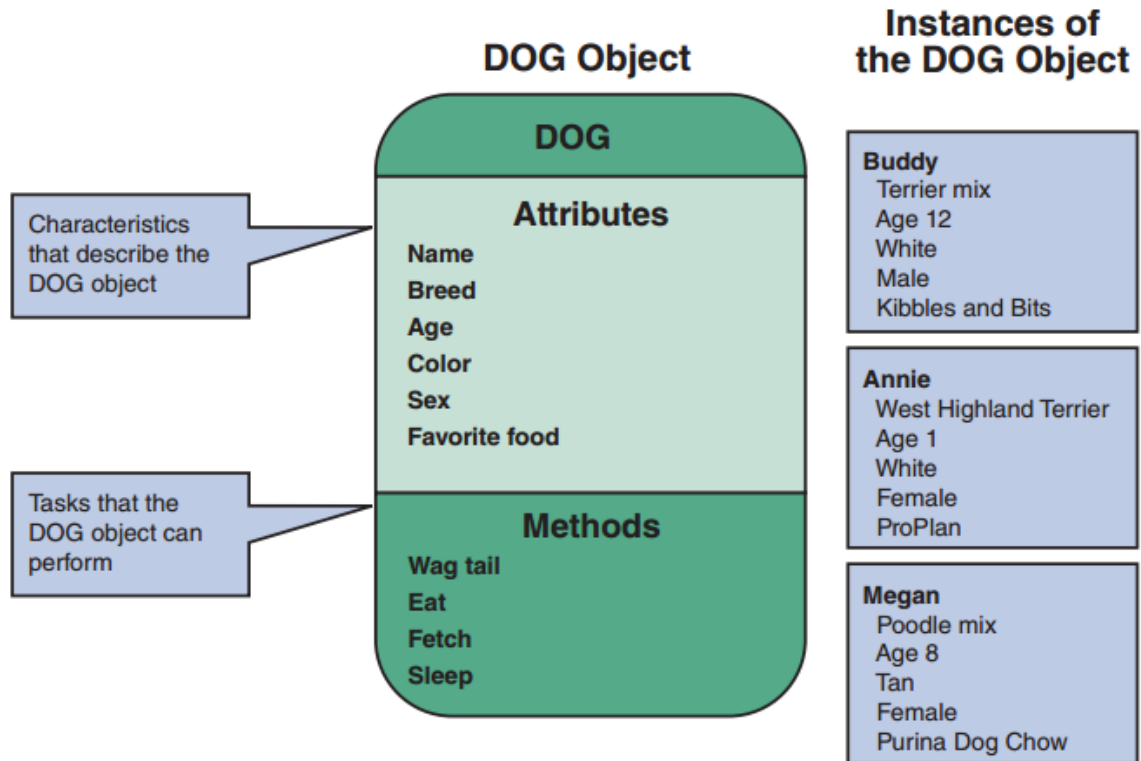


Figure 16: Dog object diagram (Shelly & Rosenblatt, 2012).

The structure of the object diagrams follows the structure of class diagrams very much. An object structure contains a name, attributes, and methods. As shown in Figure 16, the object is a dog with attributes and methods that help to define it. However, their parameters define an object at a specific instance in time. If objects are defined at multiple instances in time, it is possible to define the progress and change in state of the system.

3.3.3 Sequence Diagram

A sequence diagram (Figure 17) is a dynamic model of a specific operation of the system. It shows the objects involved and the relationship between these objects. The diagram has a vertical dimension which represents the passage of time relatively with reference to other objects. The horizontal dimension defines the objects involved. The sequence of object interactions is placed on a grid. This shows the sequence in which object messages take place, and the approximate length of time relative to other operations.

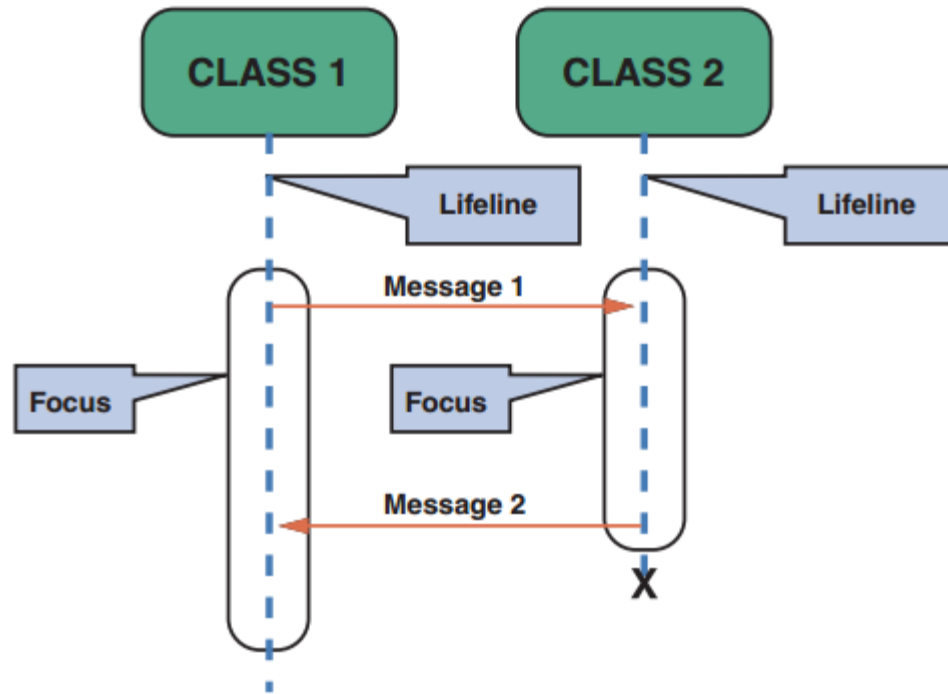


Figure 17: Sequence diagram (Shelly & Rosenblatt, 2012).

3.3.4 State Diagram

A state diagram shows the states of an object and how it transitions from one state to another. A state diagram also shows the events that cause state transitions, the conditions that guards transitions before they occur, and the actions that are taken during the life of an object (Copeland, 2003).

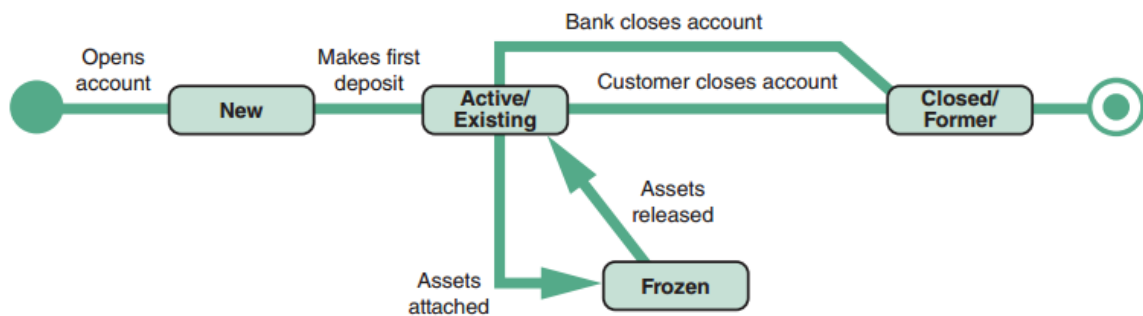


Figure 18: State transition diagram bank account (Shelly & Rosenblatt, 2012).

Figure 18 shows an example state diagram of the states and operations of opening and closing a bank account. It has a beginning represented by a filled circle when the bank account is

opened, and an end represented by an opened circle with a dot after the bank account has been closed. In between the start and end are states represented by rectangles, and the possible operations represented by lines that can be performed at each state.

3.3.5 Activity Diagram

Activity diagrams are flow charts used to layout and document a process involving steps and decision logic. They are used to (“UML Activity Diagram Tutorial | Lucidchart,” 2019):

- Demonstrate the logic of an algorithm.
- Describe the steps performed in a UML use case.
- Illustrate a business process or workflow between users and the system.
- Simplify and improve any process by clarifying complicated use cases.
- Model software architecture elements, such as methods, functions, and operations.

Activity diagrams can also “display multiple use cases in the form of a grid, where classes are shown as vertical bars and actions appear as horizontal arrows” (Shelly & Rosenblatt, 2012).

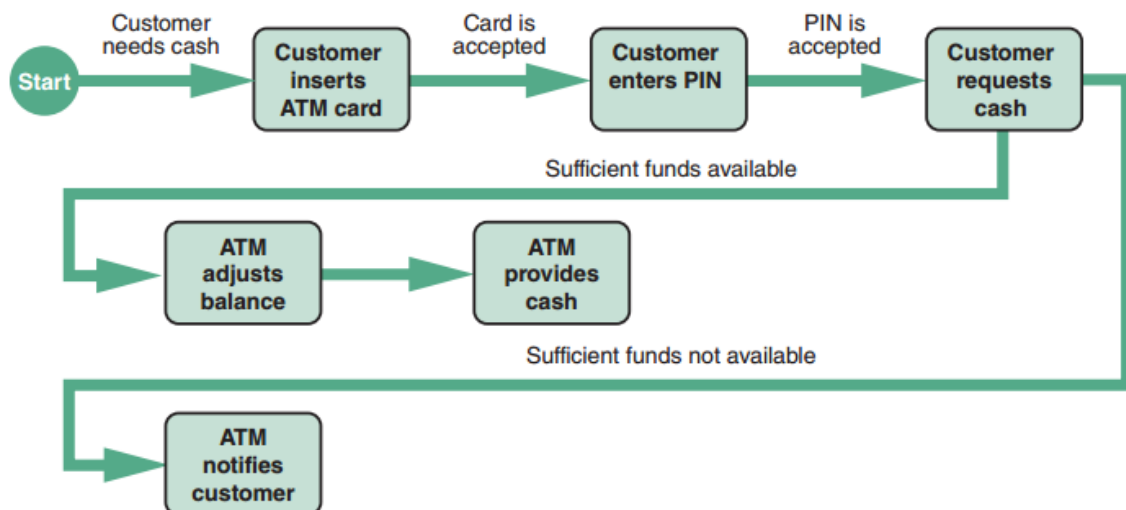


Figure 19: Activity diagram ATM cash withdrawal (Shelly & Rosenblatt, 2012).

Figure 19 is an example activity diagram that shows the process of withdrawing cash from an ATM machine. The beginning is a filled circle that leads to an event that initiates the process of cash withdrawal. Upon successful entering of the PIN, the diagram shows different paths taken determined by whether sufficient funds are available.

3.4 Design Model

The design model is the point in the analysis where the problem solution moves from the logical or abstract model to the physical model. The physical design implements real world specifications and criterion. The application architecture, user interface, inputs, and outputs are done in this stage. This model is used by programmers to understand the logical design. A system design specification is delivered at this stage to management and users for review and approval (Shelly & Rosenblatt, 2012).

3.5 Implementation Model

The implementation model is the construction of the system using the system design specification. Programs are written, tested, and documented with the objective of creating a fully functional system (Shelly & Rosenblatt, 2012). From a software perspective a programming language would be used. It is important to know that the implementation model is geared to the kind of problem being solved.

3.6 Test Model

The test model is the creation of tests to be performed by an integration and test (I&T) team. This can be implemented as manual tests using a step by step procedure, where each step must be evaluated by I&T personnel, to ensure proper functioning of the system. This can also be implemented as automated testing. Automated testing is very useful in reducing the amount of time required to run tests.

3.7 System Evaluation

The system evaluation phase assesses the overall quality of the system. This can be considered a two step process involving I&T and Quality Assurance (QA) personnel. In the first step, I&T uses the test model to determine if the system meets all specified requirements and user objectives. Any failures at this stage would be sent back to the implementation stage for rework. The second step is the execution of software quality testing (SQT) involving QA personnel. This is to ensure that the system does indeed meet all requirements before it can even be delivered to the customer (Shelly & Rosenblatt, 2012).

3.8 Proposed Application Modules & User Integration

Figure 20 gives an overall view of connecting a user to a real world problem. It shows how a user interacts with the application for developing their own research, as well as using services provided by the application. The first path of application integration takes a real world problem, develops it into a user application to fit the problem need, integrates the application by collecting appropriate information tied to the research, and then importing information into the application to build knowledge visualization networks. The second path is the manual interaction of the user with the system to use services that the system provides.

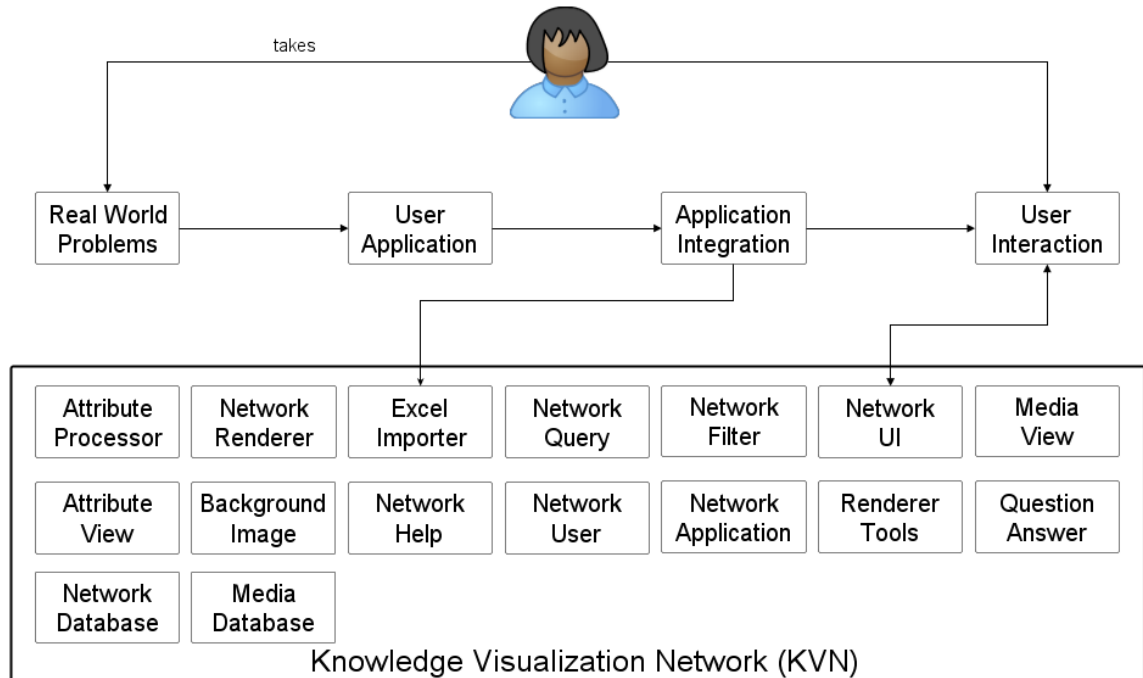


Figure 20: KVN application integration

3.9 Conceptual GUI

Figure 21 gives a concept of the application GUI. There is a primary and secondary application window. The entire application should be able to run inside the primary window. The primary window contains a toolbar with application functions, a network view responsible for displaying a network, an attribute view responsible for the viewing and editing of currently selected entity (node/link) attributes, a multimedia view responsible for displaying media of the currently selected entity (node/link), and a multimedia popup that allows a larger view multimedia. The secondary window allows for an even larger workflow where two displays can be used. Its function is to run the multimedia view separately from the primary application.

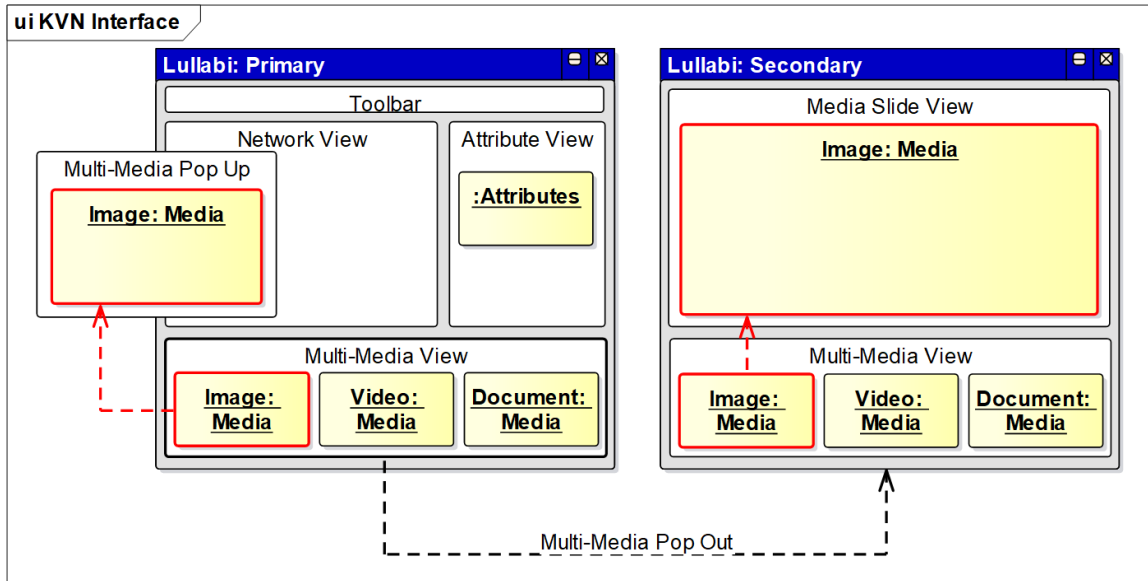


Figure 21: KVN Application GUI

Figure 22 gives a concept of the application GUI in use. It shows a network within the network view, a selected entity (C), the attributes of the selected entity in the attribute view, the multimedia of the selected entity in the multimedia view and a popup of a selected media (purple star). The figure also includes the secondary window showing multimedia information.

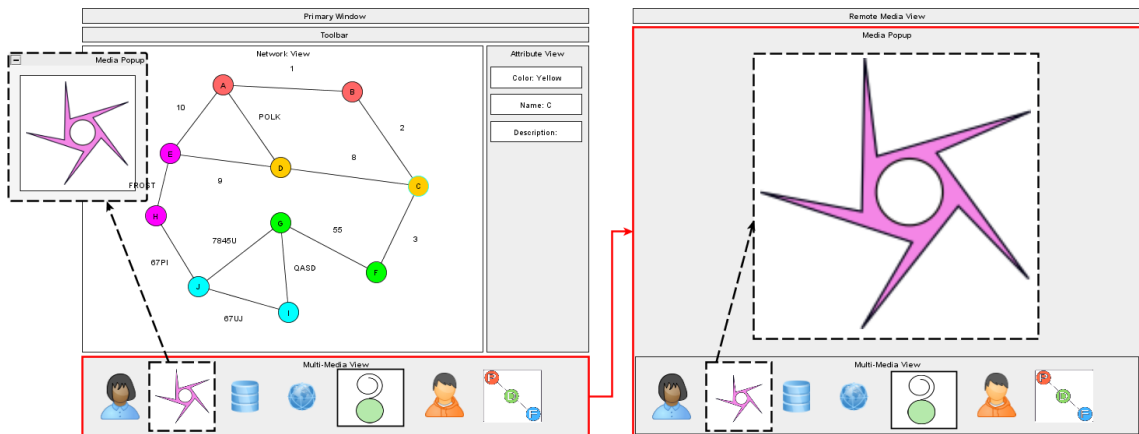


Figure 22: KVN Application Example

3.10 KVN Example Usage

To show the usefulness of KVN two use cases will be looked at. The Morgan Community Mile (MCM) and a Morgan State University (MSU) senior undergrad research project will be used to show the capabilities of KVN. The following sections give an idea of what these projects are.

3.11 Morgan Community Mile Example

The Morgan Community Mile (MCM) is an initiative to improve Northeast Baltimore through a Morgan University Community partnership that involves residents, businesses, public agencies, and other stakeholders. The program serves as a social laboratory in which the University's scholarly research, creative activities, and public service programs are concentrated in a 1 mile radius centered around Morgan in order to give significant priority to solving difficult social problems in urban communities. The purpose is to "engage with the community residents and officials in the use of knowledge derived from faculty and student research, the sharing of mutually beneficial resources, and the appropriate and timely dispatch of University experts and professionals to collaborate in addressing community concerns" (Wilson, 2011, p. 5).

3.11.1 Structure

MCM is composed of a board, staff, partners, communities, and projects. Information about MCM most likely exists in databases, spreadsheets, graphs, statistical analysis, reports, powerpoint presentations, and other forms of media. From a high level view MCM's composition has some sort of network. This network has information and resources that exists at different entities and dictates how information and resources flow throughout the structure. There are different levels to viewing MCM's structure depending on the users. The users are the small public, wide public, and researchers. The intent will be to show the advantages that KVN

brings to the table with respect to researchers. These are outlined in Table 7 and will be further discussed.

Table 7: MCM Use Case Contrast

Power Point	Website	KVN
Presenter Driven	User Driven	User Driven
Linear	Non Linear	Non Linear
Fixed	Fixed	User Definable

3.11.2 Small Public

The small public represents a small set of users who needs an overview of what MCM is. This should be nothing more than a high level view and should not get into too much underlying details about entities that exist in its structure. A power point presentation perfectly fits this use case. A presenter would guide an audience through a fixed set of slides in a linear fashion as shown in Table 7.

3.11.3 Wide Public

The wide public is a broader audience who needs more in-depth information into the inner workings of MCM. This should provide information that would not be made available from a small public use case. This information should be in-depth details about MCM's structure with underlying details about entities that exist within the network. A website would perfectly fit this use case; like the one which MCM currently hosts at <https://communitymile.morgan.edu>. The website itself has a structure that is a representation of MCM's structure, laid out in a logical format. A user self directs themselves through the website in a nonlinear way putting them in the driver seat. This allows them to consume the information at their own pace.

3.11.4 Researchers

Researchers represent a group of individuals whose goal is to understand, build, analyze, and visualize information about MCM, with relation to its resources, communities, and projects. The purpose would be to improve the location and flow of resources, understand its communities, what mostly impacts them, and what projects would best serve in making improvements based on the resources available. To accomplish these goals a tool would be needed to help analyze and visualize MCM's structure, along with information about its communities, and projects. KVN can serve this purpose to represent MCM as information networks with multimedia for network visualization enhancement and data representation. Researchers would be in the driver's seat to build information networks which are driven by researchers, nonlinear, and defined by researchers.

3.11.5 KVN

KVN can bring a unique perspective to the storage, analysis, and visualization of MCM. This would involve storing MCM's structure as information networks. These networks would contain information about entities within MCM and their relationships along with multimedia for visual enhancement. It can be used both as a research tool, and also as a tool to give multimedia oriented network presentation of the research.

For example, to determine the best way to improve the urban communities that are a part of MCM, researchers need to understand what those communities are like along with the factors that are having negative and positive effects on them. To aid in this process KVN can provide a network of the communities with descriptive attributes and multimedia. Attributes stored would be community statistics such as demographics, income level, mortality rate, etc. Multimedia would include images, videos, and reports written up about the communities. With information collected and stored in the KVN, it can be persisted to a database. The network of

the communities then acts as a research tool to analyze and visualize its information. It can show positive and negative attributes and how they relate to each other. One can show the income level and what kind of relationship it has to the number of children living under poverty. This information can then be used to determine which projects should be funded in the community. As the projects are taking place, information about the projects along with the effects on the communities can then be placed back into the network to analyze and visualize if the projects are having its intended effects.

3.12 Morgan State University Undergrad Senior Research Project

The Alpha Kappa Alpha (AKA) Sorority launched the CAPSM program to help direct high school seniors in their college application process. The program supports the enrollment/entry process by helping to research different colleges and majors, along with the overall lifestyle of each college. A group of MSU undergrad seniors were tasked with designing an implementation model for CAP. The implementation model is to be used as a framework by a local chapter AKA to direct the flow of CAP. The objective of the research project is as follows:

- provide a complete implementation model for the sorority to use to execute the program,
- develop a functional database to track and store student progress,
- provide visual representation of all program elements.

In the implementation process there are database structures that need to be developed to store information on each student. Microsoft Access will be used as the primary database to store information on students involved in the process. From Microsoft Access, data can be exported to an excel file. The excel data will then be imported into KVN which will serve as a secondary database. It will also be used to store information from a multimedia perspective and provide analysis using network filters. Figure 23 lays out the use case of KVN in the process. The

researchers must gather information and media about entities involved in the process and how they are related to each other. This information must then be transformed into an excel spreadsheet, so it can be imported into KVN through its excel importer. This would allow KVN to display networks filled with information and media within entities.

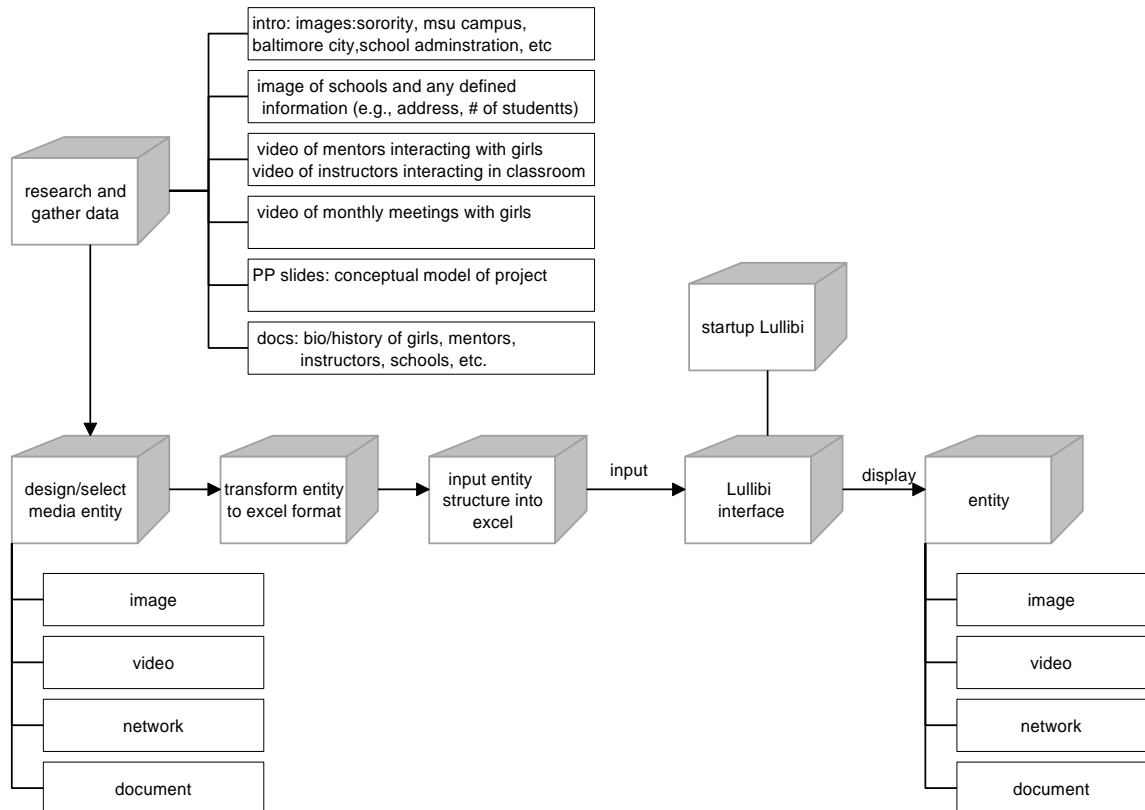


Figure 23: CAP Project

4 Chapter 4: Results

4.1 KVN Conceptual Design

4.1.1 Entity Component Model (ECM)

To understand the capabilities within KVN it is important to understand the concept of its design. At the heart of KVN exists a modification of the entity component system (ECS). ECS is an architectural design pattern popularized in game development, that follows composition over inheritance to allow greater flexibility in defining entities (Martin, 2007). Composition is done through the addition of components which add functionality to entities. KVN's ECM is a modified ECS that allows for the building of entities where components are used to represent data and functionality. The ECM is laid out in Figure 24. The components used in KVN are network, node, link, graphic, media, filter, and attributes. An entity needs to be carefully crafted dependent on its intended purpose. For example, an entity would need a node, link or network component to define it as such, but it cannot have all three. This definition means that nodes and links can be a part of a network, but networks cannot be a part of nodes or links. However, entities defined as a node or link would require the graphics component to define how it should be visualized. Moreover, the attributes component allows for user definable data on an entity. This component could be used by nodes, links, and networks. This simple example shows how components add capabilities to entities and how entities of varying types can have similar capabilities. The beauty of this model is that it can be expanded to add other components for enhanced functionality and similar capabilities between entities.

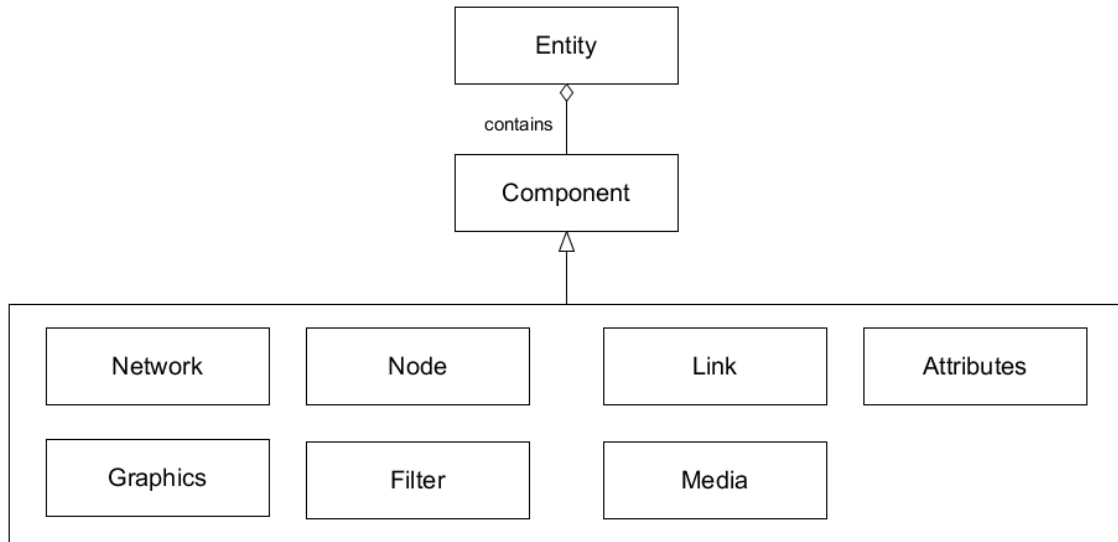


Figure 24: Entity Component Model (ECM)

4.1.2 Attribute Component, Processing, and Visualization Model

KVN has an attribute system that allows for user definable data with processing and visualization. The model is based around the “attributes” component from Figure 24. This is one of the most important components as it allows for user definable data that helps to describe an entity. Figure 25 lays out the model in a logical format. Attributes are defined by a name, value and Multipurpose Internet Media Extensions (MIME) type. A MIME type is a “standard that indicates the nature and format of a document, file, or assortment of bytes” (“MIME types,” 2018). It is used in the world wide web to understand content, so it can be properly processed. In KVN, MIME types are used to classify attributes and provide appropriate functionality based on their content. Functionality is provided through the use of attribute processors that have a MIME type and process attributes of similar MIME types. For example, image processors process images, video processors process video, document processors process documents, etc. MIME types can be custom with appropriate processors to provided other forms of functionality with in a KVN. The attribute view and media view are the final parts to the model. The attribute view ties the entire model together. It allows for the addition, editing, and visualization of

attributes in a list format. It also has a list of attribute processors. A user would add text, image, video, document, and other types of attributes to an entity using the attribute view. The attribute view would display the attribute and send it to the appropriate processor. The processor would then build a media object according to its MIME type and send it to the media view. The media view provides a place for the visualization of media and allows a user to cycle through available media and optionally popup the media for a larger view. The media is not just for images, video, and documents. It can also support sound and other forms of media such as maps, which is determined by the attribute processors provided.

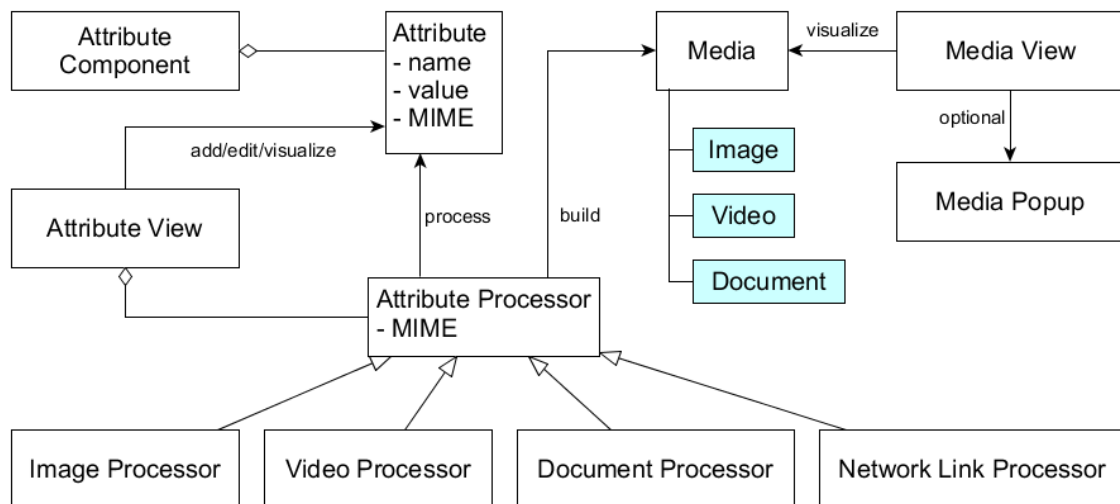


Figure 25: Attribute Component, Processing & Visualization

4.1.3 Network Link Model

KVN allows for the representation of subnetworks and other forms of links between networks using an attribute, a custom MIME type and an associated processor. The functionality is provided in KVN's network link model which is laid out in Figure 26. The attribute is a network link belonging to an entity and links to another network. The MIME type allows for the identification of the attribute as a network link and serves as an identifier that it has to be processed by the network link processor. The network link processor provides the functionality

to both create the link and initiate when and how the transition occurs. Figure 26 shows network “A” containing an entity, that has a network link attribute contained through its “attributes” component, that is linked to network “B.” The network link processor helps to create the network link, and allows the transition from network “A” to network “B.”

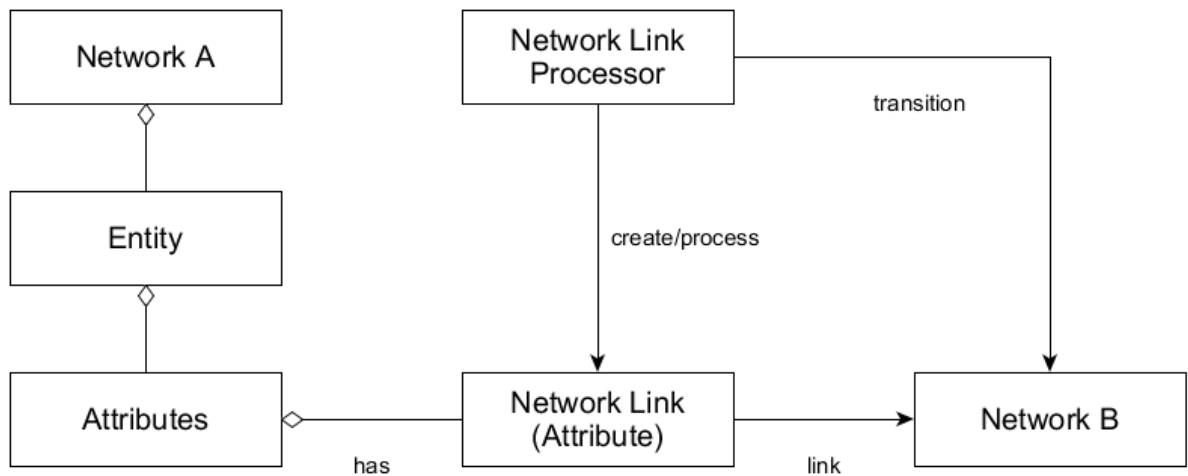


Figure 26: Network Link Model

4.1.4 Network Database Model

KVN must be able to represent, persist and provide query capabilities for networks. This functionality is provided in KVN’s network database model as laid out in Figure 27. Networks are composed of entities that are stored in a database. As information in entities change, persistent storage ensures its state will be persisted for later use. Since attributes can also represent media, the database must provide appropriate storage for multimedia. The query processor allows a user of the system to query the database. The main focus of network queries is to search the “attributes” component for user defined data. Other parts of an entity can be searched if so desired. The results produced by queries are either attributes or entities which can then be used for specific purposes. Searching can be done for a specific entity, extensively within an entity, or the entire network for a range of entities.

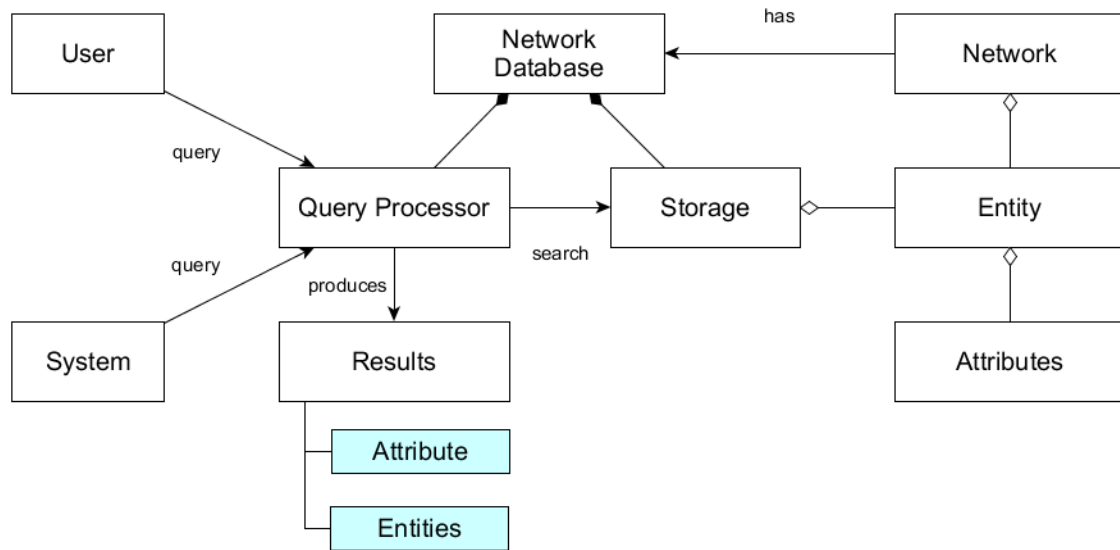


Figure 27: Network Database Model

4.1.5 Network Filter Model

Network filtering is the process of separating a network using a criteria to query the network database, and mixing it with visualization data to give an enhanced view of the network. It allows one to show how information within an entity is related and how entities within a network are related. The entire process is laid out in Figure 28. A filter is composed of multiple criteria. A criteria is composed of a query string made up using both discrete and continuous values, along with visualization data for how the criteria should be visualized. The query string serves as input to query the network database for entities. The results along with the criteria visualization data serves as input to a renderer, which are used to produce a filtered network. The filtered network can contain nodes and links visualized in such a way that allows them to have multiple classifications. Multiple classifications are a way to show how information is related to each other within an entity by through proper visualization. For example, a network filter can be used to show male entities one color, females entities using another color, and entities defined as both male and female using another or both colors. This requires the rendering of nodes using pie charts and links using multiple line segments.

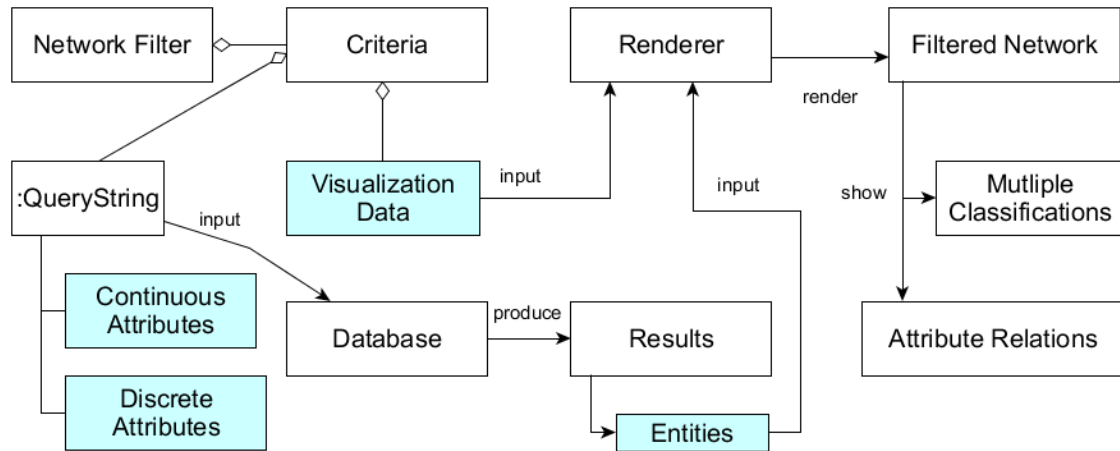


Figure 28: Network Filter Model

4.2 KVN Use Case Model

The conceptual design of KVN played a major role in driving the use case model along with the modules in Figure 20. For each module high level requirements were developed from evaluating the conceptual model along with an idea of the module's intended purpose. The high level requirements for each module were further expanded using use case scenarios to develop detail requirements. The following shows an example of this process for the network filter module. Figure 29 shows the network filter use cases. It contains high level requirements on what functionality the filter should support. To give an example of how each requirement was further developed, the use case "Activate Filter" will be used. The use case was taken and expanded upon to give an idea of the detailed steps required to activate a filter. To give a brief overview, when the filter is applied, the network is queried, results are returned, the results are then published, and the results are rendered (see Table 8).

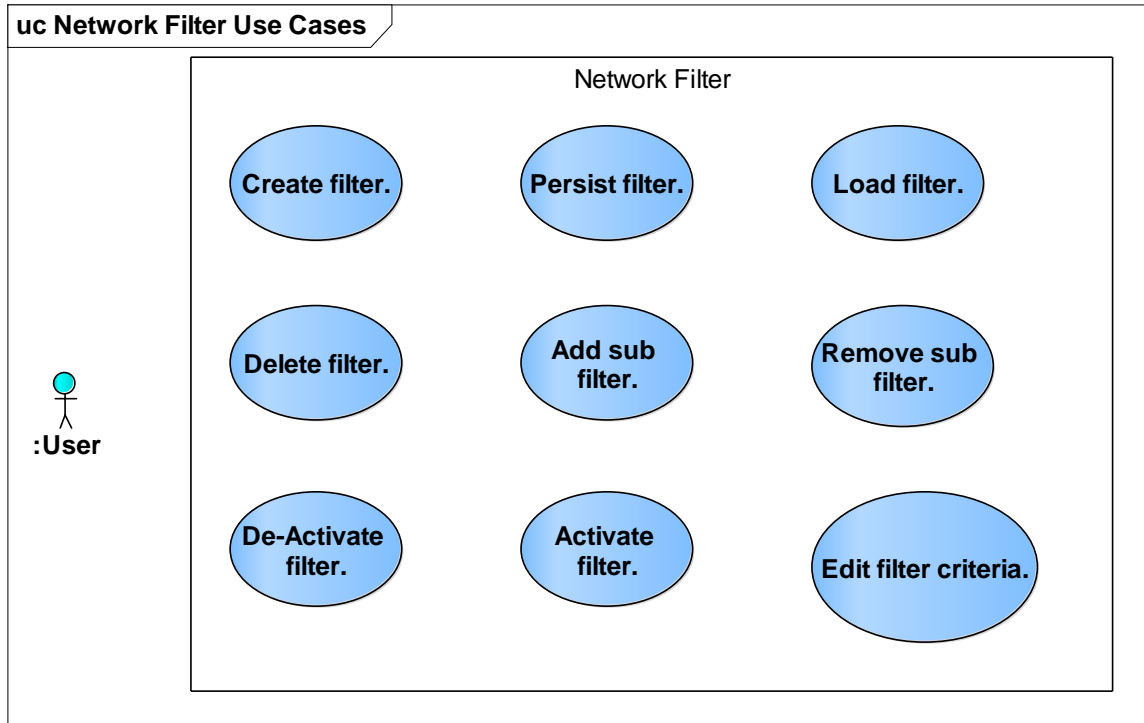


Figure 29: Network Filter Use Cases

Table 8: Use Case Scenario Activate Filter

Step	Detail
1	User applies network filter.
2	Network filter queries network database using sub filter queries.
3	Network filter builds a filter result object containing query results and visualization data.
4	Network filter publishes filter.
5	Network renderer receives the network filter message.
6	Network renderer then renders network using filter visualization data.

4.3 KVN Class Model

The use case model was used to drive development of KVN’s class diagrams (logical model) by evaluating the detailed requirements. Figure 30 gives an overview of KVN’s overall application structure in classes. This was done using structured analysis to understand and extract the structure of KVN from the detail requirements. The model contains classes for each module, data structures, and important relationships in the system. Figure 31 is a class model of the entity component system used to represent data within KVN. It serves as the building block for entities used throughout the application.

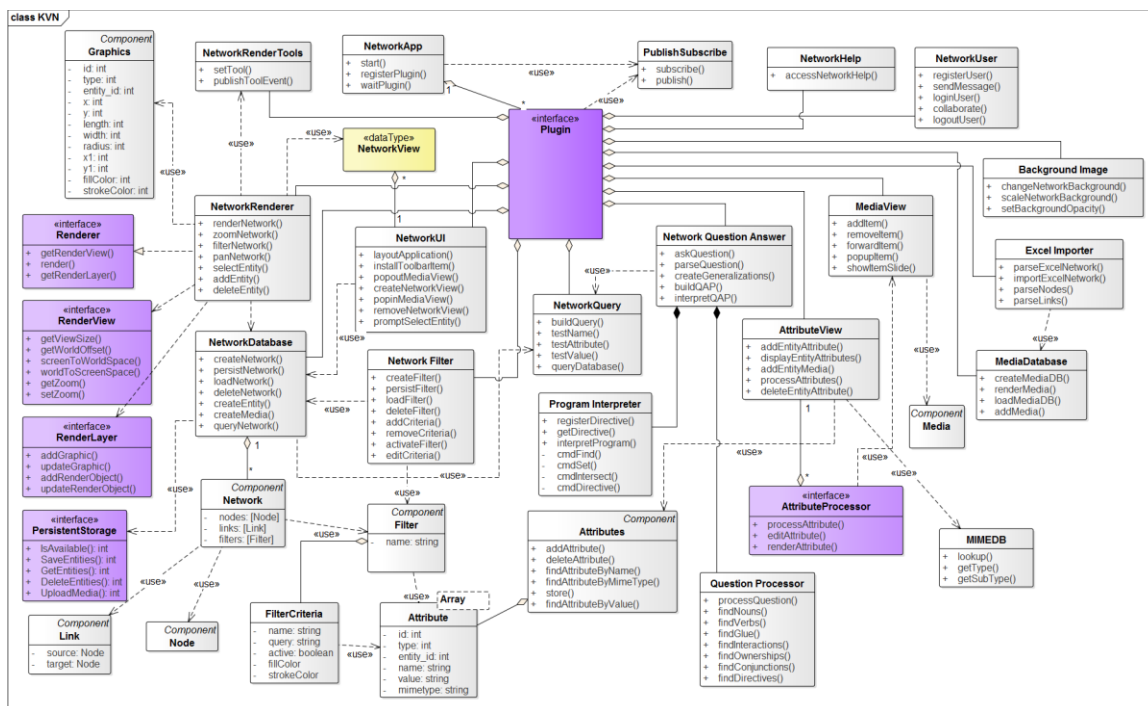


Figure 30: KVN Class Diagram

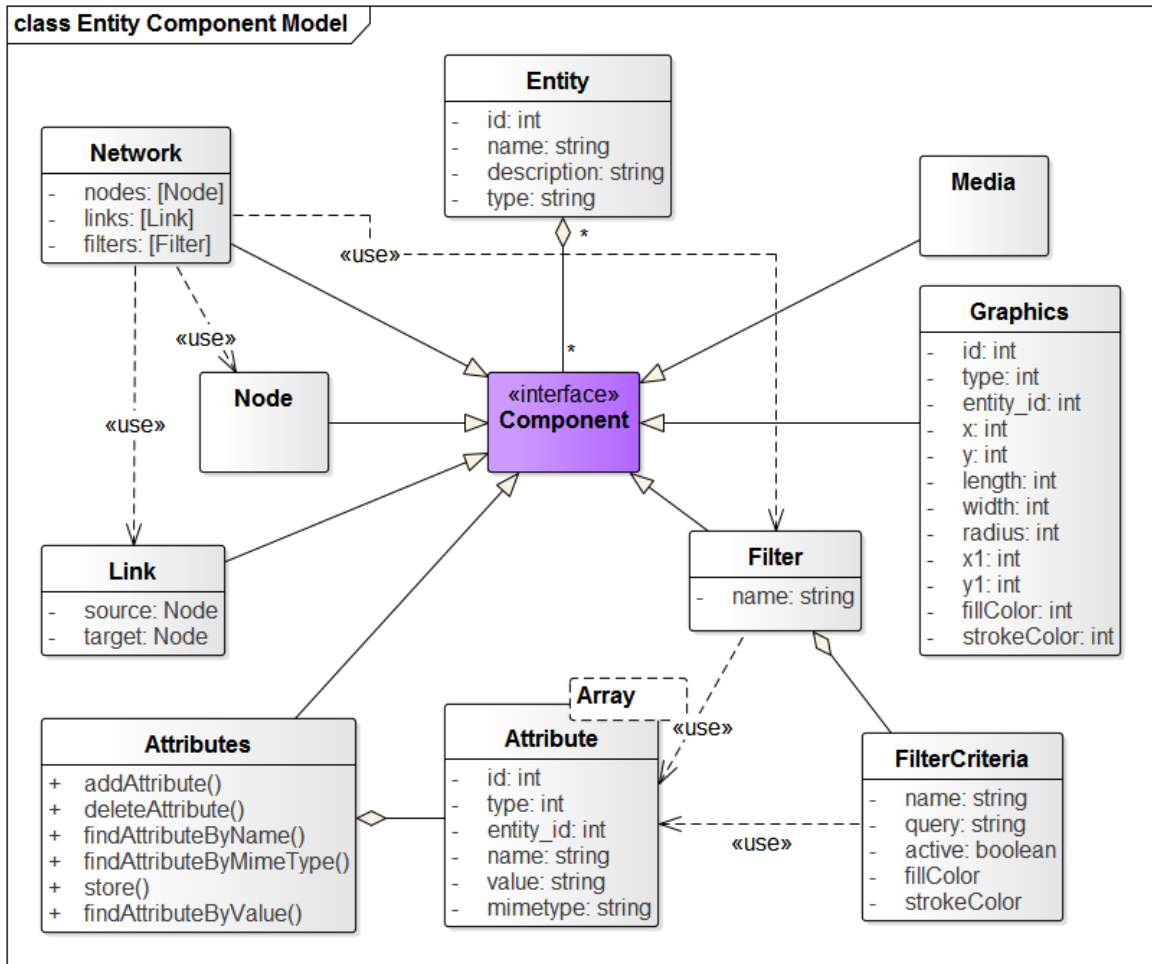


Figure 31: KVN Entity Component Model Class Diagram

4.4 KVN Application

The next step in the process was the development of the physical model, which is the application itself. The class diagram along with the use case scenarios were used to implement each module. HTML 5 and JavaScript were used to develop the client application, while PHP (Hypertext Preprocessor) and MySQL was used to implement the backend server used for persistent storage. Figure 32 shows the KVN application running in a single window. It shows the toolbar with readily available operations that can be performed in the application, the network view to display a network, the attribute view for the viewing and editing of attributes, and the

media view for the viewing of media. This is the application running in the primary window which reflects the primary window from Figure 21.

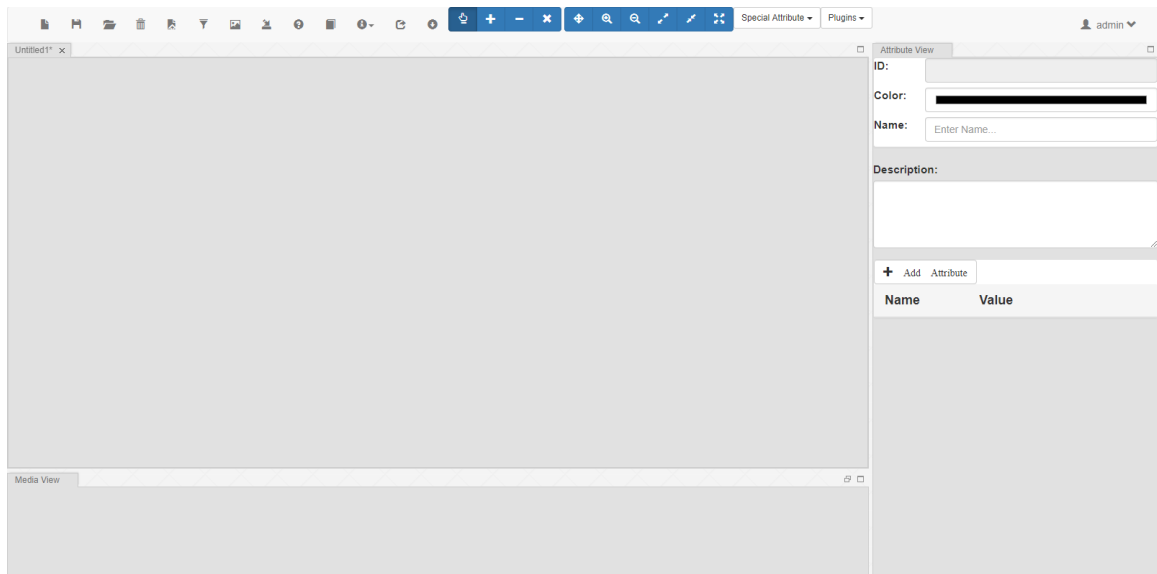


Figure 32: KVN Application Window

Figure 33 shows the KVN application running in multiple windows. In this mode the multimedia view runs in a secondary window. This mode is more suitable for when two displays are available and could be best used when giving presentations. This fully reflects the entire conceptual GUI from Figure 21.

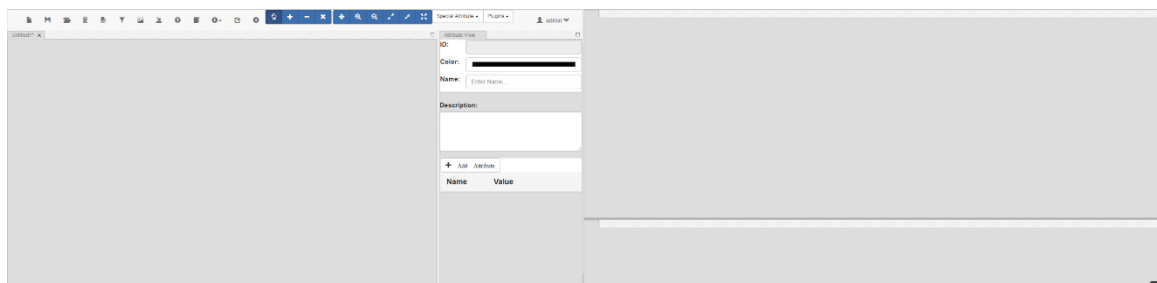


Figure 33: KVN Application Multiwindow

4.5 Example Network

Figure 34 shows the application window populated with a network. The network is displayed in the network view with 6 nodes and 6 links. Node "Ally" is selected with its

attributes visible in the attribute view. The media view displays media of the node; it shows some images and a video. These features will be further elaborated on below.

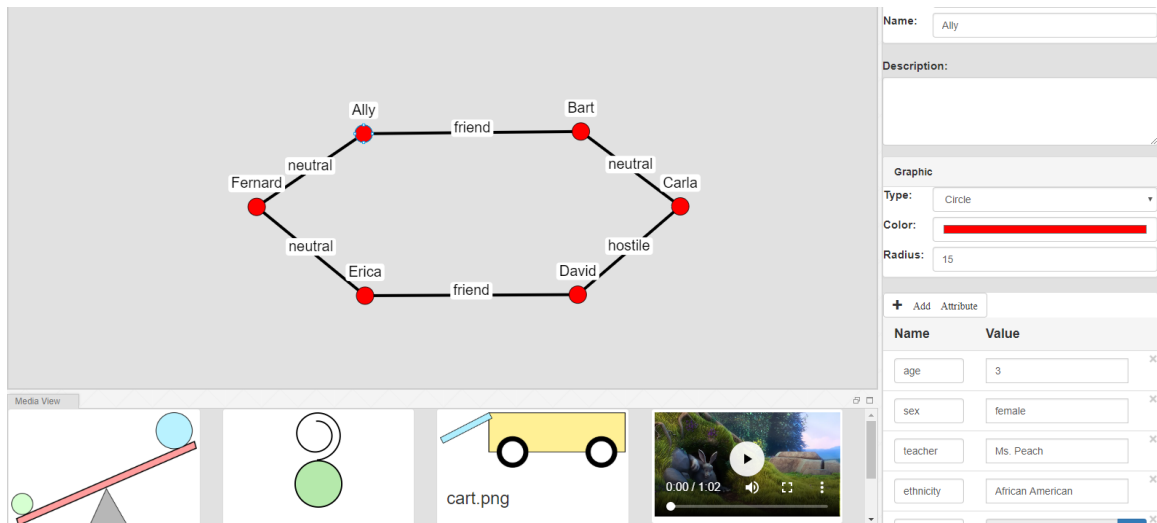


Figure 34: KVN network application with network

4.5.1 Link Attributes

In the problem definition, it was outlined that network applications often have little emphasis on attributes in relationships. KVN addresses this problem by allowing links to store attributes just like nodes. The importance of links carrying attributes will be shown in later sections. Figure 35 shows the selection of the link between “Bart” and “Carla.” In the attribute view, the link contains information about their relationship. It has an identity, age, and activity associated to it.

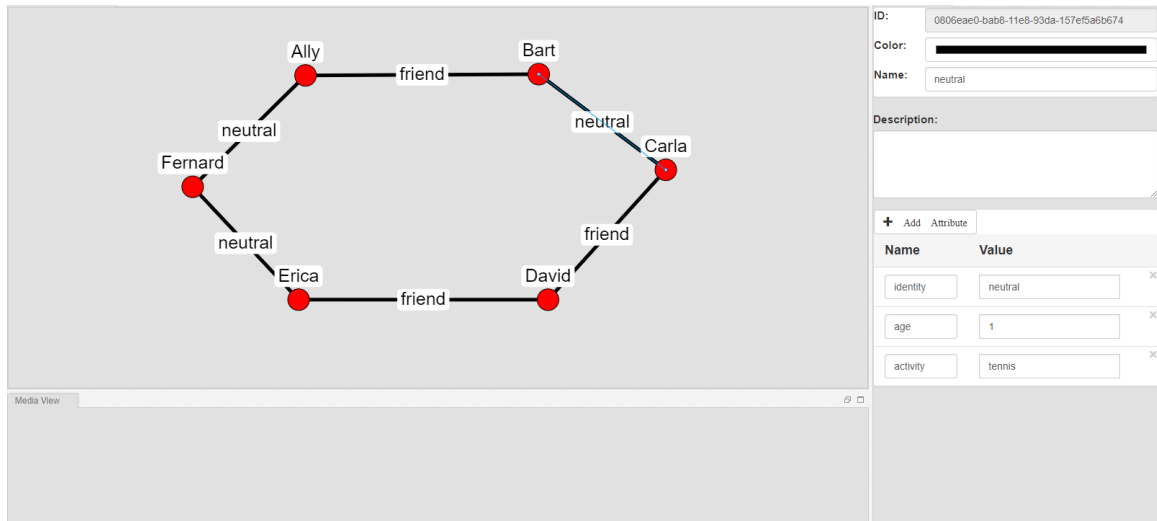


Figure 35: Attributes on links

4.5.2 Multimedia

In the problem definition, it was outlined that there is no application of multimedia to enhance information within network tools. Figure 36 shows the relationship between “Ally” and “Bart.” In the attribute view regular attributes are listed along with media attributes. In the media view the media attributes are displayed. The media view contains a pdf document, 2 images, and a video that are used to enhance the relationship between “Ally” and “Bart.” This is an example of placing greater emphasis on links in networks.

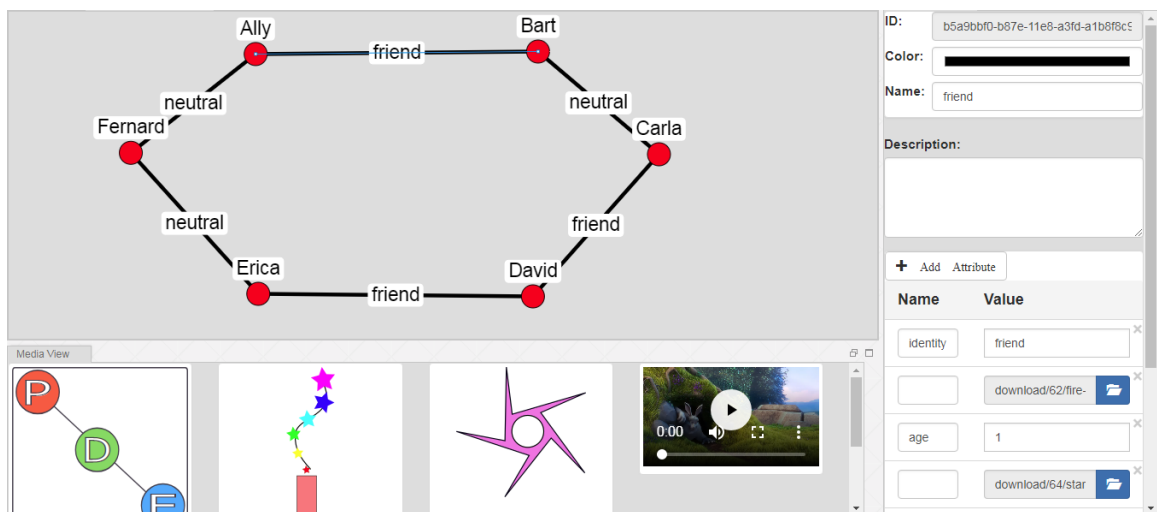


Figure 36: Selected link with media

The media view also has a popup feature to show an enlarged view of selected media. When a media is selected, the popup is displayed over the application containing the selected media. In Figure 37 the popup is displayed with a video. The popup also allows for the cycling of media using the “Previous” and “Next” buttons.

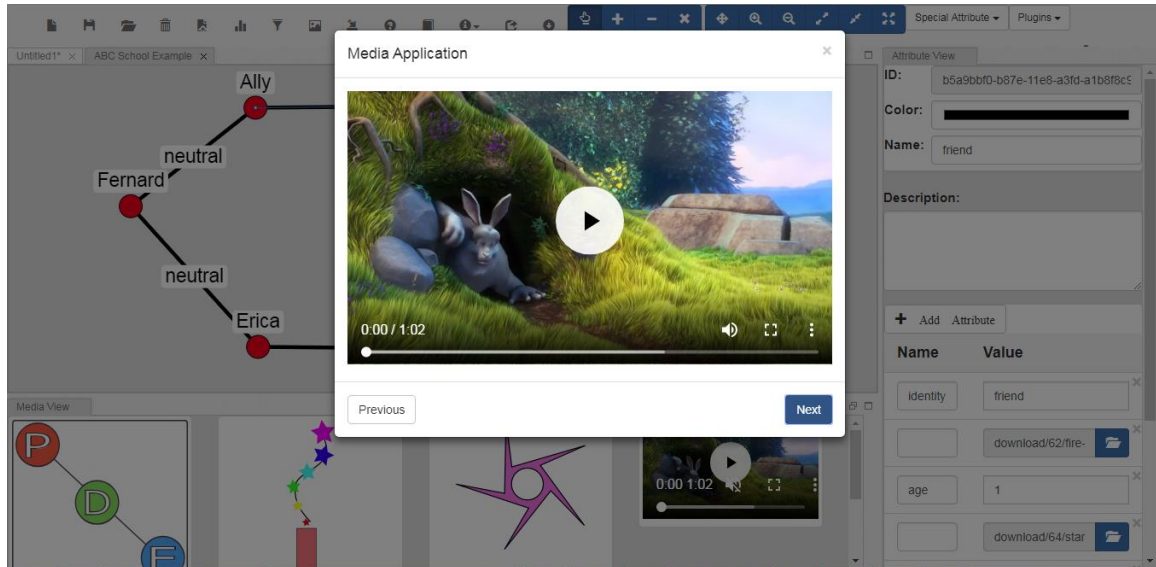


Figure 37: KVN media popup

4.5.3 Network Filter

In the problem definition, various problems of network partitions were outlined. Figure 38 gives a visual look at the network filter application used to address these problems. The figure is labeled as follows:

1. Filter name
2. Filter menu to create, load and save filters
3. Filter option to display connections (links in node clusters/nodes in link clusters)
4. Criteria table showing active criteria, criteria names, criteria queries, fill color, stroke color, and a delete option
5. Filter commands to apply the filter, remove the filter, or just cancel the application
6. Command to add criteria to the currently loaded filter in the application

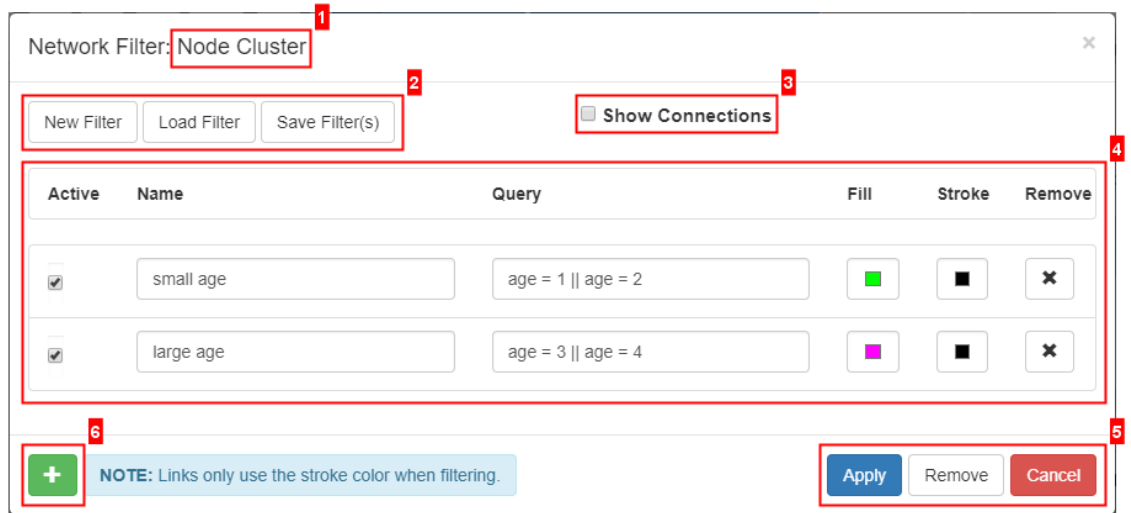


Figure 38: KVN network filter application map

4.5.3.1 Single Classification Filter

Going back to the ABC School example, the following will show how a single classification can be used to create a network partition. Figure 39 shows the filter for classification by sex where males are colored blue and females are colored purple. Figure 40 shows the network partition created by this filter. It shows 3 nodes colored blue which represent males and 3 nodes colored purple which represent the females.

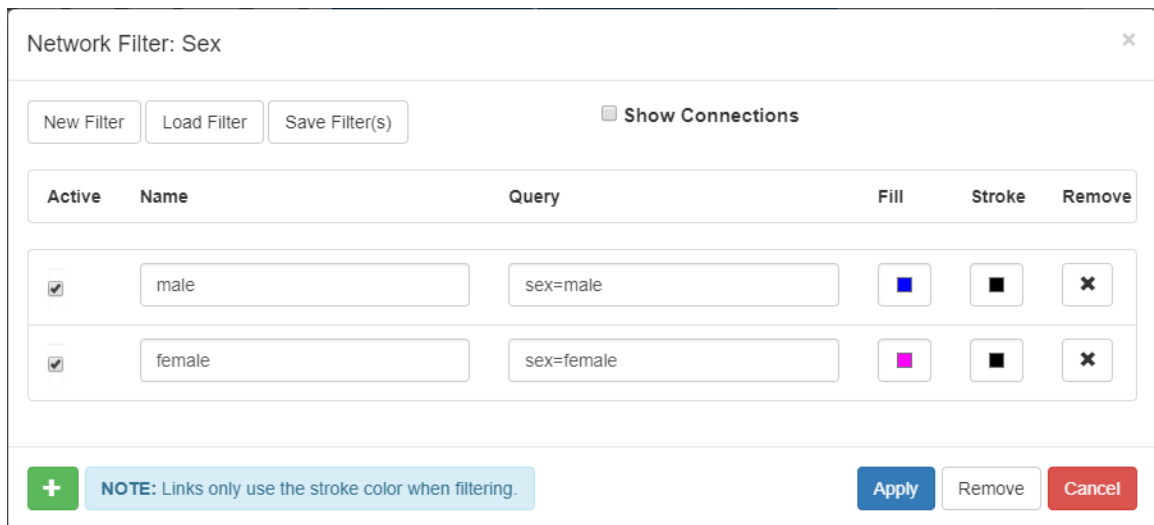


Figure 39: ABC School sex filter

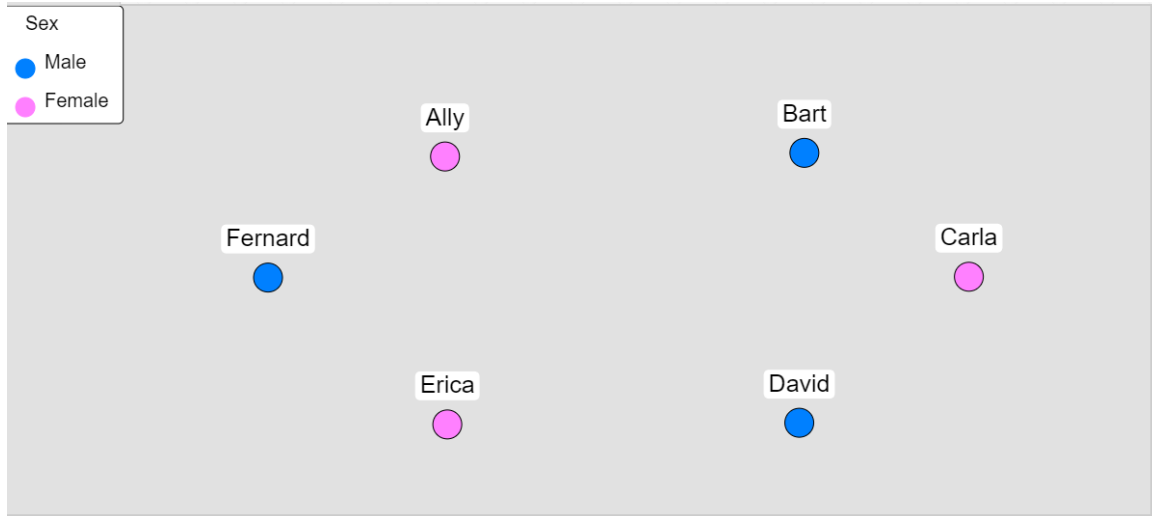


Figure 40: ABC School sex filter network

Figure 41 shows the filter for classification by teacher where students in Ms. Apple's class are colored red, and students in Ms. Peach's class are colored purple. Figure 42 shows the network partition created by this filter. There are 3 students in Ms. Apple's class and 3 students in Ms. Peach's class.

Network Filter: Teacher ✕

Show Connections

Active	Name	Query	Fill	Stroke	Remove
<input checked="" type="checkbox"/>	<input type="text" value="Ms. Apple"/>	<input type="text" value='teacher = "Ms. Apple"'/>	<input type="color" value="#ff0000"/>	<input type="color" value="#000000"/>	<input type="button" value="✕"/>
<input checked="" type="checkbox"/>	<input type="text" value="Ms. Peach"/>	<input type="text" value='teacher = "Ms. Peach"'/>	<input type="color" value="#800080"/>	<input type="color" value="#000000"/>	<input type="button" value="✕"/>

NOTE: Links only use the stroke color when filtering.

Figure 41: ABC School teacher filter

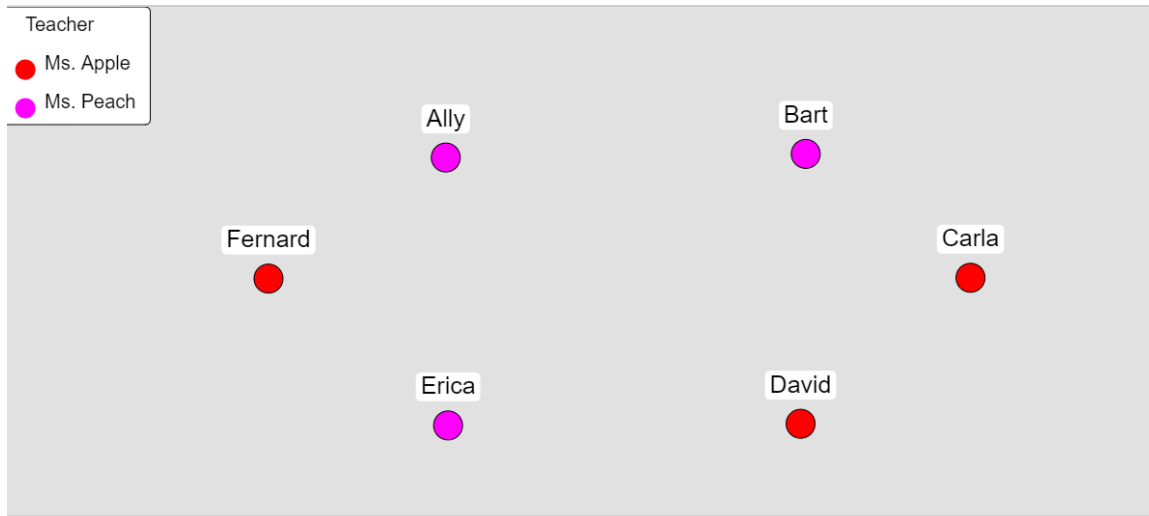


Figure 42: ABC School teacher filter network

4.5.3.2 Multiple Classification Filter

KVN also allows the partitioning on networks using multiple classifications. Figure 43 shows the filter for classification by sex and teacher. In contrast to the first 2 filters, they were single classification that had 2 parts each. By combining sex and teacher, this new filter has 4 parts. This all falls in line with Table 4. This 4 part filter uses 4 different colors. From looking at Figure 44, it shows the network displayed with 4 different colors. This verifies that KVN network filters supports multiple classifications.

There are cases where classifications can overlap. This may be a more definite term for multiple classifications. Figure 45 shows a filter to create an overlap the between filters from Figure 39 and Figure 43. When the filter is applied (Figure 46), the nodes in the network are drawn as pie charts to show that they belong to multiple classifications.

Network Filter: Sex and Teacher ✕

Show Connections

Active	Name	Query	Fill	Stroke	Remove
<input checked="" type="checkbox"/>	Male and Ms. Peach	sex=male && teacher="Ms. Peach"	■	■	<input type="button" value="✕"/>
<input checked="" type="checkbox"/>	Female and Ms. Peach	sex=female && teacher="Ms. Peach"	■	■	<input type="button" value="✕"/>
<input checked="" type="checkbox"/>	Male and Ms. Apple	sex=male && teacher="Ms. Apple"	■	■	<input type="button" value="✕"/>
<input checked="" type="checkbox"/>	Female and Ms. Apple	sex=female && teacher="Ms. Apple"	■	■	<input type="button" value="✕"/>

NOTE: Links only use the stroke color when filtering.

Figure 43: ABC School sex and teacher filter

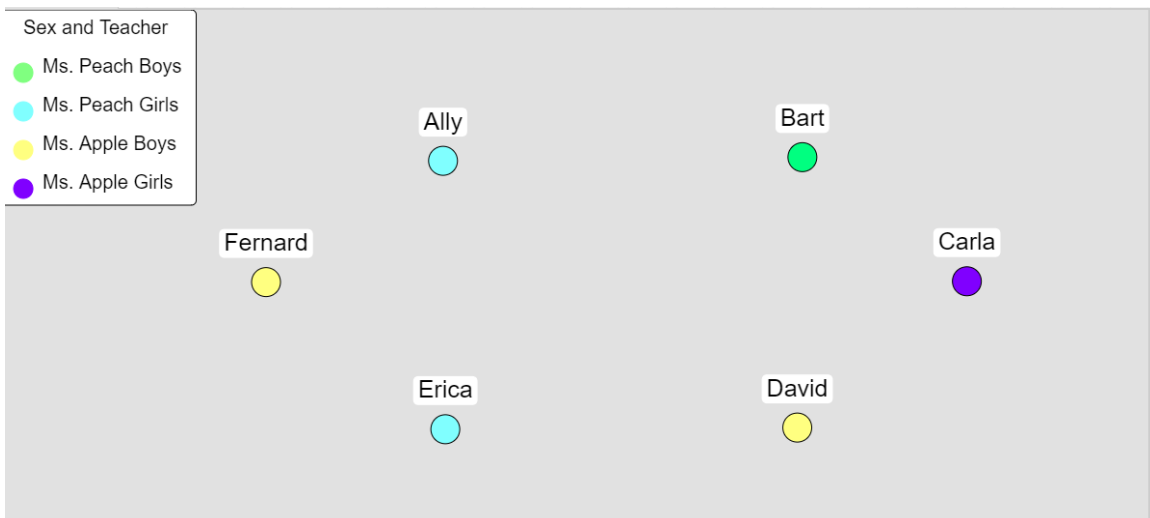


Figure 44: ABC School sex and teacher filter network

Network Filter: Sex and Teacher w Overlap x

Show Connections

Active	Name	Query	Fill	Stroke	Remove
<input checked="" type="checkbox"/>	Ms. Peach Boys	sex=male && teacher = "Ms. Peach"	■	■	<input type="button" value="x"/>
<input checked="" type="checkbox"/>	Ms. Peach Girls	sex=female && teacher = "Ms. Peach"	■	■	<input type="button" value="x"/>
<input checked="" type="checkbox"/>	Ms. Apple Boys	sex=male && teacher = "Ms. Apple"	■	■	<input type="button" value="x"/>
<input checked="" type="checkbox"/>	Ms. Apple Girls	sex=female && teacher = "Ms. Apple"	■	■	<input type="button" value="x"/>
<input checked="" type="checkbox"/>	Boys	sex=male	■	■	<input type="button" value="x"/>
<input checked="" type="checkbox"/>	Girls	sex=female	■	■	<input type="button" value="x"/>

NOTE: Links only use the stroke color when filtering.

Figure 45: ABC School sex and teacher with overlap

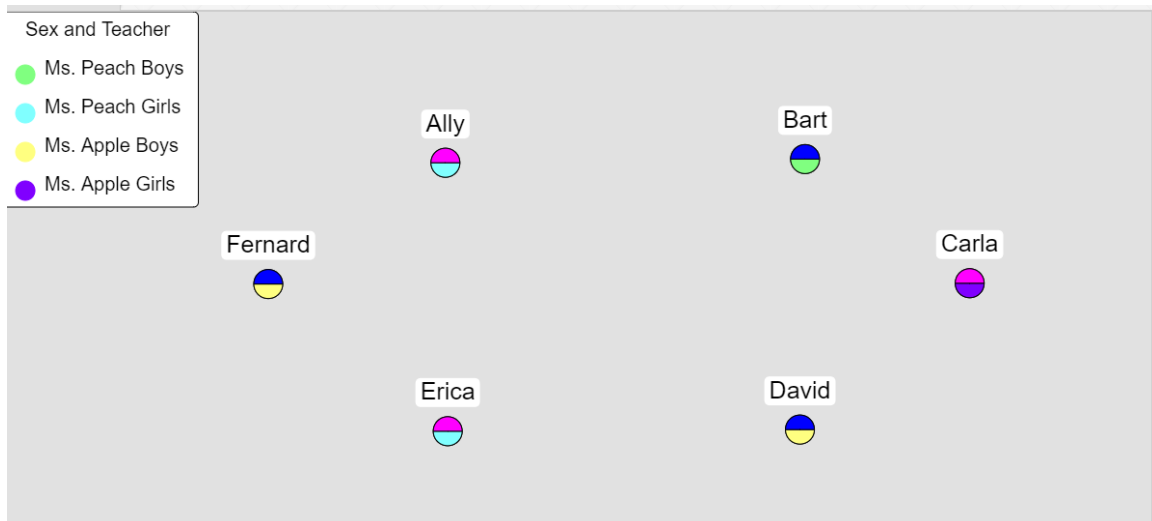


Figure 46: ABC School sex and teacher with overlap applied

4.5.3.3 Dynamic Criteria

KVN also supports network filters that use classifications as well as vectors. Figure 47 shows the application of age to create the network partition which falls in line with Table 4. By

using the criteria “age > 5” or “age <= 5,” it created 8 parts to the filter which uses 8 different colors. From looking at Figure 48, it shows the network with 5 different colors. Three of the criteria used were not met. This verifies that KVN supports the dynamic use of classifications and vectors to create network partitions.

<input checked="" type="checkbox"/>	Ms. Apple Older Girls	sex=female && teacher = "Ms. Apple"			
<input checked="" type="checkbox"/>	Ms. Apple Younger Girls	sex=female && teacher = "Ms. Apple"			
<input checked="" type="checkbox"/>	Ms. Apple Older Boys	sex=male && teacher = "Ms. Apple" >			
<input checked="" type="checkbox"/>	Ms. Apple Younger Boys	sex=male && teacher = "Ms. Apple" <			
<input checked="" type="checkbox"/>	Ms. Peach Older Girls	sex=female && "Ms. Peach" > 5			
<input checked="" type="checkbox"/>	Ms. Peach Younger Girls	sex=female && "Ms. Peach" <= 5			
<input checked="" type="checkbox"/>	Ms. Peach Older Boys	sex=male && "Ms. Peach" > 5			
<input checked="" type="checkbox"/>	Ms. Peach Younger Boys	sex=male && "Ms. Peach" <= 5			

Figure 47: ABC School dynamic criteria filter

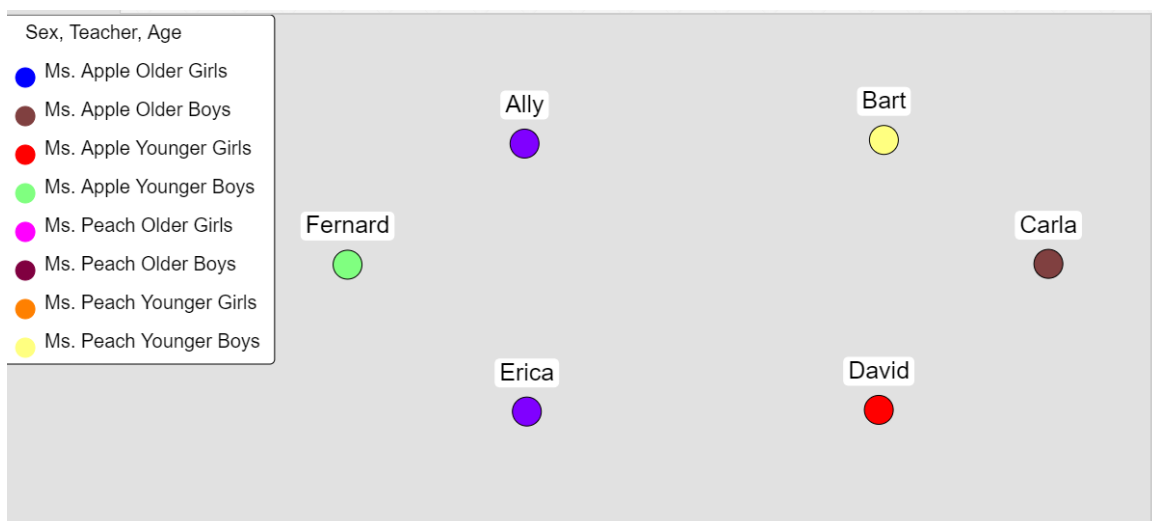


Figure 48: ABC School dynamic criteria filter network

4.5.3.4 Link Cluster

KVN also supports link clusters using a network filter which gives another example of placing greater emphasis on network links. Figure 49 shows a filter to understand the relationships among students in ABC School. Three different kinds of relationships are used; friendly relationships are shown in blue, hostile relationships are shown in red, and neutral relationships are shown in green. Figure 50 shows the link cluster created from the filter and applied to the school network. From observing the figure, it shows that the school environment has all three types of relationships. It is possible to draw conclusions from the figure. If there are mostly blue links, then the environment could be considered friendly and safe. If there are mostly red links, then the environment could be considered hostile and dangerous. Green links presents a challenge because a neutral relationship has the potential to sway one way or the other. An environment with mostly green links or a balance of all three can be unpredictable.

Network Filter: Link Cluster ✕

Show Connections

Active	Name	Query	Fill	Stroke	Remove
<input checked="" type="checkbox"/>	friend	identity=friend identity=friendly	■	■	✕
<input checked="" type="checkbox"/>	hostile	identity=hostile	■	■	✕
<input checked="" type="checkbox"/>	neutral	identity=neutral	■	■	✕

+
NOTE: Links only use the stroke color when filtering.

Figure 49: ABC School link cluster filter

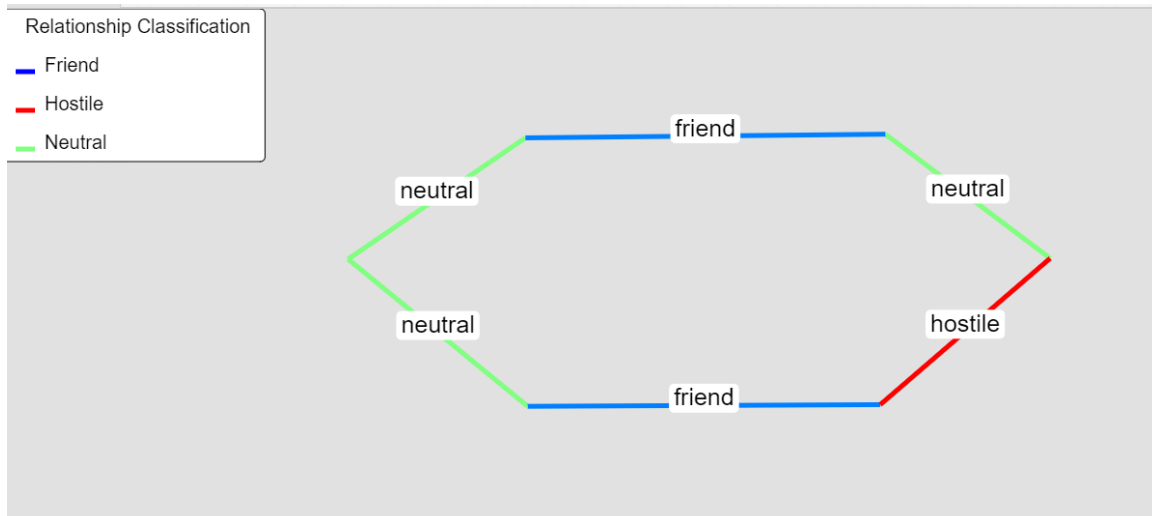


Figure 50: ABC School link cluster filter network

4.6 KVN Use Cases

KVN was used to implement a model for MCM and by MSU senior researchers for a AKA CAP project. In both examples various features provided by KVN were utilized to enhance the program or research effort. This was done to showcase the capabilities within KVN. Please see Appendix A: Morgan Community Mile and Appendix B: MSU Undergrad Senior Project Research for the example write ups.

4.7 Deliverables

There are three items delivered as a result of this research. A conceptual model of the KVN, an EA archive of the logical object oriented model and a physical implementation of KVN which is the web application itself. Some use cases were also developed to help prove KVN's unique capabilities.

4.7.1 Conceptual Model

The conceptual model lays out important concepts of KVN's design. These are provided as diagrams that lay the core foundation of KVN's unique capabilities. They serve as the core

knowledge that can be implemented in other programs. These diagrams are provided in the EA archive.

4.7.2 Logical Model

The logical model built from the analysis of the conceptual model is provided in its entirety as an object oriented model containing the artifacts of analysis. Artifacts such as the class diagrams for all major components of KVN and their relationships to each other. It lays out KVN core architecture, its support classes, and their relationships as well as all of the critical connections.

4.7.3 Physical Model

The physical model which was developed from the logical model is delivered as the KVN web application. It satisfies the major objectives of this research, which was to deliver a network tool that supports the storage of networks to a database, same capabilities at nodes and links, the viewing of multimedia data within networks and flexible network filters. It is important to note that KVN uses a database to store its networks which allow for persistence and query capabilities. Having the same capabilities for nodes and links stresses the importance of relationships in a network and the reason why they should be given similar modeling capabilities. KVN allows for the storage and viewing of multimedia within its networks to enhance information understanding and dissemination. The network filter provides dynamic filtering of networks that allows multiple classifications, multiple continuous values, and link clusters.

4.8 Future Work

While KVN brings a number of unique ideas to the network science community, there are some places where improvements and additional features can be made as future work. First, the addition of a collaboration and question answer module are two improvements for

enhancing KVN with respect to sharing and information dissemination. Sharing and collaboration would allow users to share and collaborate on networks, providing realtime and nonrealtime feedback in terms of network editing and annotations. A question answer module would allow a user to search a network for information using natural language constructs to extract specific information. This is similar to network filtering but with a more general context and different output.

Second, multiagent systems can be implemented into KVN for the simulation of entities in a network. Multiagent systems are “typically distributed systems in which several distinct components, each of which is an independent problem-solving agent come together to form some coherent whole” (D’Inverno & Luck, 2010). KVN’s ECM bear similar capabilities to the SMART (Structured and Modular Agents and Relationship Types) framework provided by D’Inverno and Luck. By implementing SMART into KVN it can provide some unique capabilities such as real time networks where entities are based on and continuously updated from a real time system, network simulation to understand how entities behave in a network, network optimization based on resources contained in entities and general computations used to solve some abstract problem.

Finally, the addition of network statistics would allow KVN to become a more general purpose tool. Currently, KVN does not provide any useful network statistics. Network statistics would be beneficial in understanding and solving network problems in abstract ways. It is also possible that network statistics could be used to aid multiagent systems in KVN that may open up new possibilities.

5 References

- Arora, M., Kanjilal, U., & Varshney, D. (2010). *Using Social Network Analysis for Information Retrieval for Mapping Information and Knowledge Flows* (SSRN Scholarly Paper No. ID 2162039). Retrieved from Social Science Research Network website: <https://papers.ssrn.com/abstract=2162039>
- Auber, D., Archambault, D., Bourqui, R., Delest, M., Dubois, J., Lambert, A., ... Vallet, J. (2017). TULIP 5. In R. Alhajj & J. Rokne (Eds.), *Encyclopedia of Social Network Analysis and Mining* (pp. 1–28). https://doi.org/10.1007/978-1-4614-7163-9_315-1
- Auber, D., Archambault, D., Bourqui, R., Lambert, A., Mathiaut, M., Mary, P., ... Melançon, G. (2012). *The Tulip 3 Framework: A Scalable Software Library for Information Visualization Applications Based on Relational Data* (Research Report No. RR-7860; p. 31). Retrieved from INRIA website: <https://hal.inria.fr/hal-00659880>
- Barabási, A.-L. (2012). The network takeover. *Nature Physics*, *8*, 14–16.
- Bastian, M., Heymann, S., & Jacomy, M. (2009). *Gephi: An Open Source Software for Exploring and Manipulating Networks*. International AAAI Conference on Weblogs and Social Media.
- Batagelj, V., & Mrvar, A. (2018). Program Package Pajek / PajekXXL (Version 5.06a). Retrieved from <http://mrvar.fdv.uni-lj.si/pajek/>
- BNIA: Baltimore Neighborhood Indicators Alliance | Providing reliable, actionable quality of life indicators for Baltimore's neighborhoods. (2018). Retrieved January 21, 2019, from <https://bniajfi.org/>
- Börner, K. (2007). Making Sense of Mankind's Scholarly Knowledge and Expertise: Collecting, Interlinking, and Organizing What We Know and Different Approaches to Mapping

(Network) Science. *Environment and Planning B: Planning and Design*, 34(5), 808–825.

<https://doi.org/10.1068/b3302t>

Chen, J., Fagnan, J., Goebel, R., Rabbany, R., Sangi, F., Takaffoli, M., ... Zaiane, O. (2010).

Meerkat: Community Mining with Dynamic Social Networks. *2010 IEEE International Conference on Data Mining Workshops*, 1377–1380.

<https://doi.org/10.1109/ICDMW.2010.40>

Copeland, L. (n. (2003, January 28). State-Transition Diagrams. Retrieved January 13, 2019, from

StickyMinds website: <https://www.stickyminds.com/article/state-transition-diagrams>

Darus. (2005). *apache2dot*. Retrieved from <http://www.chaosreigns.com/code/apache2dot/>

Desale, D. (2015, June). Top 30 Social Network Analysis and Visualization Tools. Retrieved

January 13, 2019, from <https://www.kdnuggets.com/2015/06/top-30-social-network-analysis-visualization-tools.html>, <https://www.kdnuggets.com/2015/06/top-30-social-network-analysis-visualization-tools.html>

D’Inverno, M., & Luck, M. (2010). *Understanding agent systems* (2nd rev. and extended ed).

Berlin: Springer.

Ellson, J., Gansner, E. R., Koutsofios, E., North, S. C., & Woodhull, G. (2004). Graphviz and

Dynagraph—Static and Dynamic Graph Drawing Tools. In M. Jünger & P. Mutzel (Eds.), *Graph Drawing Software* (pp. 127–148). https://doi.org/10.1007/978-3-642-18638-7_6

Enterprise Architect (Version 12) [Java]. (2015). Retrieved from

<https://sparxsystems.com/products/ea/index.html>

Freeman, L. C. (2004). *The development of social network analysis: a study in the sociology of science*. Vancouver, BC : North Charleston, S.C: Empirical Press ; BookSurge.

- Gansner, E., & North, S. (1997). An Open Graph Visualization System and Its Applications to Software Engineering. *Software Practice and Experience*, 30.
[https://doi.org/10.1002/1097-024X\(200009\)30:11<1203::AID-SPE338>3.0.CO;2-N](https://doi.org/10.1002/1097-024X(200009)30:11<1203::AID-SPE338>3.0.CO;2-N)
- Gansner, E. R., Koren, Y., & North, S. (2005). Graph Drawing by Stress Majorization. In: Pach J. (Eds) *Graph Drawing. GD, 2004*. (3383.).
- Gephi: The Open Graph Viz Platform (Version 0.9.2). (2017). Retrieved from <https://gephi.org/>
- Graphviz: Graph Visualization Software (Version 2.38). (2014). Retrieved from <https://www.graphviz.org/>
- Hanneman, R. A., & Riddle, M. (2005). *Introduction to Social Network Methods*. Riverside, CA: University of California.
- Heesch, D. (2018). Doxygen: Main Page. Retrieved January 13, 2019, from <http://www.doxygen.nl/>
- Herr, B. W., Huang, W., Penumarthy, S., & Börner, K. (2006). Designing highly flexible and usable cyberinfrastructures for convergence. *Annals of the New York Academy of Sciences*, 1093, 161–179. <https://doi.org/10.1196/annals.1382.013>
- Home :: Bugzilla :: bugzilla.org. (2019, January). Retrieved January 13, 2019, from <https://www.bugzilla.org/>
- Jia, L., Fu, W., Jia, R., Wu, L., Li, X., Jia, Q., & Zhang, H. (2018). Identification of potential key protein interaction networks of BK virus nephropathy in patients receiving kidney transplantation. *Scientific Reports*, 8(1), 5017. <https://doi.org/10.1038/s41598-018-23492-2>
- Kalamaras, D. (2018). SOcNETV: SOCIAL NETWORK ANALYSIS AND VISUALIZATION SOFTWARE (Version 2.4). Retrieved from <https://socnetv.org/>

- Kamada, T., & Kawai, S. (1989). An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1), 7–15. [https://doi.org/10.1016/0020-0190\(89\)90102-6](https://doi.org/10.1016/0020-0190(89)90102-6)
- Lewis, T. G. (2009). *Network Science: Theories and Applications*. Hoboken: John Wiley & Sons.
- Liang, Y., Qiao, L., Peng, X., Cui, Z., Yin, Y., Liao, H., ... Li, L. (2018). The chemokine receptor CCR1 is identified in mast cell-derived exosomes. *American Journal of Translational Research*, 10(2), 352–367.
- Martin, A. (2007, November 11). Entity Systems are the future of MMOG development Part 2. Retrieved January 13, 2019, from <http://t-machine.org/index.php/2007/11/11/entity-systems-are-the-future-of-mmog-development-part-2/>
- MIME types. (2018). Retrieved January 13, 2019, from MDN Web Docs website: https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types
- Moradifard, S., Hoseinbeyki, M., Ganji, S. M., & Minuchehr, Z. (2018). Analysis of microRNA and Gene Expression Profiles in Alzheimer's Disease: A Meta-Analysis Approach. *Scientific Reports*, 8(1), 4767. <https://doi.org/10.1038/s41598-018-20959-0>
- Mrvar, A., & Batagelj, V. (2016). Analysis and visualization of large networks with program package Pajek. *Complex Adaptive Systems Modeling*, 4(1), 6. <https://doi.org/10.1186/s40294-016-0017-8>
- National Research Council [U.S.] (Ed.). (2005). *Network science / Committee on Network Science for Future Army Applications, Board on Army Science and Technology, Division on Engineering and Physical Sciences, National Research Council*. Washington, DC: National Academies Press.
- Neiryneck, T., & Borner, K. (2007). Representing, Analyzing, and Visualizing Scholarly Data in Support of Research Management. *2007 11th International Conference Information Visualization (IV '07)*, 124–129. <https://doi.org/10.1109/IV.2007.94>

- Nepusz, T., Yu, H., & Paccanaro, A. (2012). Detecting overlapping protein complexes in protein-protein interaction networks. *Nature Methods*, 9(5), 471–472.
<https://doi.org/10.1038/nmeth.1938>
- Nooy, W. de, Mrvar, A., & Batagelj, V. (2005). *Exploratory social network analysis with Pajek*. New York: Cambridge University Press.
- PubGene: Connect–Discover–Share. (2019). Retrieved January 13, 2019, from
<https://www.pubgene.com/>
- pubmeddev. (2019). PubMed NCBI. Retrieved January 13, 2019, from
<https://www.ncbi.nlm.nih.gov/pubmed/>
- Shah, P., & Mehta, R. (2017). Comparative Analysis of Social Network Analysis and Visualisation Tools. *International Journal of Scientific Research in Science, Engineering and Technology*, 3(1), 508–513.
- Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., ... Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11), 2498–2504. <https://doi.org/10.1101/gr.1239303>
- Shelly, G. B., & Rosenblatt, H. J. (2012). *Systems analysis and design* (9th ed). Boston: Course Technology Cengage Learning.
- Takaffoli, M., Sangi, F., Fagnan, J., & Zaïane, O. R. (2010). *A Framework for Analyzing Dynamic Social Networks*.
- The DOT Language. (2019, January). Retrieved January 13, 2019, from
<https://www.graphviz.org/doc/info/lang.html>
- Tsang, C. H., Lau, C. S., & Leung, Y. K. (2005). *Object-Oriented Technology: From Diagram to Code with Visual Paradigm for UML*. Singapore: McGraw-Hill.

- UML Activity Diagram Tutorial | Lucidchart. (2019). Retrieved January 13, 2019, from <https://www.lucidchart.com/pages/uml-activity-diagram?a=1>
- Wambeke, B., Liu, M., & Hsiang, S. (2012). Using Pajek and Centrality Analysis to Identify a Social Network of Construction Trades. *Journal of Construction Engineering and Management*, 138(10), 1192–1201. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0000524](https://doi.org/10.1061/(ASCE)CO.1943-7862.0000524)
- Wilson, D. (2011). *Growing the Future, Leading the World: The Strategic Plan for Morgan State University, 2011-2021*. Retrieved from http://www.morgan.edu/Documents/ABOUT/StrategicPlan/StrategicPlan2011-21_Final.pdf
- Wong, L. (2001). PIES, a protein interaction extraction system. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 520–531.

Appendix A: Morgan Community Mile

MCM was implemented in KVN to show the advantage it brings to the table in contrast to power point presentations and websites. Figure A-1 shows MCM represented in KVN. It shows the main structure where the head node is MCM. Attached to MCM are the different parts which are the staff, board, partners, projects communities, and history. Each node contains information on what that sub structure of MCM represents. However, the nodes by themselves are not useful if the bigger picture cannot be seen. Each node also contains a network link to another network that gives a bigger picture of what exactly is contained within that sub structure.

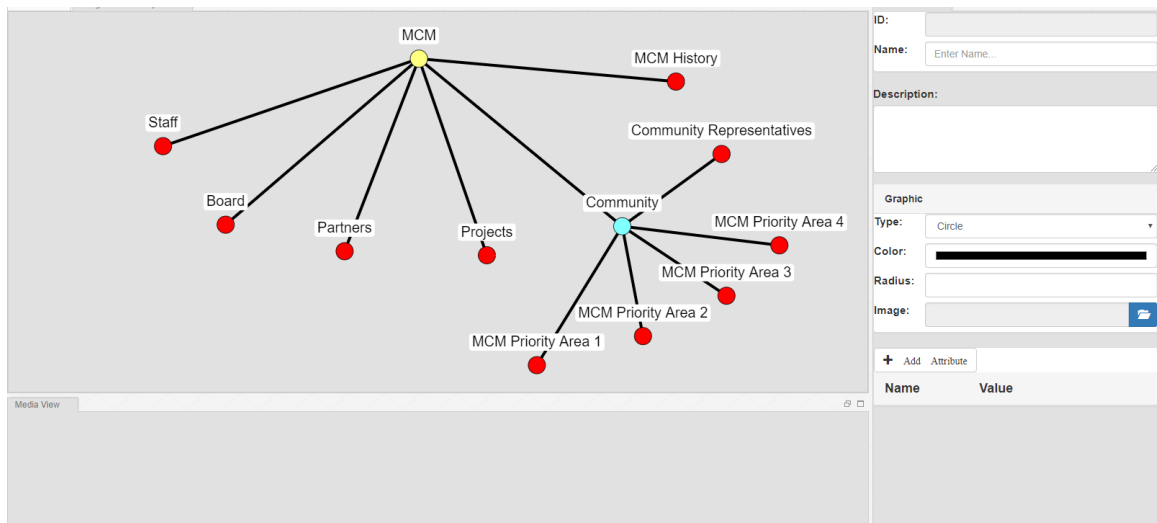


Figure A-1: MCM as a network

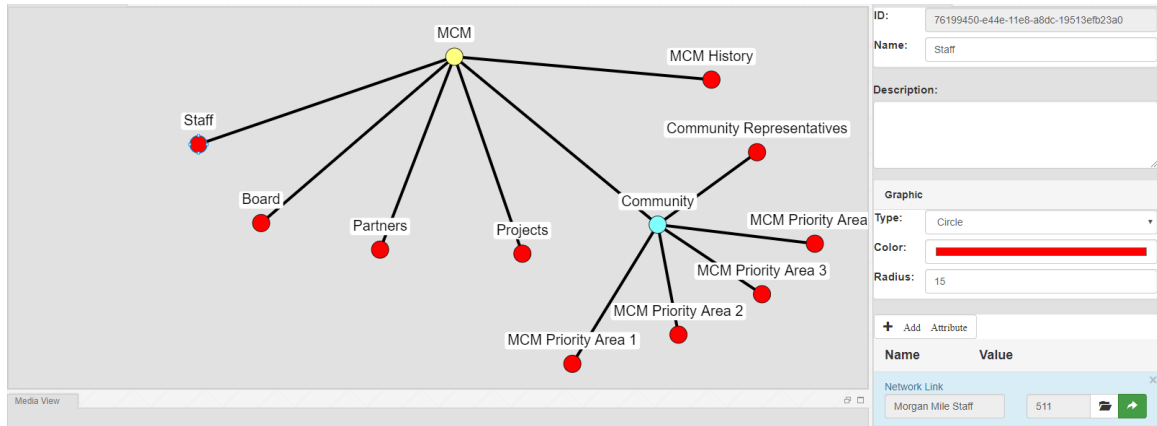


Figure A-2: MCM to Staff Network Transitions

Figure A-2 shows the staff node selected. In the attribute view there is a network link with a green arrow to travel to the staff network. When the button is selected, the staff network is opened as shown in Figure A-3. This is an example of how network links can be used in MCM to represent sub networks and travel to them. Alternatively, one can open the network alone using the applications open network tool. The staff network contains the staff members of MCM. At the top is the head node which is the executive director. At the bottom from left to right is the director, community statistician, and online consultant. With the executive director selected, the attribute view contains information about him, along with the media view showing available media. This can be done for all staff members.

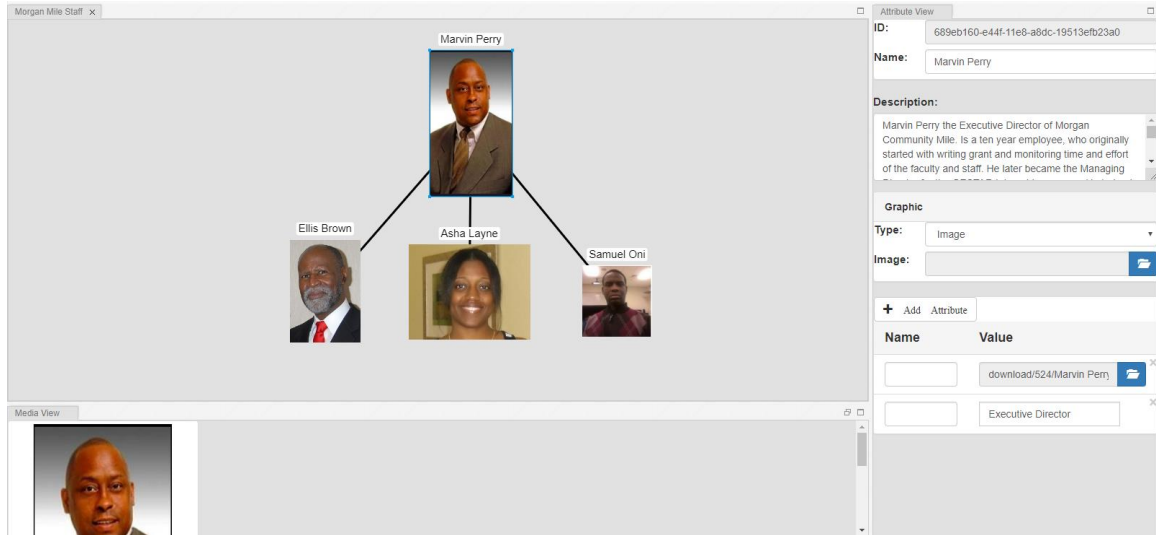


Figure A-3: MCM Staff

Figure A-4 shows a network of the board members in MCM. The head node is purple as a means of representing the board. Under the board, the network is classified according to the organizations that each board member falls under. The figure contains a legend of the organizations. There are 3 organization. The 4th is not really an organization but a classification that shows there are some board members that do not belong to any organization. By selecting a board member, one can gain further information on them and how to contact them.

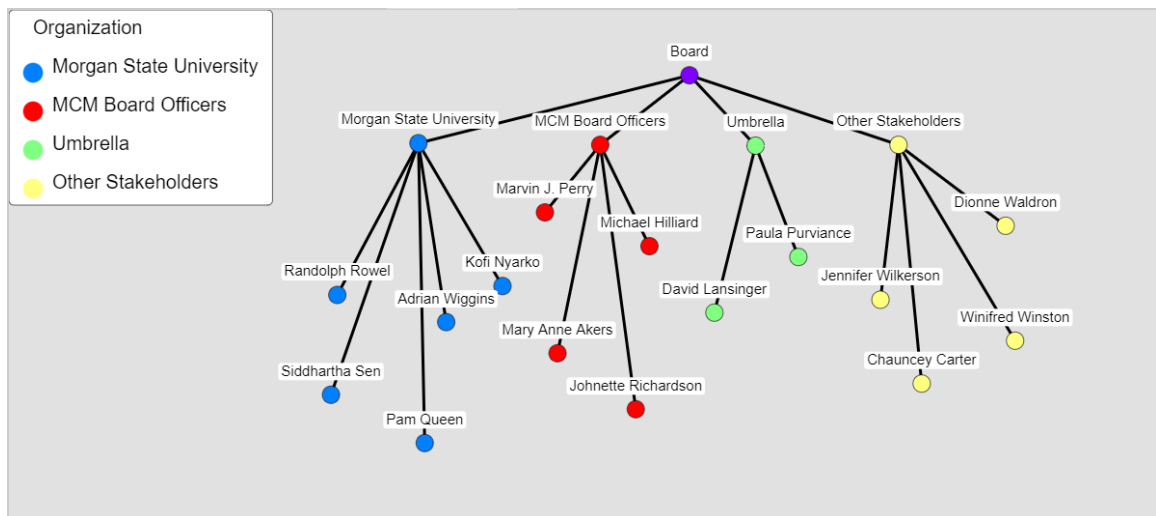


Figure A-4: MCM Board

Figure A-5 shows a network of partners involved in MCM. These partners are community organizations from each community involved in MCM. The information attached to these nodes would be information on the community organizations. The figure shows the community “Coldstream, Homestead, Montebello” selected. In the attribute view contains a link to the community website, the name of the community president, and email address. This sort of information could be stored for each community organization.

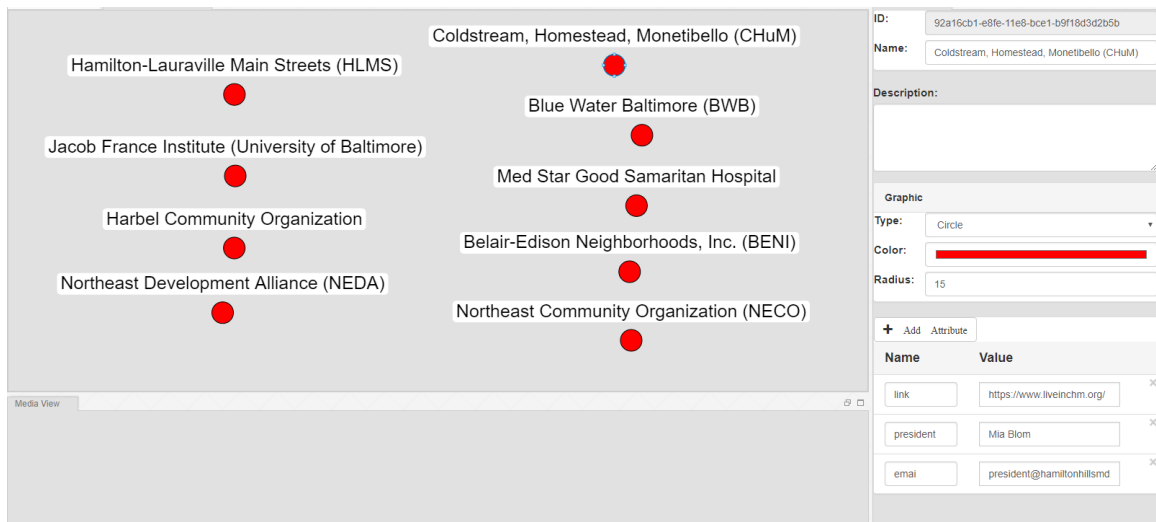


Figure A-5: MCM Partners

Figure A-6 is a network of the projects involved in MCM. It shows a total of 5 projects with the Morgan Stream Walk selected. In the description panel is a description of the project along with media about the project in the media view. This network could serve as a means of storing information of all projects along with media. From a research perspective it could serve as a means of analyzing and visualizing what has been done, what is happening, and what will eventually be taking place through the use attributes and media. Alternatively, these projects could be linked back to the communities that they affect, to track how they are affecting those communities.

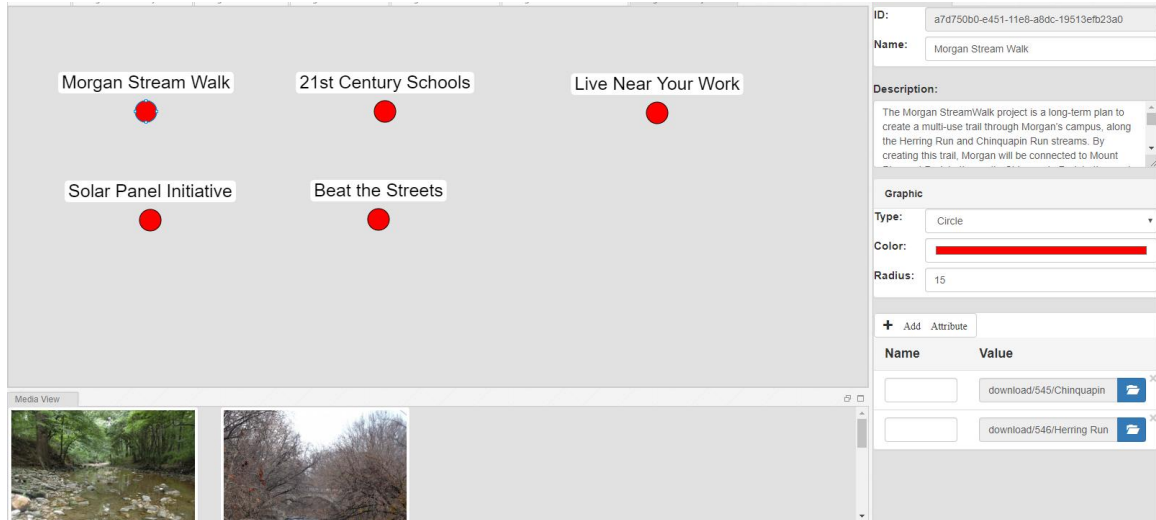


Figure A-6: MCM Projects

Figure A-7 represents one of the most important networks in MCM right beside the main network. This represents all the communities involved in MCM against a background image map of where they are located. In order to understand the power behind this network, it is important to understand how it was built. A background image from Google Earth was imported to provide the map. The communities were then manually created and placed in their appropriate location on the map. Morgan State University was also placed on the map to show its location in relation to the communities. It can be seen from the figure that MSU falls relatively in the center of the communities. Whether or not the distance is an actual mile would require a scale.

Statistics about each community from (BNIA) Baltimore Neighborhoods Indicators Alliance (“BNIA: Baltimore Neighborhood Indicators Alliance | Providing reliable, actionable quality of life indicators for Baltimore’s neighborhoods,” 2018) were then compiled into a spreadsheet. The spreadsheet was then imported into the network using the excel importer, to fill each community with its statistical data. Because there was so much information available, manually entering them into the application would be time consuming so the information was imported to save time. If one were to select a community, the statistical data would be

displayed in the attribute view. This would be good for displaying the information about a particular community as seen in Figure A-8. It shows the community “Lauraville” selected with attributes displayed in the attribute view. The list can be scrolled to view and make modifications to the information.



Figure A-7: MCM Community Network



Figure A-8: MCM Community Selection

With so much information available for each community, one may want to analyze the information in relation to itself or across the entire network. This is a perfect example using the

network filter. Figure A-9 shows the income level along with the child poverty of each community. From looking at the network itself it is difficult to understand the visualization, so the legend exists for a better understanding of what they mean. Each color in the legend is a representation of the different criteria used for the network filter. From looking at the figure it shows each community represented as a pie chart. One part represents its income and the other part its child poverty level. This shows how networks in KVN allow multiple classification of entities based on attribute values. This network shows the relation between income and poverty. The community “Lauraville” has an income level greater than \$60000 with child poverty less than 20 percent. The community “Greater Govans” has in income level from \$20000 to \$40000 with child poverty from 40 to 50 percent. The conclusion that can be drawn from this visualization is that communities with high income has lower child poverty levels, and communities with low income has higher child poverty levels.

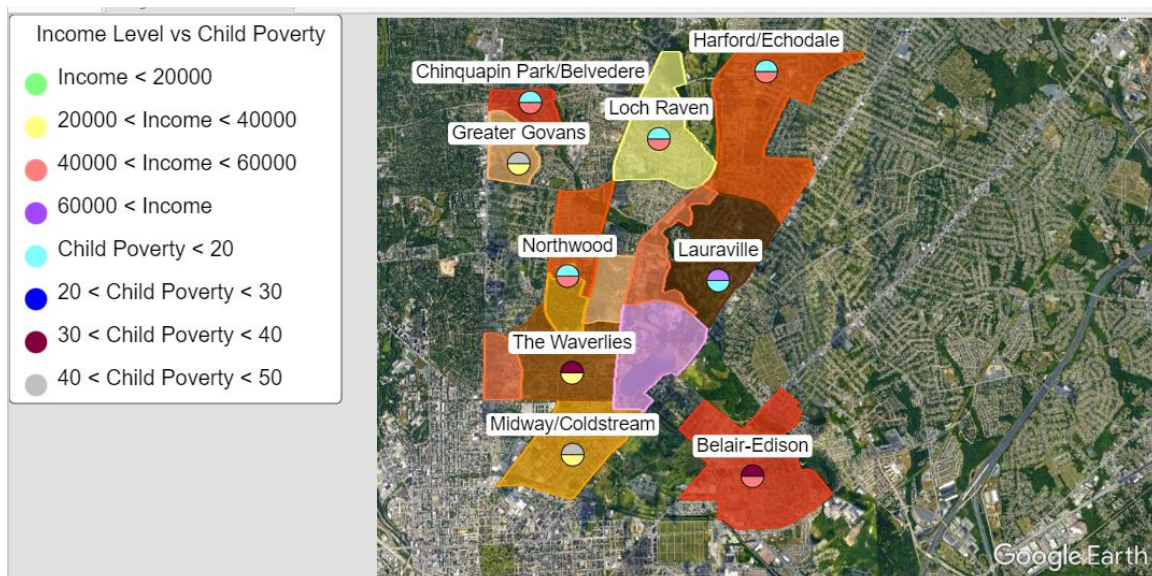


Figure A-9: MCM Community Income vs Child Poverty

Appendix B: MSU Undergrad Senior Project Research

KVN was also used by a couple of MSU students as a part of their undergrad project. It allowed the students to define information with respect to college with in a spreadsheet and import the information into KVN using the excel importer with a media database. For the creation of the media database, KVN's media database application was used as shown in Figure B-1 which gives a sample of the media. With the database defined, the media was dragged into the application. The files were then uploaded to the database and visualized in the application. To import the information into KVN the excel importer was used as shown in Figure B-2. The excel file was selected which contained information about the students, their relationships, and media. The media database was also selected to determine where the importer would find the media defined in the excel spreadsheet. The excel file was then imported as an information network as seen in Figure B-3. From looking at the network it has 4 students with relationships amongst all 4 as friends. The figure also shows the student named "Favour" selected. "Favour" is shown in the media view with an image of her transcript. The attribute view list information related to things about college with respect to "Favour."

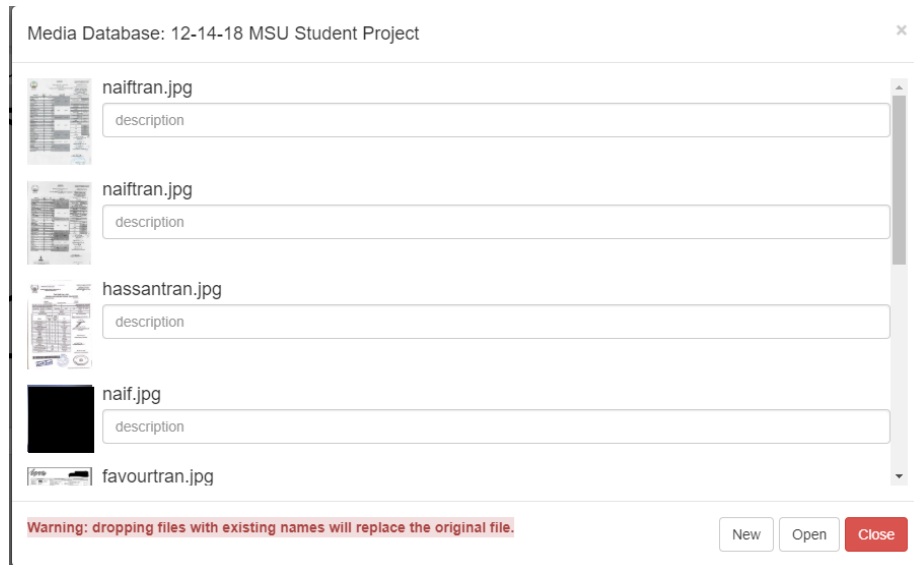


Figure B-1: MSU Student Project Media Database

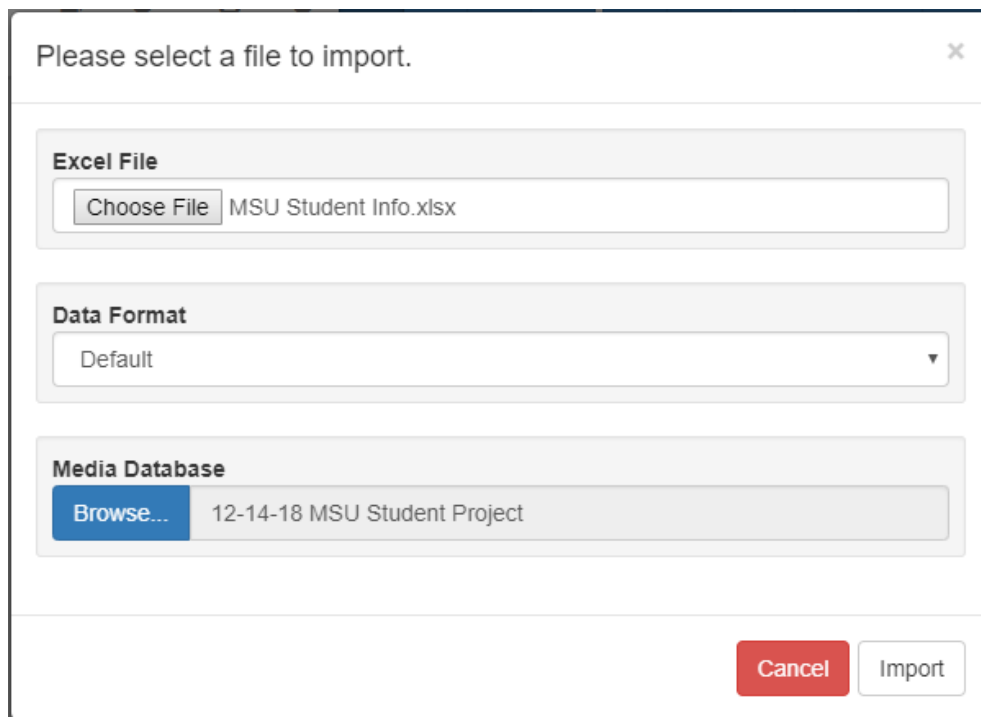


Figure B-2: MSU Student Project Excel Import w Database

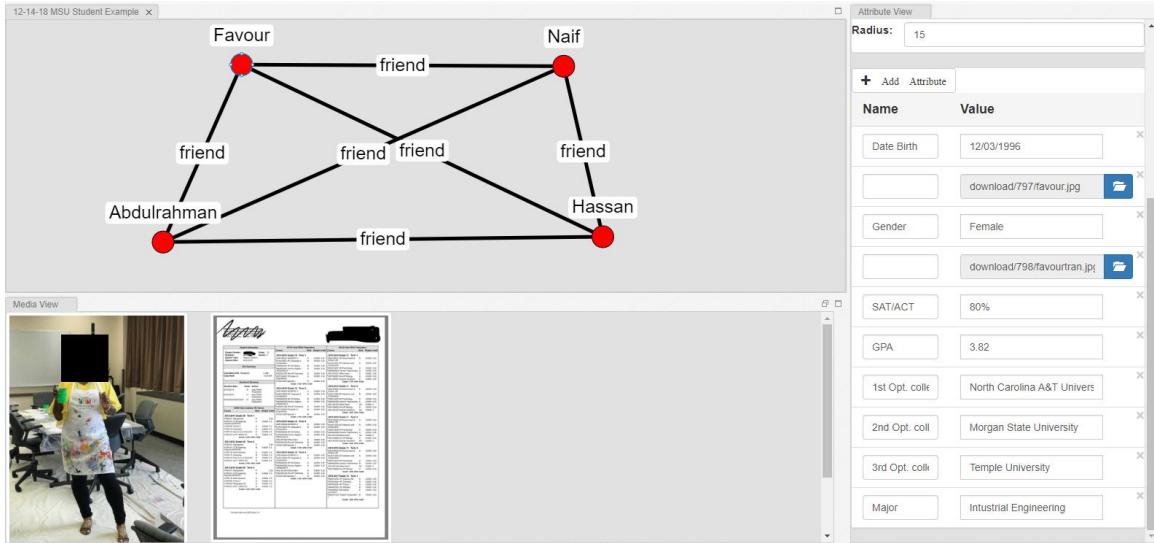


Figure B-3: MSU Student Project Network