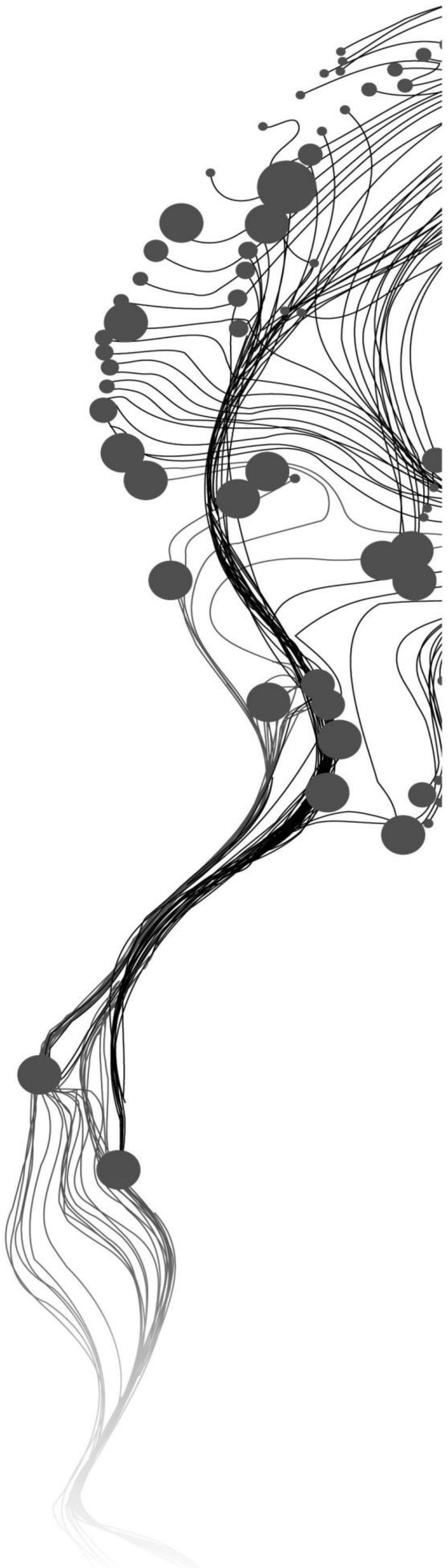
Linking an Existing ABM for Disease Diffusion with a Social Network Analysis and Visualization Module

Molham Darwish
February, 2012

Supervisors:
Ir. P.W.M. Ellen-Wien Augustijn
Dr. O. Huisman



Linking an Existing ABM for Disease Diffusion with a Social Network Analysis and Visualization Module

Molham Darwish
Enschede, the Netherlands, February, 2012

Thesis submitted to the International Institute for Geo-information Science and Earth Observation in partial fulfillment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation. Specialization: Geoinformatics.

Supervisors:
Ir. P.W.M. Ellen-Wien Augustijn
Dr. O. Huisman

Thesis Assessment Board

Dr.Ir. R.A. de By
Dr. ir. P.R. van Oel

Disclaimer

This document describes work undertaken as part of a programme of study at the Faculty of Geo-information Science and Earth Observation of the University of Twente. All views and opinions expressed therein remain the sole responsibility of the author, and do not necessarily represent those of the Faculty.

Abstract

Health is the most essential and basic element for human being. For this, modeling the spread of diseases is considered one of most important aspects in the geo-information sciences nowadays. Studying these models from social network perspectives becomes more significance, because it determines social relationships between individuals, and how these relationships could influence the disease diffusion over a network of relations, as well as analyze how the spread of information about the disease could change the behavior of individuals.

In order to study the network features of a disease model, an existing spatially explicit simulation model for the spread of pertussis that has been developed using Repast Simphony software, has been used as base to analyze the social relationships network using a software library that includes number of features to measure the network elements and visualize it graphically. An integrated methodology has been introduced to incorporate the existing model with a social network analysis library, in order to study the social interaction network features of the model and change the behavior of agents according to the spread of information about the disease. This methodology has been conceptually discussed by developing a conceptual framwork and explained in order to translate the theoretical concepts into implementation. Number of standalone frameworks and libraries for the social network analysis are reviewed and evaluated in order to find an appropriate structure for representing the network elements.

An implementation plan has been conducted in order to realize the conceptual framework, and a number of experiments have been carried out to get variety of outcomes which have been analyzed in order to verify and validate the extended model.

Depending on the work accomplished in this research and the outcomes, a list of conclusions and recommendations to improve the work and develop it including the implementation of the broadcasting of the information system with awareness level system and change the individuals' behavior accordingly.

Keywords: social network analysis, network visualization, social interaction network, social network measures, Repast, disease diffusion, change behavior.

Acknowledgments

"Read In the name of the Lord who created, (1) Create the man from a clot. (2) Read and the Lord is the Most Bounteous, (3) who teaches by the pen, (4) Teaches the man that which he knew not. (5) " Al-Alaq

One year and half were gone, now we are reading the end of this journey and we must say the last word that each student is waiting but when he arrive it, he has another feeling.

And now I must thank everyone helps me to let this thesis finish. The first one I should start with is Ms. Ellen-Wien Augustijn my first supervisor who wasn't only my supervisor; she was also my second mother. Her patience, strong and friendly support, I would never forget what you did for me. It's my honor to be in your team of research Ms. Ellen-Wien. I would like thank my second supervisor Dr. Otto Huisman for his support and suggestions. I should also thank Mr. Bas Retsios the ITC Repasts expert and best programmer ever seen who helped me especially in my implementation. I should say thank you Mr. Sascha Holzhauer form Kassel University for providing me with the needed library and his support during my implementation.

Great thanks to Mr. Gerrit Huurneman and Dr. Connie Blok for their psychological and social support during my study. I should thanks the librarian staff for their kind helps in providing me with articles and materials for reading. Many thanks to all staff and employees at ITC for their kind help and support.

Great thanks to all my GFM 2010-2011 classmates whom I spend a very nice time with them. Proudly I can say I have brothers and sisters all over the world.

Many thanks also to Shaheen Abdulkareem for her supports and advices during my research work.

I have and I should say thanks for the World Bank joint Japan program for the scholarship which gave me the chance to be one of the ITC students.

I would thanks who they are the reason of my life, to who is the Lighthouse of my life and lit it with hope, you'll always be my hero as long as I live, to you my dearest father, I appreciate all that you have ever sacrificed for us. To whom profit Mohamed said the paradise is under her feet, Mom I know how often we took you for granted, when we were growing up, we always assumed you'd be there for us when we needed you, and you always were. I really thank you both for supporting me along my life and nowadays with your love, prays. A special thanks to my big brother Dr. Ayham Darwish for whatever I needed that somehow found its way to me. You are one of my favorite people. All the Obstacles we've been through, I just wanted to say "Thank You" For being a good person to me when no one else did and for being a great supporter of me you aren't only my brother, you are also my closed friend, many thanks to my little brother Anas Darwish for his kindness and his Cheerful soul.

Last but not least, a very great thanks to the person who gave my life a new meaning, help me in achieving my goals, and inspiring me with love, I can't even tell you how much your supporting and encouraging means to me. Amany Omran you are the most tremendous human I have ever met, I really thanks the God to be my dearest love for ever and to be my spirit (I love you so much).

Table of Contents

List of figures

List of tables

1. Introduction

1.1. Motivation and problem statement

Health is one of the most important issues in human life. Diseases can really mean a down turn in our life, as it may affect the prosperity and growth of societies. For this reason, governments are investing big efforts to keep their population healthy. Despite of these efforts, there are breaking headlines in media talking about an epidemic outbreak somewhere around the globe which affected a large number of people within a short period of time, as this epidemic could be upgraded to become a pandemic which does not distinguish one society from others and affects in both small and large regions (Glaser, 2007).Both epidemics and pandemics threaten the developed as well as developing countries, because they are responsible for many death cases, as in H1N1 flu case.

The nature of the disease and the interactions between individuals are the main reasons for the diffusion of infections. Every day, people interact with each other for many different reasons such as in work places, schools and shopping centers etc. which leads to maximize the spread potentiality since these interactions may occur between infected and susceptible individuals.

Many researches and studies were conducted that focus on how individual interactions could spread epidemics (Christley et al., 2005). Some studies of epidemic diffusion focused on networks of individuals and provide a number of techniques to study and identify how epidemics diffuse(Colizza, Barthélemy, Barrat, & Vespignani, 2007). Other studies introduced theoretical frameworks to study and analyze contagious infections (Hill, Rand, Nowak, & Christakis, 2010), while others used the newest technologies to provide a simulation for the infections diffusion; one of the widely used computer technique provided in many studies is the Agent-Based Modeling approach (ABM).
ABM is used as an integrative methodology to simulate multi-agent interaction. ABM is not only used to model disease diffusion (Perez & Dragicevic, 2009), but also for many social aspects such as eco-systems or demography that are based on the interaction between individuals or agents and their environment in a network of social interactions (Fent, Scheffran, Prskawetz, & C., 2006). Generally in ABM, these simulations generate a "synthetic" population and "synthetic" contacts between agents. Synthetic population is a method which takes data of accumulated population in addition to sample population data as input and the output would be a complete list of population individuals, each with associated attribute data (Ryan, Maoh, & Kanaroglou, 2007). The aim of developing and building a synthetic population is to create the required population as precisely and accurately as possible compared to the true population (Nakamya, Moons, & Wets, 2009).

For the simulation of disease diffusion, ABM simulates the individuals' interactions with the consideration of spatio-temporal aspects in a spatially explicit environment. For example, (Abdulkareem, 2010), in her thesis's work has developed a spatially explicit agent based simulation model for pertussis disease diffusion in Enschede using Repast Simphony tool. The current pertussis diffusion model has generated a synthetic population that reflects the reality, but certain criteria need to be applied to evaluate that the synthetic population displays the correct social interactions.

Social Network analysis (SNA) is a methodological approach to study social relationships between individuals. It is a set of methods to describe, understand, explain and design the social structures (Brandes & Wagner, 2003); these methods specifically allow an investigation of the relational aspects of these structures (Sabater & Sierra, 2002). SNA is used to analyze the interactions in actual populations as well as simulated populations and contains a number of measures that can be used to characterize a population.

The most fundamental and most used measure is centrality (M. E. J. Newman, 2007). Centrality investigates the most important or central individual in the network. Furthermore, various number of centrality measures have been developed to analyze social networks (S. P Borgatti, 2005), such as degree centrality(Freeman, 1979), which is the number of links incident upon a node. Freeman (1979) also defined two measures, closeness as the mean geodesic distance (i.e., the shortest path) between a vertex and all other vertices reachable from it, and betweenness as a centrality measure which counts the number of geodesic paths that pass through a particular node. Integration and radiality measures another two measures which examine the degree an individual is connected and reachable within a network respectively (Valente & Foreman, 1998).

Walling et al. (1999) identified some measures such as “transitivity”, and “average number of contacts” which are relevant to airborne disease diffusion models. “Transitivity” is the average proportion of secondary contacts that also contact each other while “average number of contacts” indicates how many people might potentially get the infection from an infected person (Wallinga, Edmunds, & Kretzschmar, 1999).

But the question, how could these measures be linked with simulations to analyze the social interactions and generate synthetic population reveal the true interactions?

SNA have already been implemented to simulate the interaction between agents in social interaction networks by Holzhauer (2009). Holzhauer's research showed that “dealing with social interaction networks is becoming more and more attractive to modelers”. In his research, he introduced and applied number of measures provided to analyze the interaction between individuals in the simulation model; he used Repast Simphony which contains built-in library for different types of network measures.

In response to the disease diffusion, people can change their behaviors and activities to avoid being infected and to avoid their family members to get the infection indirectly, by following certain scenarios to deal with the surrounding environment in disease spread case. For example the case of the pertussis model, if a child was infected; their parents will not send him to the school until he/ she is recovered from the disease, the parents for child’s friend will avoid letting their child to visit or meet him during his/her sickness, so they protect their child as well as themselves from infection. In ABM context, behavioral change of agents could be implemented to allow the agents to complete the interactions successfully and make them more realistic (Meyer & Szirbik, 2007).Change of agents behaviors could be realized in two directions, i.e. supposing the interfere between two agents A1 and A2, A1’behavior will be adjusted to A2’s behavior, in order to avoid negative outcomes, orA1 attempt to change A2’s behavior to abort negative influences effect A1(Piunti, Castelfranchi, & Falcone, 2007).In SNA, the usage of social networks measures influences the agents’ behavior, so they can form their positions within the network to adapt their behavior and the behavior of adjacent agents (Holzhauer, 2009), when some level of conditions has been reached(Alam, 2005), such as the minimum distance between sick and susceptible agents should be reached in order to adjust the behavior of the susceptible one.

In my research, I have purposed to find and develop a methodology for linking the pertussis disease diffusion model (Abdulkareem, 2010) with a social interaction network module (Holzhauer, 2009). The implementation of the desired methodology has been achieved by the application of a list of social interaction measures that are relevant for disease diffusion models; to analyze the social interaction model and accordingly, the social interaction network will change its state dynamically as the agents are changing their positions and their behaviors according to the interactions. In addition to a prototype that illustrates the results of the implementation. The validation of the methodology has been undertaken by the use of empirical population data of the disease model which reflects the real social interaction.

Generally, carrying out this research would help to have a more realistic model that shows the main factors in societies, which have an impact on the spread of infectious diseases.

1.2. Research Identification

This section explains the identification of the research objectives and the corresponding research questions that need to be answered. Also it includes, the aimed innovation and related works of other researchers in tackling similar problems.

1.2.1. Research Objectives

The general objective of this research was to find and develop a methodology for linking an existing geographically-enabled Agent Based Model for disease diffusion with a social interaction network module with adapting the social interaction network in order to implement changes in the social behavior of an agent based on the information about the disease spread over the network, in addition to analyzing and validating the results of this implementation.
In order to achieve the general objective, the following sub objectives have been addressed:

- To identify relevant measures of social relationships network analysis.
- To determine how to link disease model and social relationships network analysis.
- Propose adjustment to the activities implemented in the social interaction model of the pertussis model.
- To identify how the broadcasting of information about the disease could change the social behavior of agents according to their positions.
- To analyze the social interaction generated by the pertussis model and compare it with the empirical data.

1.2.2. Research Questions

The sub objectives have been addressed by answering couple of questions, which have drawn the outline to achieve the research objective and sub objectives.

1. Which social interaction network measures are relevant to study the disease diffusion?

    1.1. Which criteria will be used to identify the relevant measures?

    1.2. What is the list of relevant measure to be adopted in the methodology?

2. How to develop a methodology to link the disease model and social network analysis.

    2.1. How is the synthetic contacts population in the current model integrated?

    2.2. How can these measures be integrated in the ABM model?

    2.3. How to evaluate the synthetic social interaction in the disease spread model?

3. What improvements need to be applied to the models in order to implement the link between the two models?

    3.1. Do the measures will be incorporated to the social interaction model?

    3.2. Do the measures will be adjusted to reflect the social activities in the pertussis model?

- How the broadcasted information about the disease could change the social behavior of agents according to their positions.

    3.3. How to change the social interaction network to be dynamic.

    3.4. How the social interaction model could be improved to implement the change of the agent's behaviors?

    3.5. What change will be implemented to the social activities?

4. Which criteria will be used to validate the social interactions of the disease model?

    4.1. Which empirical data will be used to evaluate the model?

    4.2. How the social interactions of the disease model would be tested and evaluated?

1.2.3. Innovation Aimed At

The novelty of this research would be developing a valid methodology to link a geographically-enabled Agent Based Model with a social interaction network model by the implementation of different social interaction network measures.

Moreover, changing in the social behavior of an agent based on its position in the network thought to be a novelty.

The validation of a social simulation model and making sure that an implemented model matches the real-world is thought to be an innovative character for the domain of Geo-informatics, since the ways the model might be validated are raised to be one of the key challenges in ABM.(Crooks, Castle, & Batty, 2008).

1.2.4. Related Work

ABMs are capable of representing very complicated systems that are irreducibly heterogeneous, random and contain complex interactions. The principle of ABM is to represent explicitly the heterogeneous parts of a system in a computer model, rather than attempting to “coarse grain”. This method enables us to take into consideration temporal and spatial aspects of disease diffusion, individual contact processes as well as their behavior that greatly affects the course of an epidemic outbreak. Therefore, representing the spread of diseases geographically in a social interaction manner and analyzing it, by using different modeling tools and techniques, has a great importance(Perez & Dragicevic, 2009).

Recently, many researchers have introduced many computational modeling approaches to study social interaction modeling of the epidemics diffusion. For instance, for the case of the diffusion of pertussis, number of models has been conducted to simulate the diffusion of pertussis between individuals. Hethcote (1997) introduced computer simulations to imitate the mathematical model of the pertussis transmission and vaccination model to approximate the spread of pertussis from 1940 to 2040. In the United States, computer simulations using an age-structured epidemiological model for pertussis transmission and vaccination have been used to obtain estimates of the potential effects of the addition of adult pertussis booster vaccinations every 10 years on pertussis incidence in both adults and children in the USA(Hethcote, 1999). the model developed by Hethcote (1997, 1999) was used to make comparison of the likely effects of adopting different vaccination strategies in Australia on pertussis transmission by age group over a 50 year time period(Hethcote, Horby, & McIntyre, 2004). Based on the model developed by Hethcote (1999), this simulation model has been adjusted to the Dutch state in order to simulate the spread of pertussis in the Netherlands (Augustijn, Abdulkareem, Huisman, & Flacke, 2010). The used model in this study was the social interaction model of spared of pertussis which is going to be analyzed in this research by the use of social network measures. The pertussis model developed by Abdulkareem, S.A (2010) was based on Individual Space-Time Activity Based Model (ISTAM) which was developed by Yang and Atkinson (2007) and implemented using Repast, java and UML with incorporating three objects: the infectious diseases, individuals and activity bundles(Yang & Atkinson, 2007).

In the other hand, many researchers introduced and studied various numbers of social network analysis methods and concepts, as well as introduced number of measures for the social interactions networks. Centrality is one of the most studied notions in SNA (S. P Borgatti, 2005). In his study, Borgatti (2005) introduced numerous numbers of measures which have been developed to analyze social networks, such as degree centrality, closeness, betweenness, eigenvector centrality, and flow betweenness. He outlined several typologies of network flow which is commonly encountered during flow processes; such is goods, money, gossip, e-mail, attitudes, infections, and packages. He matched between the different measures of centrality and the defined types of flows to examine the relationship between them and which measures are appropriate for which flow process.

Different application have been developed in the field of Network analysis and modeling of social interactions, such as Java Universal Network / Graph Framework (JUNG) (O’Madadhain, Fisher, Smyth, White, & Boey, 2007) which comprise the implementation of number of analysis algorithms and measures to analyze the social network data, as well as other modeling frameworks as NetLogo, Repast J, and Repast Simphony. Based on Repast toolkits, Holzhauer (2009), in his work, introduced newly developed software library in order to provide for the practitioners a framework to deal and analyze social interaction models such as ABM framework. The name of that library is ReSoNetA (Repast Social Network Analysis), which adds new network functionalities to the Repast J framework, and offers extended network measures to analyze the networks.

In this research, the goal was to study different network measures, and select the relevant measures which could be matched with the spread of infections in order to analyze the pertussis simulation model and the social interactions between different agents, with adapting the social interaction network in order to implement changes in the social behavior of an agent based on the position in the network.

1.3. Project Set-up

This section briefly describes on how the study have been attempted to address the formulated research questions. It includes the description of the proposed approach and methods to be used, the work flow that leads from the initial research questions to the final results and conclusions, and the research work plan that gives an overview of the sequence of the research with respect to the allocated time. Finally, it will describe the potential risks that can be occurred during this research and their solutions.

1.3.1. Method Adopted

The research approach has been set to study and analyze the social network model to identify set of measures, which are applicable for the simulation model of pertussis spread in a social interaction network, and develop a methodology to linkage the selected measures with the social interaction model. The selected measures will be used to adapt the agent's behavior based on its position in the network, subsequently use of empirical data to check the validity of the implementation.

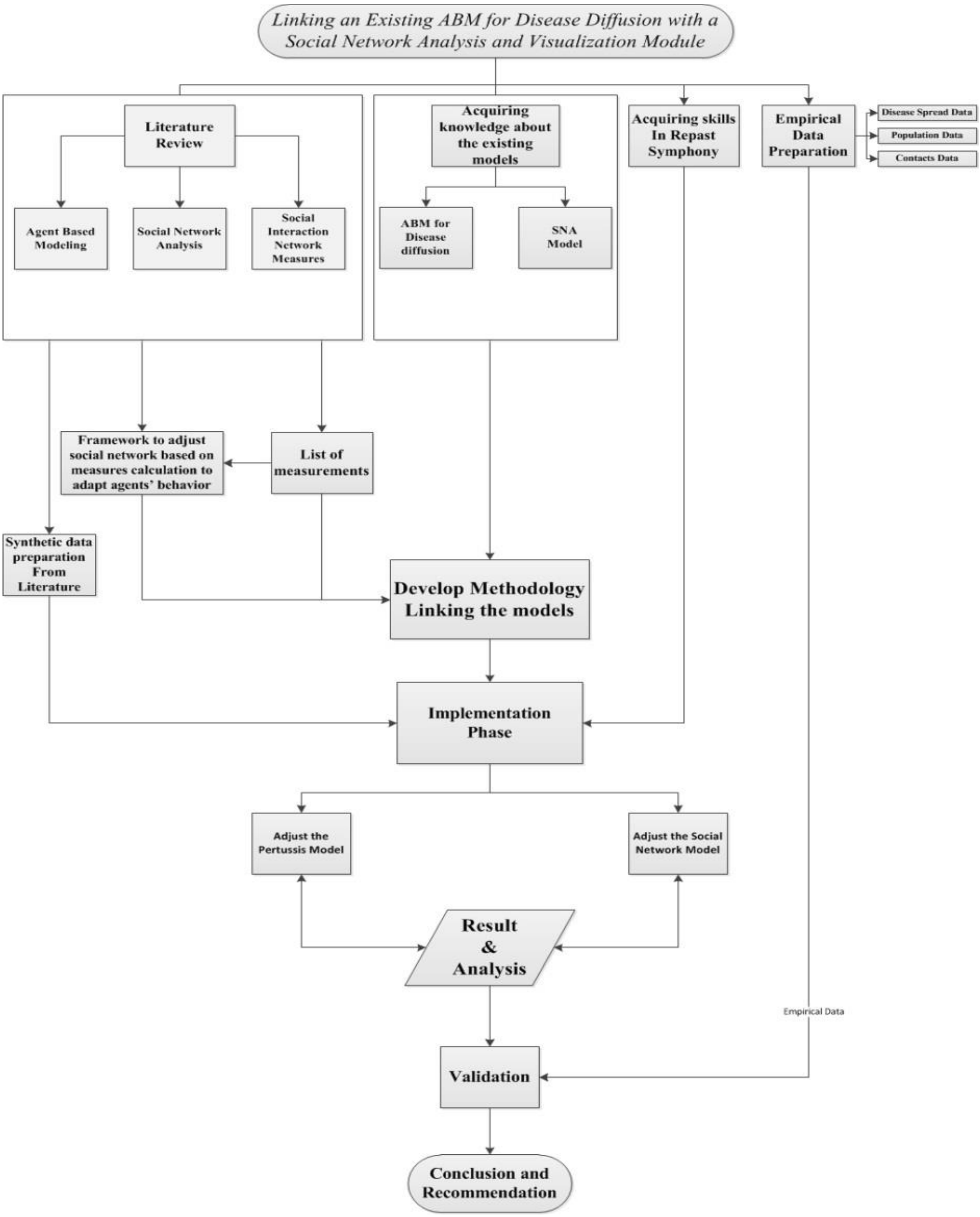
In order to achieve the research objectives and answer the research questions, a methodology has been developed during the research as described in the flow chart below. This methodology consists of the following phases:

- Literature Review includes:
  - Study related literatures about social interaction models.
  - Study related literatures about the concept of Social Network Analysis.
  - Study of Social Interaction Network Measures to select the relevant measures for disease spread simulation.

  The expected outcomes of reviewing related literatures understand the possibilities to link the social interaction model with social network analysis and determine the relevant social interaction measures, and define the concept of changing an agent's behavior based on its position or adapting the adjacent agents' behavior.
- Acquiring knowledge about the two models to be linked:
  - Agent Based Simulation model for the spread of pertussis in Enschede region(Abdulkareem, 2010).
  - Social Network Analysis and Visualization Module for Repast Models (Holzhauer, 2009).

  The gained knowledge of this phase is to understand the methodology used to analyze the social interaction network of pertussis model, how the measures included in SNA model are implemented with simulation models and to set the blueprint to find and develop the desired methodology of linking the two models.
- Find and develop a methodology to analyze the social interaction model by linking the two models with the implementation of the selected and relevant measures with the social interaction model, and provide means to adjust the social interaction network to make it dynamic based on calculations of these measures in order to formalize an agent's position to adjust its behavior.
- Acquiring skills In Repast Symphony tool, which is java-based modeling software, so using this software requires being familiar with the software framework and recalling the Object Oriented Programming (Java).

- Synthetic and empirical Data preparation including disease spread data and social interaction data, in order to use the data to validate the results of the implementation, by looking through the literature and other sources to extract the required data for the implementation and validation.
- Implementation of the methodology using Repast Symphony tool. The implementation could be accomplished:
  - Adjustment will be added to the pertussis model to implement the desired methodology.
  - Adjustment will be added to the Social Network model to implement the desired methodology.
- Use empirical data to check the validity of the implementation.
- Draw a conclusion and set of recommendations according to the results.


Linking an Existing ABM for Disease Diffusion with a Social Network Analysis and Visualization Module
Literature Review
Agent Based Modeling
Social Network Analysis
Social Interaction Network Measures
Acquiring knowledge about the existing models
ABM for Disease diffusion
SNA Model
Acquiring skills In Repast Symphony
Empirical Data Preparation
Disease Spread Data
Population Data
Contacts Data
Framework to adjust social network based on measures calculation to adapt agents' behavior
List of measurements
Synthetic data preparation From Literature
Develop Methodology Linking the models
Implementation Phase
Adjust the Pertussis Model
Adjust the Social Network Model
Result & Analysis
Empirical Data
Validation
Conclusion and Recommendation

1.4. Organization of the thesis

This research thesis is organized into the following chapters after this chapter of introduction:

- Chapter 2 addresses the theories and background of the research. In addition, it provides the reader with the concepts used to address the sub objectives and theoretical answers regarding the research questions. An overview about the related works and previous works can be found within this chapter.
- Chapter 3 provides detailed description on the conceptual design of the methodology adopted and how the implementation will be accomplished. It has been provided with some diagrams explaining the architecture of the model to give the reader better understanding about the conceptual model.
- Chapter 4 reviews software and libraries that contains features to analyze and visualize social relationships networks. They are evaluated according to data model and structure, visualization techniques and descriptive methods regarding social network analysis.
- Chapter 5 presents and discusses the implementation approach used to achieve the laid out objective in line with the conceptual framework in chapter 3.
- Chapter 6 presents and analyzes the results of the implementation approach used in order to verify and validate the extended model.
- Finally, chapter 7 discusses conclusion based on the analysis of the results, and suggests recommendations of further improvements to be conducted on the work.

2. Background Information

2.1. Disease diffusion

An epidemic (outbreak) occurs when an infectious disease spreads rapidly infecting many people. This diffusion may occur in one community or even extend to several countries. It can spread from one place to another through time (Cliff, 1981) and also last from days to years. Sometimes a single case of a contagious disease is considered an outbreak. This may be true if it is an unknown disease, is new to a community, or has been absent from a population for a long time. For example, in 2003, the severe acute respiratory syndrome (SARS) epidemic took the lives of nearly 800 people worldwide.

2.1.1. Pertussis

Pertussis, also known as whooping cough, is an air borne and respiratory tract disease caused by bacteria called Bordetella that lives in the mouth, nose, and throat (world Health Organization, 2010). It is a serious infection that spreads easily from person to person. The infection causes coughing spells so severe that it can be hard to breathe, eat or sleep. It can even lead to cracked ribs, pneumonia or hospitalization.

Early symptoms of pertussis are similar to the common cold or bronchitis and may include runny nose, sneezing and low-grade fever. The infection also causes coughing that lasts for weeks, even months. Sometimes a “whoop” sound occurs while gasping for breath during a bad coughing spell. However, the “whoop” is not always present; adults rarely have the classic “whoop.”

Pertussis can occur at any age, regardless of vaccination history. Apnea rather than cough may be the initial or most important symptom in infants less than 6 months of age. A clue to the diagnosis in infants only is an elevated white blood count (over 15,000/mm^3) with a predominance of lymphocytes. Pertussis among older children, adults, and those previously immunized can be milder than classic whooping cough; the symptoms may be no more distinctive than other upper respiratory tract infections (Health, 2011).

Prior to the introduction of pertussis vaccination called DTP3, most mortality cases of children occurred in young children(Boven, Melker, Schellekens, & Kretzschmar, 2000). After introduction of the vaccination, the illness and death are sharply decreased. Now the highest incidence, morbidity and mortality are in infants. More cases have recently been occurring in adolescents and adults. This change in the epidemiology of pertussis is due to the waning immunity that occurs after both disease and vaccination, and to a reduction in natural boosting. Thirty per cent of adults with a cough lasting longer than 2 weeks may have pertussis. Most infants and young children who contract pertussis are infected by a family member. In 1996, 2,771 cases of pertussis were reported to the Inspectorate of Health in the Netherlands (population 15 million), compared with 319 cases in 1995. With epidemic cycles expected every 3 to 5 years and a recent outbreak in 1994, this rise was unexpected. After the introduction of pertussis immunization with a whole-cell vaccine in the National Immunization Program (1952), the incidence of pertussis in the Netherlands decreased significantly (De Melker et al., 1997). The annual incidence of pertussis notification from 1989 - 1995 is shown in Figure (2.1).

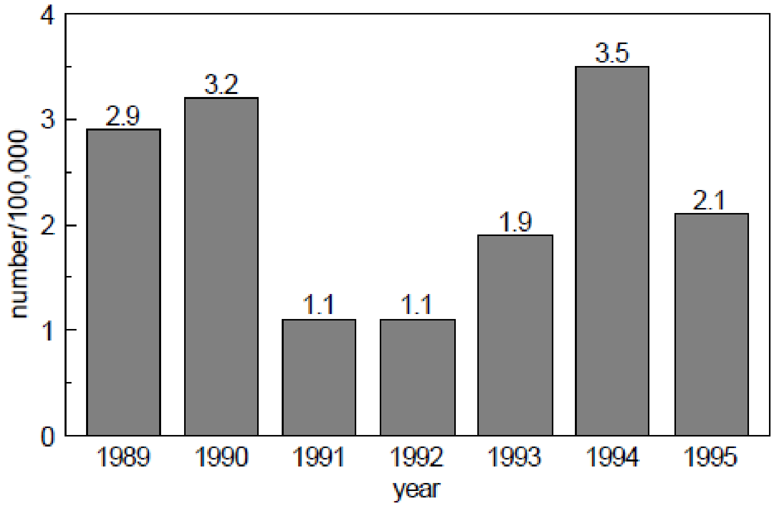


Figure (2.1) annual incidence of pertussis estimated from notification by registration date, 1989 - 1995

2.1.2. Social interaction and spread of diseases

In social science, a social relation or social interaction refers to the acts, actions, or practices of two or more people mutually oriented towards each other's selves, that is, any behavior that tries to affect or take account of each other's subjective experiences or intentions (Rummel, 1976). This means that the groups involved in the social interaction must be aware of each other--have each other's self in mind. This does not mean being in sight of, or directly behaving towards each other. Friends writing letters are socially interacting. Social interaction is not defined by type of physical relation or behavior, or by physical distance. It is a matter of a mutual subjective orientation towards each other. Thus even when no physical behavior is involved, social interactions refer to particular forms of externalities, in which the actions of a reference group affect an individual's preferences (Scheinkman, 2009).

The social interaction between individuals in society could happen by face to face meeting or physical interaction, at schools, shopping centers, sport centers, transportation facilities.

The communication between people and interacting also could occur non- physically for example by phones, emails, social media, etc…

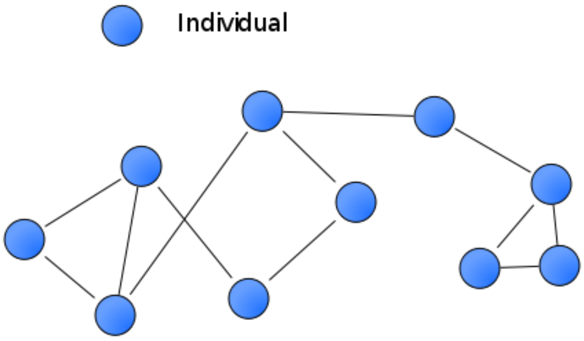
Although social interaction is complex, it is critical to human health, both mentally and physically. Whatever the way people interact, the interaction influences how epidemics outbreaks spread across different communities. Human interaction plays an important task in the spread of infectious diseases, and understanding the influence of behavior on the spread of diseases can be key to improve control efforts (Funk, Salathe, & Jansen, 2010). Human behavior has been intricately linked with the spread of infectious diseases, the direct interacting between individuals will spread the illness from sick to susceptible persons depending on the type of the disease (airborne epidemic, sexual diseases, etc…), while the communications through newest communities (social medias for example) are becoming widespread among people of all ages (Xu & Liu, 2010), in which it provide us a platform to share opinions and awareness publicly, which in turn would influence the epidemics outbreaks, warning the people through these medias to take certain actions to avoid getting infected, such as get vaccinations in early stages, reduce the possibility to interact with others, aware their children and relatives, etc…

Human social behavior is an extremely important determinant of infectious disease transmission. Mathematical infectious disease modeling relies on assumptions made about the social contact patterns of populations and population sub-groups. It is therefore crucial to develop a detailed understanding of the structure and characteristics of social networks at a whole population level, i.e. to understand how people are connected and the nature of these connections.

Cohen & Brissette (2004) introduced the concept of social integration which refers to the degree to which one participates in the social community he / she belongs to. In their article, Cohen &

Brissette (2004) discussed the implication of social integration for health. They studied the case of the common cold and the impact of the participation in a diverse social network on health. It is found that the greater the social diversity, the lesser the susceptibility to infectious illness (Cohen & Brissette, 2004).

2.2. Human behavior



To diffuse different diseases require different levels of interaction. For instance, smallpox requires very close (physical) contact of infected and susceptible individuals. Similarly, infectious sexual disease such as HIV/AIDS requires sexual (physical) contact. On the other hand, diseases like pertussis, influenza, and measles require only a conversational proximity between people since these diseases diffuse through airborne droplets. It has been argued that defining social interaction types, which are causes by the diffusion of airborne diseases, are harder than that of sexually transmitted diseases. Because, according to the analysis of Read et al. (2008), per day individuals has approximately seven times more conversational contact than physical contact.

An epidemiologist, that is, a scientist who studies diseases and their spread throughout a population, stated that early detection of health trends is crucial to staying ahead of an outbreak and potentially saving many lives. Behavioral responses to the spread of a disease have often been reported anecdotally, there has been relatively little systematic investigation into how behavioral changes can affect disease dynamics. The emergence of a disease combines two elements: the introduction of the pathogen into the human population and its subsequent spread and maintenance within the population. Ecological factors such as human behavior can influence both of these elements, and consequently ecology has been recognized to have an important role in the emergence of disease (Antia, Regoes, Koella, & Bergstrom, 2003). During the outbreak of influenza A (H1N1) in 2009, the effect on societies, partly through public measures but also through personal and uncoordinated responses, has been noticeable. The public reaction to this disease was sustained and widespread, and, interestingly, part of this reaction resulted from individual behavioral responses to the presence of the disease. Funk et al (2010) classified that human behavior is based on attitudes, belief systems, opinions and awareness of a disease and all these factors can change over time, both in an individual and in the population on the whole. The dynamics in these attributes is one element that is relevant to understanding the impact of behavioral responses to a disease. There are a large number of behavioral changes that can influence the spread of infectious diseases, such as vaccination behavior and its interaction with the epidemiology of a disease, reductions in the number of potentially infectious contacts (such as the figure above when an infected individual would change his behavior to avoid transfer the infection to other individuals in the group), wearing of face masks or practice of better hygiene. In order to study the effects of individuals' behavioral changes, Funk et al (2010) proposed a number of mathematical models for the spread of infectious diseases as an important tool for investigating and quantifying such effects, such as the SIR (susceptible–infected–recovered) model (Kermack & McKendrick, 1927) which forms the basis of almost all the disease models studied since. Based on that model, the behavioral change is relevant for infectious disease dynamics; it must affect either the disease state (susceptible, infected or recovered) of the individual, the infection rate or the recovery rate, or the contact network structure relevant for the spread of disease. For example, in models where a behavioral change constitutes a decision to vaccinate, the individual will leave the susceptible state and move directly into the immune state. As another example, consider a model where disease occurrence causes increased social distancing, such a behavioral change could be modeled either as a decrease in the communication rate, or as reduction in the contacts intensity.

Moreover, the socioeconomic characteristics of societies and the human demographical features characterize disease diffusion dynamics. Lert & Dray-Spira (2002) studied the social inequalities in the course of chronic HIV infection and the impact of social characteristics on health status and its determinants in the context of HIV infection. Several types of difference associated with socio-demographic characteristics may explain the social inequalities: such as differences in access to health care, differences in adherence to treatments, differences in comorbidity and differences in living conditions, life style and health behaviors (Lert & Dray-Spira, 2002), in addition to a number of socioeconomic factors which would have influence on the clinical and/or biological progression of the disease. Some of these reported factors is gender, ethnic group, transmission category, income level, occupational status or educational level to be associated with disease course (Lert & Dray-Spira, 2002).

By summarizing the different aspects that could influence the spread of diseases, we come to the following list of changes in behaviors that would play a role in the diseases outbreak:

- Vaccination behavior.
- Reductions in the number of potentially infectious contacts.
- Wearing of face masks.
- Practice of better hygiene.
- Socioeconomic characteristics.

2.3. Modeling epidemics

Conceptualizing a population as a set of individuals linked together to form a large social network provides a rich perception for better understanding the spread of some infectious diseases. This population is used to show the potential usefulness of a network approach in evaluating the infectious agent hypothesis when studying a disease or disease outbreak of unknown etiology and in developing strategies to limit the spread of an infectious agent transmitted through personal relationships (Klovdahl, 1985). ABMs are capable of representing very complicated systems that are irreducibly heterogeneous, random and contain complex interactions. The principle of ABM is to represent explicitly the heterogeneous parts of a system in a computer model, rather than attempting to “coarse grain”. This method enables us to take into consideration temporal and spatial aspects of disease diffusion, individual contact processes as well as their behavior that greatly affects the course of an epidemic outbreak. Therefore, representing the spread of diseases geographically in a social interaction manner and analyzing it, by using different modeling tools and techniques, has a great importance(Perez & Dragicevic, 2009).

Recently, many researchers have introduced many computational modeling approaches to study social interaction modeling of the epidemics diffusion. For instance, for the case of the diffusion of pertussis, number of models has been conducted to simulate the diffusion of pertussis between individuals. Hethcote (1997) introduced computer simulations to imitate the mathematical model of the pertussis transmission and vaccination model to approximate the spread of pertussis from 1940 to 2040. In the United States, computer simulations using an age-structured epidemiological model for pertussis transmission and vaccination have been used to obtain estimates of the potential effects of the addition of adult pertussis booster vaccinations every 10 years on pertussis incidence in both adults and children in the USA (Hethcote, 1999). the model developed by Hethcote (1997, 1999) was used to make a comparison of the likely effects of adopting different vaccination strategies in Australia on pertussis transmission by age group over a 50 year time period(Hethcote et al., 2004). Based on the model developed by Hethcote (1999), this simulation model has been adjusted to the Dutch state in order to simulate the spread of pertussis in the Netherlands (Augustijn et al., 2010). The used model in this study was the social interaction model of spared of pertussis which is going to be analyzed in this research by the use of social network measures. The pertussis model developed by Abdulkareem, S.A (2010) was based on Individual Space-Time Activity Based Model (ISTAM) which was developed by Yang and Atkinson (2007) and implemented using Repast, java and UML with incorporating three objects: the infectious diseases, individuals and activity bundles(Yang & Atkinson, 2007).

2.4. Social relationships Networks

A network—also called a graph in the mathematics literature—is made up of points, usually called nodes or vertices, and lines connecting them, usually called edges. Mathematically, a network can be represented by a matrix called the adjacency matrix A, which in the simplest case is an n x n symmetric matrix, where n is the number of vertices in the network. The adjacency matrix has elements (M. E. J. Newman, 2007).

$$A_{i\,j} = \begin{cases} 1 \text{ if there is an edge between vertices i and j,} \\ 0 \text{ otherwise.} \end{cases}$$

The matrix is symmetric since if there is an edge between i and j then clearly there is also an edge between j and i. Thus Aij = Aji.

In some networks, the edges may be weighted (Badrow, Helfert, & Steinicke, 2010), i.e. specifying the intensity of the relationship or the probability of an information transmission. moreover, we can distinguish between different types of graphs, such as undirected graph in which edges have no orientation (so the edge (a, b) is identical to the edge (b, a) as shown in figure (2.2.a), while the edges are ordered pairs in the directed graph, figure (2.2.b) (Badrow et al., 2010).

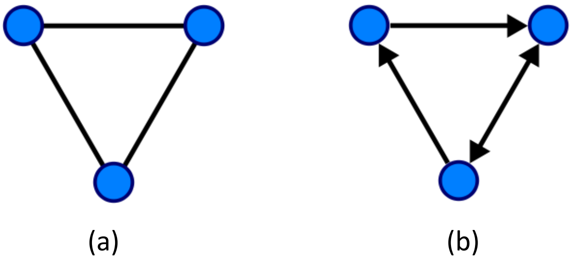


Figure (2.2): undirected and directed graphs

Associated with each path is a distance equal to the number of edges in that path, The shortest paths linking a given pair of points are called geodesics (Freeman, 1979).

2.5. Social network measures

Social network analysis (SNA) is a quantitative analysis of interactions between individuals or organizations. By measuring such social structures it is possible to identify important actors, group constructions or equivalent roles of actors within a social network (Wasserman & Faust, 1994).
Within graph theory and network analysis, there are numerous number of measures applied to vertices within a graph that determine the relative importance of a vertex within the graph (for example, how important a person is within a social network, or, in the theory of space syntax, how important a room is within a building or how well-used a road is within an urban network).

2.5.1. Types of network measures

A primary use of graph theory in social network analysis is to identify “important” actors. The importance of an actor is defined by the number of main contacts an actor has with others, in addition to the number of secondary contacts of that actor. Centrality and prestige concepts seek to quantify graph theoretic ideas about an individual actor’s prominence within a network by summarizing structural relations among the g nodes. Group-level indexes of centralization and prestige assess the dispersion or inequality among all actors’ prominences.

2.5.1.1. Centrality

Actor has high involvement in many relations; regardless of send/receive directionality (volume of activity). Many of the centrality concepts were first developed in social network analysis, and a lot of the terms used to talk about centrality reflect their sociological origin (M. Newman, 2010).

The most fundamental and most used measure in SNA is centrality (M. E. J. Newman, 2007). Conceptually, centrality is fairly straight forward: we want to identify which nodes are in the ‘center’ of the network. In practice, identifying exactly what we mean by ‘center’ is somewhat complicated. Furthermore, various number of centrality measures have been developed to analyze social networks(S. P Borgatti, 2005), such as degree centrality (Freeman, 1979).

- Degree Centrality:

An actor with high degree centrality maintains numerous contacts with other network actors. Actors have higher centrality to the extent they can gain access to and/or influence by others. A central actor occupies a structural position (network location) that serves as a source or conduit for larger volumes of information exchange and other resource transactions with other actors. Central actors are located at or near the center in network diagrams of social space. In contrast, a peripheral actor maintains few or no relations and thus is located spatially at the margins of a network diagram.

$$C_D = d(n_i) = \sum_j X_{ij} \qquad (1)$$

Where: C_D degree centrality of node i, $d(n_i)$ degree of the node n_i, X_{ij} total number of ties j that connects the node i.

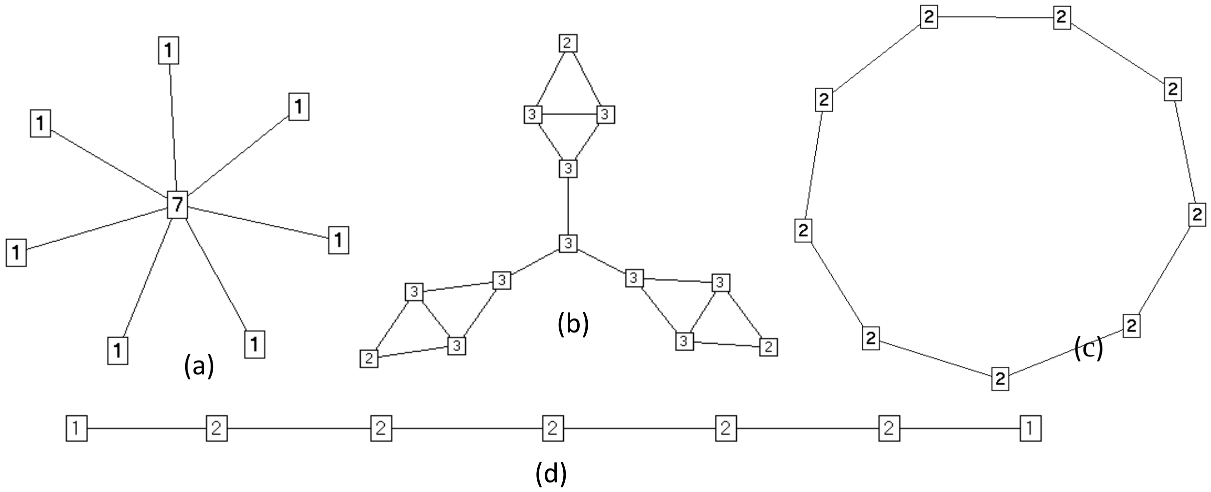


Figure (2.3): degree centrality examples

In degree centrality, the node with the most ties is the most important node. In the star network in figure (2.3.a), the central node has a degree value of 7 since it has 7 ties with other nodes within the network, but the seven peripheral nodes each have degree centrality = 1. The 10 nodes in the network shown in figure (2.3.b), each have degree centrality = 3 unless the three peripheral nodes have smaller degree centralities = 2, while all nine circle graph nodes represented in figure (2.3.c)

have identical degree centrality = 2, so no central nodes exists, since in the line graph in figure (2.3.d), the two end nodes have degree centralities = 1 and those in the middle = 2.
It is worth to compare the value of a certain node to others, let's say the largest one. This method is called Normalization, which means to divide the value of a node by the maximum value within the network. As a result, all nodes values range from 0-1 and a value close to one indicates it is close to the largest value. It is also important also to estimate the measured value of a given node as it is calculated for a certain network size. This method is called standardization, usually by dividing the centrality value of a node by the maximum possible degrees (g-1 nodes if everyone is directly connected to i), and express the result as either a proportion or percentage.

$$C'_D = {}^{d(n_i)}\!\big/_{(g-1)} \tag{2}$$

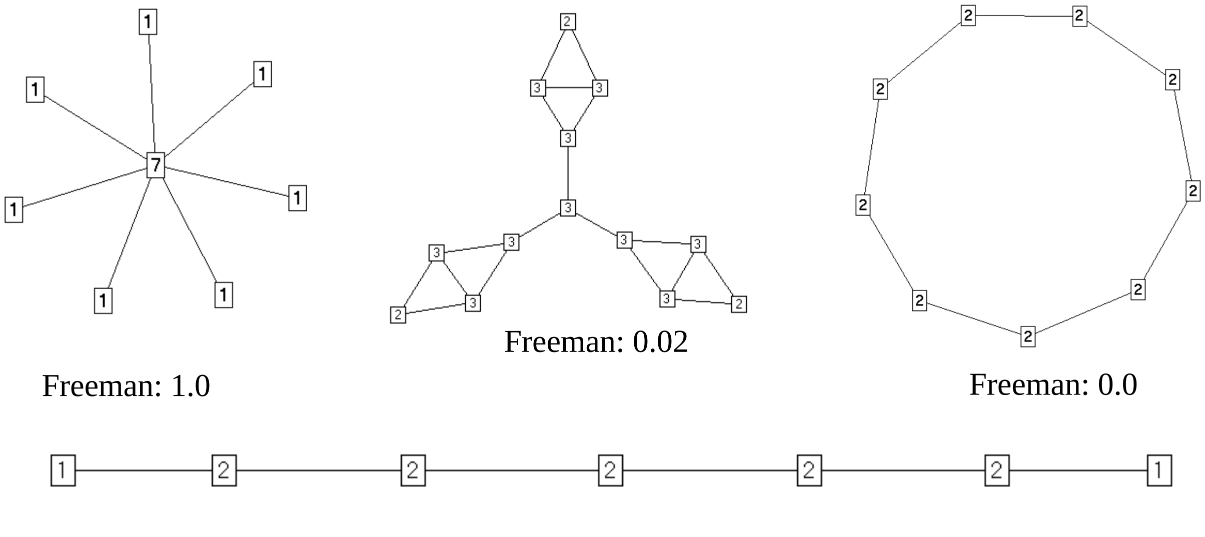
Where: C'_D standardized degree centrality of node I, $d(n_i)$ degree of the node ni, $(g\text{-}1)$ maximum possible ties connected to node i.

In the star network in figure (2.3.a), the standardized value for the central node is 1.00 and the seven peripheral nodes each have standardized degree centrality = 0.143. The 10 nodes in the network shown in figure (2.3.b), each has standardized degree centrality = 0.25 unless the three peripheral nodes have standardized degree centrality = 0.167, while all nine circle graph nodes represented in figure (2.3.c) have identical standardized degree centrality = 0.25, since in the line graph in figure (2.3.d), the two end nodes have standardized degree centrality = 0.167and those in the middle = 0.333.

If we want to measure the degree to which the graph as a whole is centralized, we look at the dispersion of centrality or normalizing degree, by the use of Freeman's general formula for normalization (which ranges from 0 to 1)(Scott, 2000).

$$C_D = \frac{\sum_{i=1}^{g}[C_D(n^*) - C_D(n_i)]}{[(g-1)(g-2)]} \tag{3}$$

Where: $C_D(n^*)$ maximum degree centrality in the whole graph, $d(n_i)$ degree of the node ni, g number of nodes.
The normalization values of the networks in figure (2.3) are calculated in figure (2.4).



Freeman: 0.02

Freeman: 1.0

Freeman: 0.0

Freeman: 0.07

Figure (2.4): normalization degree according to Freeman's formula

Freeman (1979) also defined two measures, closeness as the mean geodesic distance (i.e. the shortest path) between a vertex and all other vertices reachable from it, and betweenness as a centrality measure which counts the number of geodesic paths that pass through a particular node.

- Closeness centrality:

A node is considered important if it is relatively close to all other nodes. Closeness is based on the inverse of the distance of each node to every other node in the network, or the inverse of the sum of geodesic distances from node i to the g-1 other nodes. It is expressed by the formula (4):

$$C_c(n_i) = \left[\sum_{j=1}^{g} d(n_i, n_j) \right]^{-1} \tag{4}$$

Where: $C_C(n_i)$ closeness centrality of node i, $d(n_i, n_j)$ geodesic distance from node i to node j, g: total number of nodes in the network.

Closeness can be calculated only for connected graphs, because distance is “infinite” (undefined) if members of a nodal pair are not mutually reachable (no paths exist between i and j). Normalize a closeness index by dividing by a maximum possible distance expressed as a proportion or percentage.

$$C_C^{'}(n_i) = (C_C(n_i))(g-1) \tag{5}$$

Where: $C'_C(n_i)$ normalized closeness centrality of node i, $C_C(n_i)$ closeness centrality of node i, $(g-1)$ maximum possible distance

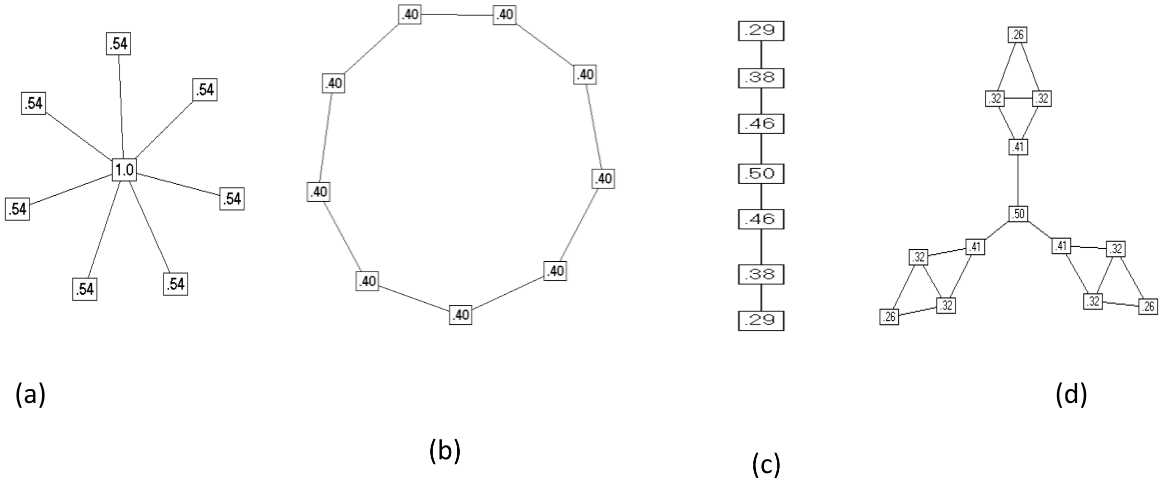


Figure (2.5): examples of Closeness Centrality

node	a		b		c		d	
	Closeness	Normalized	Closeness	Normalized	Closeness	Normalized	Closeness	Normalized
1	0.143	1.00	0.050	0.400	0.048	0.289	0.021	0.255
2	0.077	0.538	0.050	0.400	0.063	0.375	0.027	0.324
3	0.077	0.538	0.050	0.400	0.077	0.462	0.027	0.324
4	0.077	0.538	0.050	0.400	0.083	0.500	0.034	0.414
5	0.077	0.538	0.050	0.400	0.077	0.462	0.042	0.500
6	0.077	0.538	0.050	0.400	0.063	0.375	0.034	0.414
7	0.077	0.538	0.050	0.400	0.048	0.286	0.034	0.414
8	0.077	0.538	0.050	0.400			0.027	0.324
9			0.050	0.400			0.027	0.324
10							0.021	0.255
11							0.027	0.324
12							0.027	0.324
13							0.021	0.255

Table (2.1): calculation of Closeness Centrality and normalized closeness

- Betweenness centrality

A person positioned on communication paths can control communication flow, and is thus important. Betweenness centrality counts the number of shortest paths between i and k that actor j resides on. A central actor occupies a "between" position on the geodesics connecting many pairs of other actors in the network. As a cut point in the shortest path connecting two other nodes, a between actor might control the flow of information or the exchange of resources, perhaps in the case of disease, a child potentially would transmit the infection from the school to his parents at home. If more than one geodesic links a pair of actors, assume that each of these shortest paths has an equal probability of being used (Freeman, Borgatti, & White, 1991).

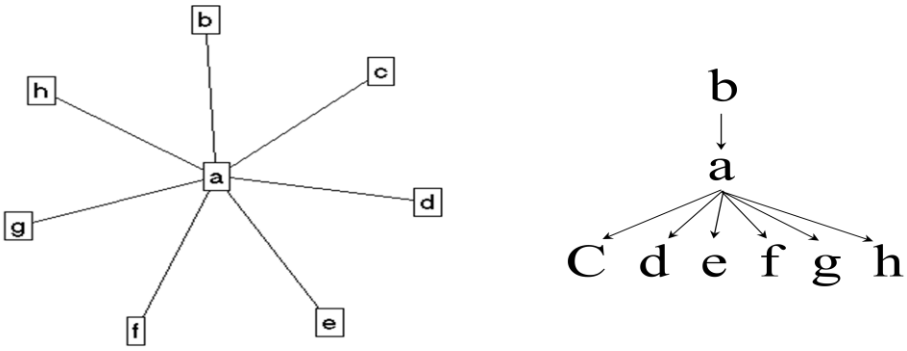


Figure (2.6): Betweenness Centrality

In figure (2.6), node (a) lies on the shortest paths between node (b) and the remaining nodes in the graph. For calculating the betweenness centrality, equation (6) is usually used.

$$C_B(n_i) = \sum_{j<k} g_{jk}(n_i)/g_{jk} \qquad (6)$$

Where: $C_B(n_i)$ betweenness centrality of node i, $g_{jk}(n_i)$ the number that actor *i* is on, g_{jk} the number of geodesics that is connecting jk

The betweenness centrality is usually normalized by equation (7):

$$C'_B(n_i) = C_B(n_i)/[(g-1)(g-2)/2] \tag{7}$$

Where: $C'_B(n_i)$ normalized betweenness centrality of node i, $C_B(n_i)$ betweenness centrality of node *i*, g the total number of nodes in the graph. Figure (2.7) shows examples of betweenness centrality calculation.

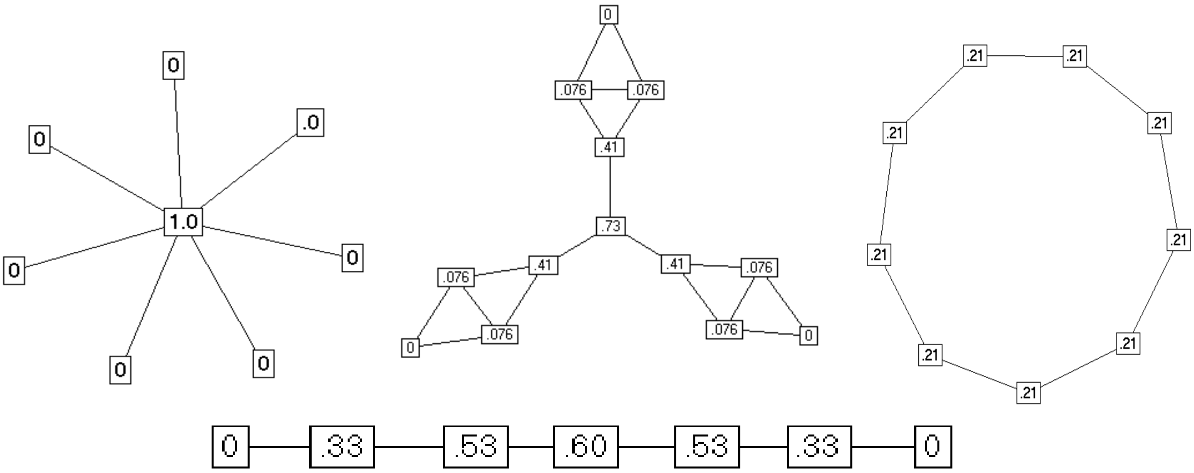


Figure (2.7): Betweenness Centrality calculation

- eigenvector centrality

Another common measure of centrality is eigenvector centrality (Bonacich, 1987) which measures the importance of a node in a network. Eigenvector centrality is defined as the principal eigenvector of the adjacency matrix defining the network. The defining equation of an eigenvector is:

$$\lambda v = A v \tag{8}$$

Where: A is the adjacency matrix of the graph, λ is a constant (the eigenvalue), v is the eigenvector.

The equation lends itself to the interpretation that a node that has a high eigenvector score is one that is adjacent to nodes that are themselves high scorers. Mathematically, eigenvector centrality is closely related to the influence measures. The idea is that even if a node influences just one other node, who subsequently influences many other nodes (who themselves influence still more others), then the first node in that chain is highly influential. In figure (2.8), The node diameter represents higher eigenvector centrality.

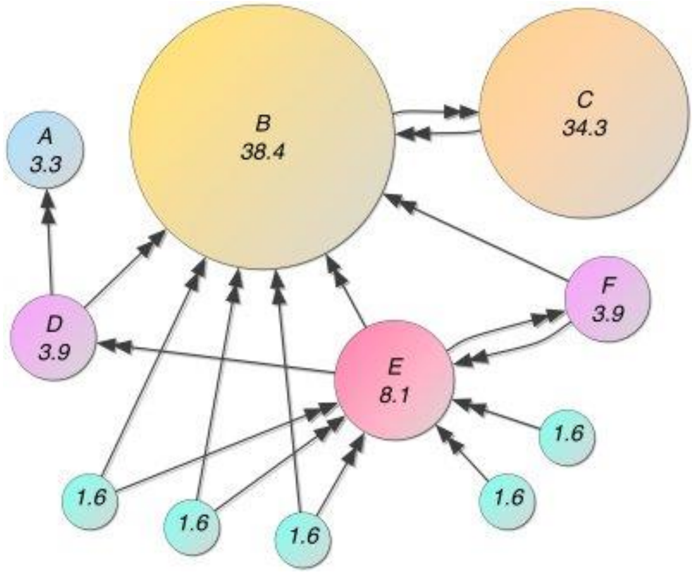


Figure (2.8): Eigenvector Centrality example

2.5.1.2. Prestige

Actor that receives many directed ties, but initiates few relations (popularity > extensivity).
Prestige measures of prominence apply only to directed graphs, taking into account the differences between sending and receiving relations, since for these measures, the direction is important property of the relation (Batagelj & Mrvar, 1999).

- Degree Prestige

Degree is one of the measures of prestige. According to meaning of relation, it represents support or influence (Batagelj & Mrvar, 1999). A prestigious actor enjoys high popularity, shown by receiving many ties from others. Hence, measure actor-level degree prestige as in-degree (in contrast to degree centrality which measures actor out-degree, above). To standardize the degree prestige index, divide out-degree by network size (g-1).

- Proximity Prestige

This analog to closeness centrality considers the proximity of actor i to other actors in its influence domain, the set of all network actors that can reach actor i, directly and indirectly (revealed by nonzero entries in the *i*th column of the distance matrix).

One proximity measure is average distance from influence domain actor j to actor i:

$$\sum \frac{d(n_j, n_i)}{I_i} \tag{9}$$

Because this measure ignores actors unable to reach actor i, proximity prestige can be calculated for unconnected graphs, but the values depend on the network's size.
A proximity prestige standardization suggested long ago by (Lin, 1976) reflects how proximate actor i is from the set of all actors:

$$P_P(n_i) = \frac{I_i/(g-1)}{\sum d(n_j, n_i)/I_i} \tag{10}$$

This index is the ratio between the proportion of actors in the influence domain to the average distance of these actors to actor i. If actor i is unreachable, PP = 0; if all actors are directly tied to actor i, PP = 1.

2.5.1.3. Other network measures

In order to measure the degree an individual is connected and reachable within a network, (Valente & Foreman, 1998) introduced integration measures (how well an individual is connected in the network) and radiality measures (how well an individual's ties reach out into the network).
Walling et al. (1999) identified some measures such as "transitivity", "characteristic path length" and "average number of contacts" which are relevant to airborne disease diffusion models. "Transitivity" is the average proportion of secondary contacts that also contact each other and "characteristic path length", which is defined as the number of contacts in the shortest path between two individuals in a network with the average taken over all possible pairs (Wallinga et al., 1999), where "average number of contacts" indicates how many people might potentially get the infection from an infected person(Wallinga et al., 1999).

2.6. software libraries that use social network analysis and features

1. **JUNG:** The JUNG (Java Universal Network/Graph) Framework is a free, open-source software library that provides a common and extendible language for the manipulation, analysis, and visualization of data that can be represented as a graph or network. It is written in the Java programming language, allowing JUNG-based applications to make use of the extensive built-in capabilities of the Java Application Programming Interface (API), as well as those of other existing third-party Java libraries (O'Madadhain et al., 2007). In JUNG, a number of algorithms are applied to analyze and measure network elements, such as calculation of network distances, flows, ranking measures, and measuring the "influence", "authority", or "centrality" of a given vertex/edge.
2. **Repast**: repast is a widely used free and open-source agent-based modeling and simulation toolkit. Three Repast platforms are currently available, namely, Repast for Java (Repast J), Repast for the Microsoft .NET framework (Repast .NET), and Repast for Python Scripting (Repast Py). Each of these platforms has the same core features. However, each platform provides a different environment for these features. Taken together, the Repast platform portfolio gives modelers a choice of model development and execution environments. Repast Simphony (Repast S) extends the Repast portfolio by offering a new approach to simulation development and execution. The Repast S runtime is designed to include advanced features for agent storage, display, and behavioral activation, as well as new facilities for data analysis and presentation.
3. **MORe:** it stands for (Managing Ongoing Relations). MORe is a newly developed software library which adds network functionality to Repast J framework. It uses several features of the recently released Repast Simphony framework, and proposes an extensible network measure framework, which enables the user to compute and analyze any network measure through GUI (graphical user interface) elements.

2.7. Visualization of social interaction networks

Visualization of the social interactions among a network plays an important role in SNAs to explore and study individual and groups activities (Caschera, Ferri, & Grifoni, 2008). Different approaches are proposed to explore the network components (nodes and edges), for example, in order to differentiate gender, age classes or ethnic groups of individuals within a network, Freeman (2005) proposed to use different shapes and colors to identify certain nodes within this network (Freeman, 2005). Generally, to visualize any network, three aspects should be followed: the substantive aspect of the network, the design and the algorithm will be applied to realize the design (Brandes & Wagner, 2003). Some visualization algorithms are such as Uniform layouts, Spectral layouts, Layered layouts and Radial layouts. As well as, Repast Simphony provides number of layouts for network visualization, in addition to MORe which extends and improves the visualization capabilities.

3. Conceptual framework

Chinn and Kramer (1999) define a concept as a "complex mental formulation of experience". Concepts are the major components of theory and convey the abstract ideas within a theory. While the theoretical framework is the theory on which the study is based, the conceptual framework is the operationalization of the theory.

This chapter is intended to present and describe the conceptual design for the extension of the pertussis model by linking the model with the social network analysis library.

3.1. Requirements for the model

Ultimate aim for implementing the social network is to pass messages from agents to agents. Disease is spreading via contact from person to person. This happens when two people are at the same location at the same time and one of them is infected and the other person is susceptible. However, there is also another mechanism and this is the fact that people can get information about the disease and will change their behavior to reduce the risk of getting infected. Important is that this information is not only passed via face to face contact (when two agents meet) but also in many other ways. This may be via contact from person to person other than when these two people meet, for example, via telephone, social media or e-mail. But, this may also be in the form of advertising campaigns organized by the government, where one message is send to many people and the contact is one directional. In this smaller degree this also applies to a school, sending a message to the parents of their children. This message is one directional, from one organization to many others.

Capturing the social interaction network is necessary to facilitate sending these types of messages.
Besides sending messages the agents in this model should be able to receive messages, store this information and change their behavior accordingly.
This requires that the adjustments of the model cover the following aspects:

- The display of the social interaction network according to different queries, such as the agents who have certain number of contacts.
- Model should facilitate the transfer of messages from a governmental body or public organization to a large number of agents.
- Model should facilitate the transfer of messages from a diseased his/her friends without these people meeting.
- Model should facilitate the transfer of messages from a person to another person (agent to agent) when these two agents meet.
- New agents should be introduced that represent the governmental agencies
- Agents should be enable to store and process "messages" they receive
- Agents should be enabled with rules about the change of behavior when the level of messages has exceeded a threshold.
- Model should be able to identify different types of social relationships.

In this conceptual chapter, the capturing of the social interaction network itself will be explained but also the adjustments to the agents in order to send and receive messages and the change of behavior that is based on this.

In order to incorporate any existing social interaction network model with software framework that shall analyze and visualize social networks during simulation, number of steps should be followed for the social network model:

1. Capture the social network elements.
2. Should be able to measure the social interaction of agents.
3. Visualize the social network of the pertussis simulation model.
4. Should be able to change the behavior of agents.

The main components are discussed according to the four main phases of the methodology described in chapter (1), capturing the social network elements, analyzing the social interaction of agents during the simulation, visualization of the social network of the disease model then to change agents' behavior and adapt their positions based on the measures. The four main stages are shown in figure (3.1):

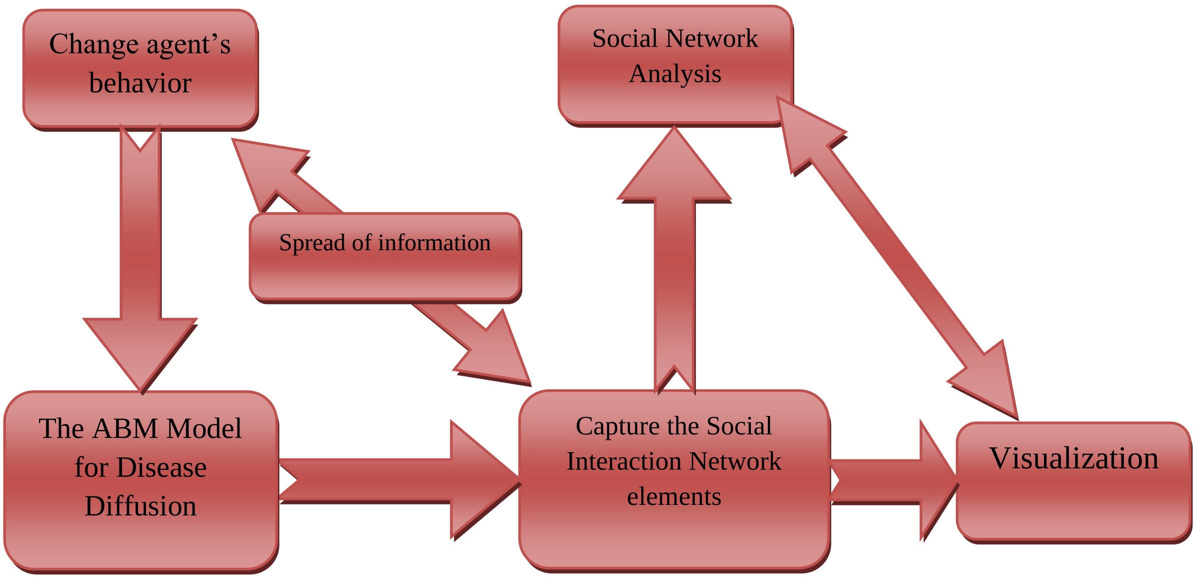

Change agent's behavior
Social Network Analysis
Spread of information
The ABM Model for Disease Diffusion
Capture the Social Interaction Network elements
Visualization

Figure (3.1): the main stages of the methodology

3.2. Capturing the social network

The conceptual framework of building models in Repast Simphony was improved to organize the way the agents are stored and represented, in addition to the environments where they behave and interact. Repast introduces context that host agents and are structured in a hierarchical approach. Contexts allow the agents to adjust themselves with respect to set of data, depending on which context they belong to (Tatara, 2007). Projections are responsible for the spatial arrangement of agents and involve different types such as continuous space, grid, and GIS projection as the current model is or networks.
The network context consists of a set of elements which comprises the network; those elements are objects to represent the agents of the model or nodes, and ties or edges to represent the relationships between those agents from network perspective. Networks in Repast are based upon both the edge and node interfaces and their implementing classes.

In order to build the network and represent different interactions between different agents, the model should be built in a dynamic manner, because during the simulation, the agent could meet different agents and has new relationships with them, or some breakable relationships could be changed during the simulation. So that in order to reflect the variant changes in the simulation, the model should be dynamic, to be able to add new agents, remove agents, add new edges between agents or remove existing edges during the simulation.

In order to capture the nodes of the social network elements from the disease model, the following steps need to be taken:

1. Enable the social context in the existing model.
2. The network context represents the agents as nodes stored in an array list.

The array list for representing the agents would include data about each node such as nominal data, health status, vaccination level, age group, current activity, color and size values that may optionally be adjusted by node properties. That means the user specifies a range for the shape size and the actual size changes according to a certain agent value like for example its degree centrality.

In order to capture the edges to represent the interaction between agents, the following choices were made. In the model, a number of interactions are found between different agents, even physical interactions (face to face contact) or non physical interactions (via phones, emails or any other conversational contact). All those interactions should be represented in the social relationship network. Different types of interactions between agents in the model will be represented by different types of edges; each edge type would represent certain interaction (such as family member, classmate, colleague…). In any network, different types of edges could be used, it may be directed, undirected, or mixed (Brandes & Wagner, 2003). The directed edges could be with one direction (like the "parent" will link to the "child" or "teacher" to her "students" in a class) or two directions "adult – adult" relation. But the problem occurs here since Repast Simphony does not allow for multiple types of edges at a certain relation, because when creating the network context, the user should identify the type of the network whether it is directed or undirected network. Another distinction between edges is that are already existing, these are unbreakable edges (the members of a household for example). There are also breakable edges that are considered to be permanent while simulating (the children in a school class or the people within a department at work) but there are non-permanent edges (people that meet each other by accident). These last types of edges are the edges that will be updated (created) during the simulation run, while the first two edges are loaded at the start of the simulation.

The edges between pairs of nodes are represented by adjacency matrix, in which nodes are listed both in the columns and rows. If there is a relation between two nodes, the value 1 is assigned to the field that represents the row node and the column node, and zero indicates that there is no edge between the particular nodes. For the case of undirected networks, the matrix is symmetric. For instance, table 3.1 shows the adjacency matrix of a directed network.

	Family 1 member 1	**Family 1 member 2**	**Family 2 member 1**	**Family 2 member 2**	**Family 3 member1**
Family 1 member 1	0	1	0	1	1
Family 1 member 2	1	0	0	1	0
Family 2 member 1	0	0	0	0	1
Family 2 member 2	1	0	0	0	0
Family 3 member1	0	1	1	0	0

Table (3.1): example of an adjacency matrix of a directed network

Since in the current pertussis model the number of agents reaches 60000 agents, the representation of nodes by an array list, as well as the representation of the links between those nodes by an adjacency matrix are unsuitable in case that large number of agents, in that case we need a matrix of size 60000 x 60000 to store all the relations between agents, considering that those relations are dynamic and would be changed during the simulation, so that representation would consume both memory and time to represent, which make this solution is not a good solution for large network.

To represent the large network with 60000 agents and the interactions between them, it is possible to imitate the representation provided with the software PAJEK, which is program for analyzing and visualizing large networks with some thousands or even millions of vertices (Batagelj & Mrvar, 1999). In Pajek, analysis and visualization are performed using 6 data types:

- Network main object (graph: vertices and lines);
- Permutation (reordering of vertices, ordinal properties) reordering of vertices;
- Vector (numerical properties of vertices) values of vertices;
- Cluster (subset of vertices) subset of vertices (e. g. one class from partition);
- Partition (nominal or ordinal properties of vertices) tells for each vertex to which cluster the vertex belongs; and
- Hierarchy (general tree structure on vertices) hierarchically ordered clusters and vertices (Batagelj & Mrvar, 2003).

In the disease model, it can be detected the flow of infection from illness individuals to susceptible persons during the simulation. In the social interaction network, the idea is to detect the flow of information about the disease from person to person, even by physical contact when they meet at school or workplace, or via non-physically contacts, such as from government to citizens by media or ad campaigns, or from school represented by the headmaster sends messages to children’s parents. In the network, different types of edges would be needed:

- Directed edges in two directions (bidirectional) in which information will pass equally in two directions, figure (3.2.a).
- Directed (unidirectional) edges in which information will spread in one direction but not in the other, as from parent to child, from government to adult or from school manager to parents, figure (3.2.b).

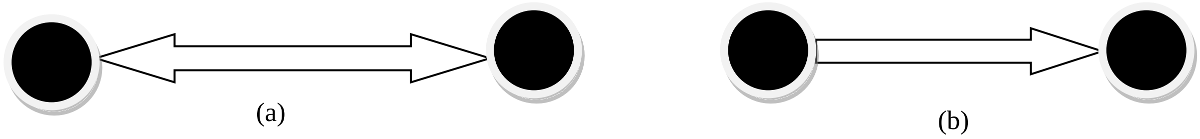


Figure (3.2): type of edges which spread information about disease

3.3. Messages

In order to collect information about any disease, receive them and broadcast it to others, different types of communication systems of agents could be realized. They work just like the five sense organs of human beings. In her research, Zhou (2006) introduced sensor and broadcaster as communication system main types. The task of sensors is to collect information about the disease as eyes or ears function (Zhou, 2006), when a broadcaster as a human being's mouth, facilitates to broadcast information about disease to others, via phone or any other conversational media, or such the governmental task when it announces about a disease outbreak.

3.3.1. Adjustments to agents

Conceptually, new agents should be introduced to the model. The government is an example of such an agent. It has the behavior to check the network if the disease level reaches a certain stage the government can send messages to warn the public or announce a vaccination campaign. The government will not use the social network but simply send messages to all agents or subsets of agents (campaign may target a certain subset in the population). Individual agents can be "sensitive" to the messages from the government, or not. The message has only one direction. School agent represents the director or head supervisor of the school who can send messages to all the parents of the children that go to this school to warn them. This type of messaging is also not using the social network but is like the government but for a more particular set of agents. This message has only one direction.

3.3.2. types of messages

There are different types of messages and for all messages apply that some of the agents will receive the message and others will not. Perhaps a random group can be retrieved from the total agent population. For governmental messages, they are broadcasted in one direction. From school agent to all children's parents, messages are also broadcasted in one direction. All parents should receive the message.

Messages can also be transferred when two people meet. This is already in the current model. In the current model people can transfer the disease but they do to spread warnings. Then people that are ill can broadcast a message to their family members and friends.

3.4. Behavior change

Change of behavior also would be arisen in the network due to the information about the disease that might pass through the network of agents.

There is different ways to induce the change of behavior (Funk et al., 2010):

- Warning from the government about an epidemic to affect the behavior of people to avoid the infection, or increasing for example the vaccination level.
- Direct contact between people would lead to affect the behavior.
- Indirect contact also affects the change of behavior.

Figure (3.3) illustrates a flow chart of different actions will be taken by agents when information about the disease are spreading over the network.

The sequences diagram in figure (3.4) shows the sequence of actions when someone becomes ill. This agent will pass a message to certain parts of its network. Government and schools can also send a message (unidirectional messages). Person that receives a message can pass it on.

- Agents need an extra attribute “awareness level” this includes two mechanisms; it is a numeric value that increases when agents receive a warning message and the second mechanism is that it will go down a little bit with every tick.
- Agents need to have a “threshold value”. This value can differ per agent. When the “awareness level” reaches the threshold the agents will take action
- The agents will need to know which action he/she needs to perform when the threshold has been reached. This can be “get vaccinates” or “avoid contact”. Avoid contact can mean that healthy children do not go to school for example.
- Not every message has the same impact. Some messages will immediately increase the “awareness” to the maximum and others will only give a small increase.
- Randomness may also be used. Perhaps if people pass on the message is determined by a random factor.

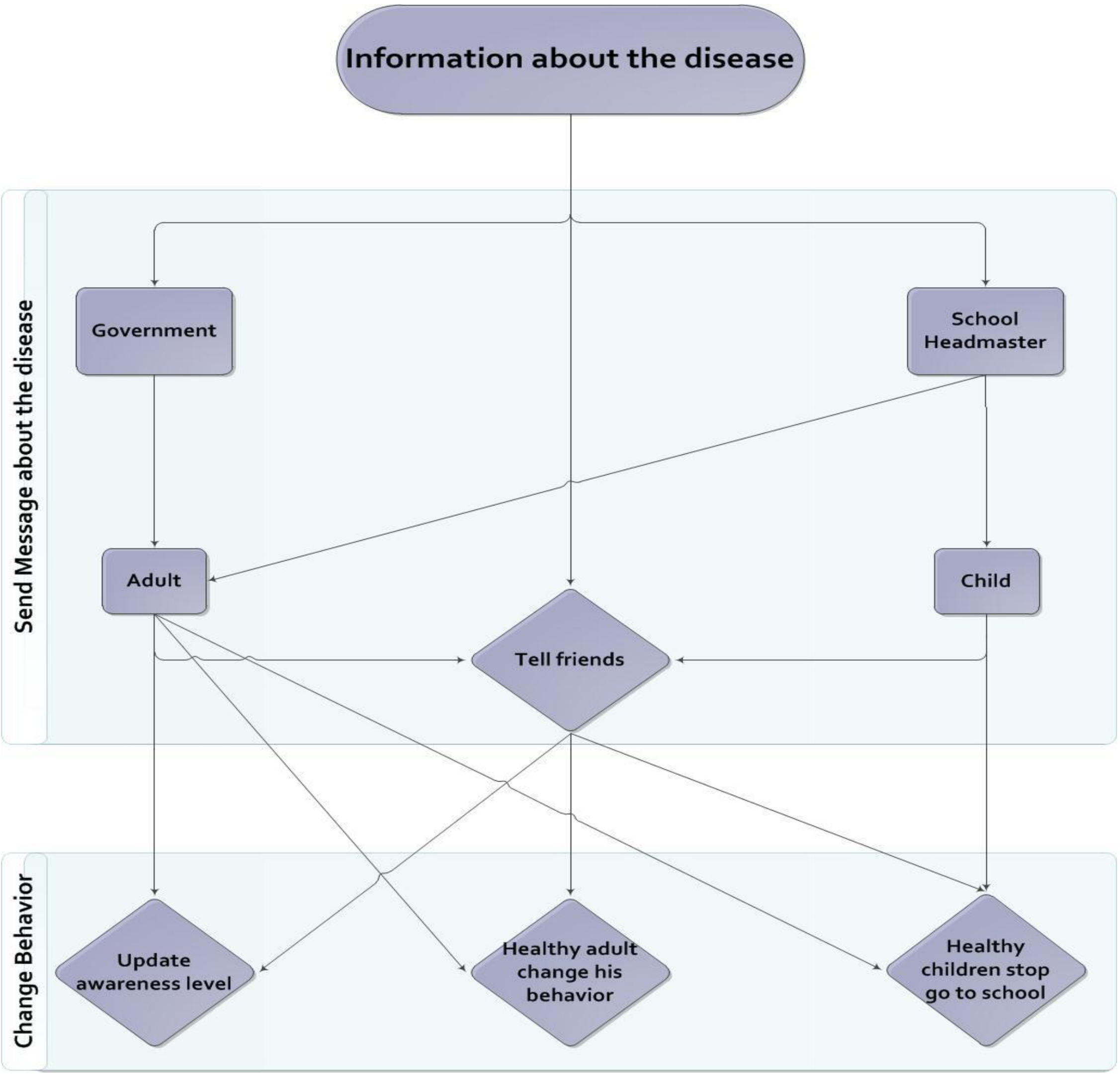

Information about the disease
Send Message about the disease
Government
School Headmaster
Adult
Child
Tell friends
Change Behavior
Update awareness level
Healthy adult change his behavior
Healthy children stop go to school

Figure (3.3): change behavior when there is information about disease are spreading over the network

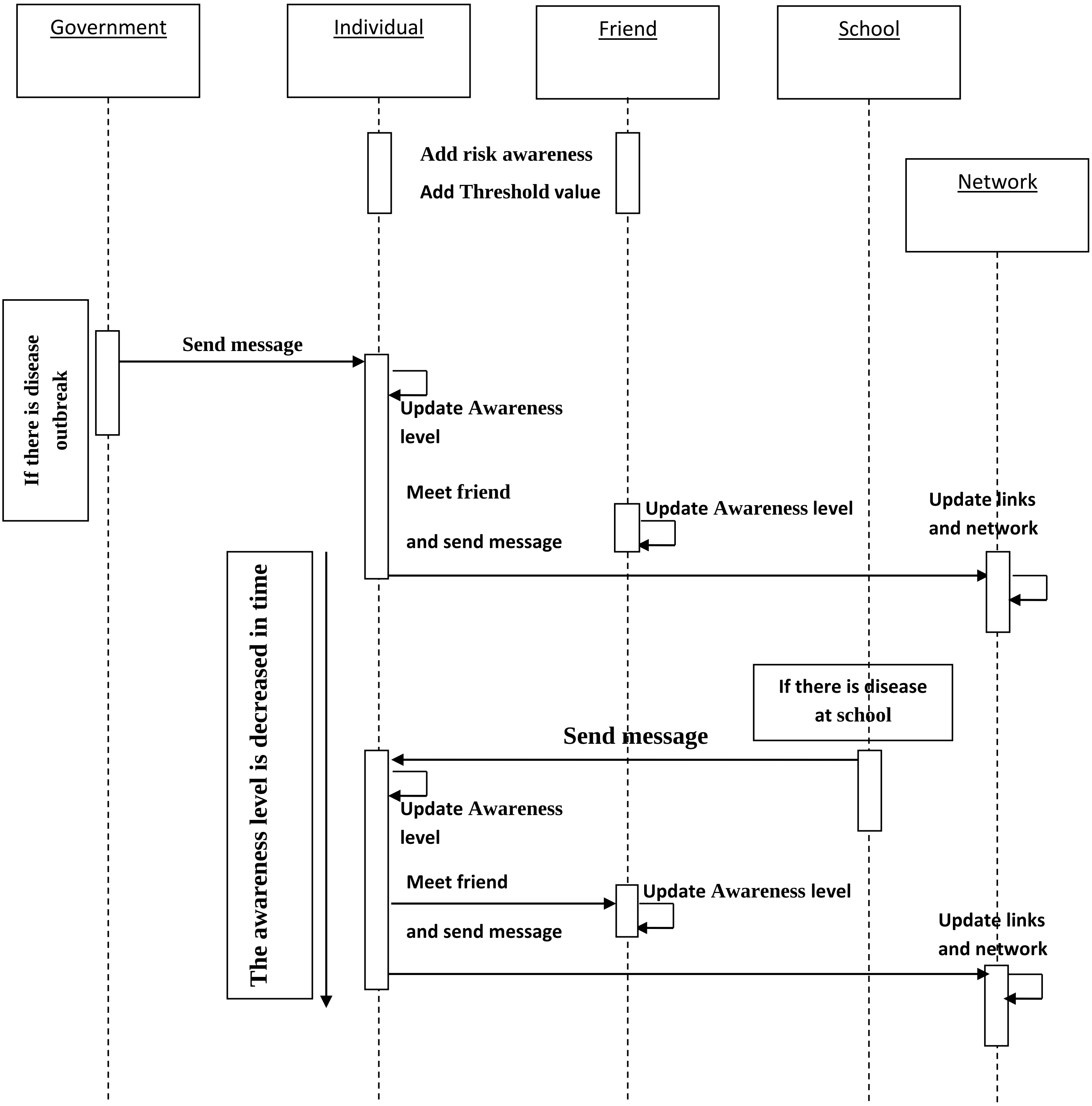


Figure (3.4): Sequence diagram: updating network visualization

3.5. social network analysis

The purpose of using the social network analysis is to validate the synthetic population generated by the pertussis model and its social interactions. These measures could be used to analyze the possibility of a message to reach big number of contacts when it passes over the network.

Most important measures are used to analyze the flow of information over a network: the number of links an agent has (degree of centrality), so that it measures the number of contacts an agent has to broadcast the message, and the maximum number of links people is apart to get a message about a disease and increase the awareness level accordingly.

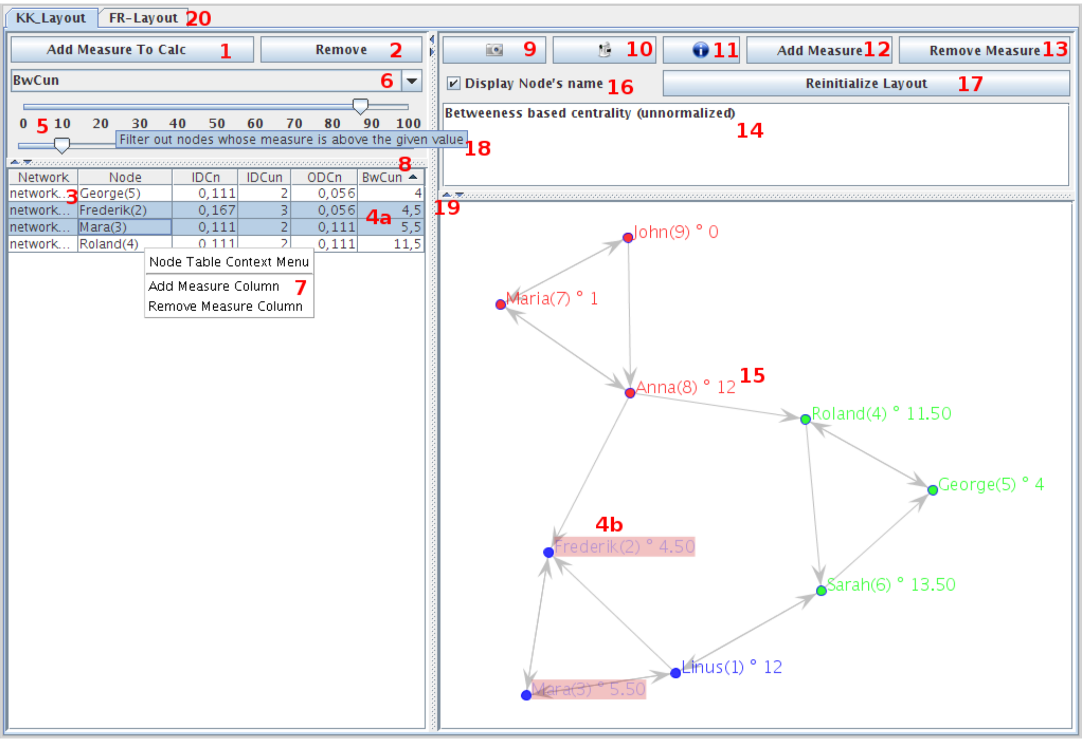
Import the software library that contains all java interfaces and classes that help to analyze and visualize the social network model during the simulation and apply certain measures for the analysis of the network components in order to implement the most appropriate measures that are relevant to the case of disease diffusion.

By incorporating the SNA libraries, it could be applicable to use the features found in the library to analyze and check the possibility of certain agent to play a role in order to pass messages about the disease to a large number of persons who are with a direct contact or even secondary contact with that agent.

As discussed in chapter (2), social network analysis contains large number of analysis tools and measures to check the features of network and its elements.

In the model, a number of measures were selected to analyze the network elements and were relevant to the disease case and passing information about the disease. Those measures as discussed in chapter (2) are degree centrality which calculates the number of direct contacts each agent has with other agents. This measure considers the importance degree of an agent to transmit information about the disease to other agents. Closeness centrality calculates the inverse of the sum of geodesic distances from agent to the other agents in its network. Betweenness centrality which counts the number of shortest paths between agent i and agent k that agent j resides on.

Figure (3.5) shows an example of the interface for applying network measures and how it would be illustrated.



Network	Node	IDCn	IDCun	ODCn	BwCun
network...	George(5)	0,111	2	0,056	4
network...	Frederik(2)	0,167	3	0,056	4,5
network...	Mara(3)	0,111	2	0,111	5,5
network...	Roland(4)	0,111	2	0,111	11,5

Figure (3.5): screenshot of elements of the user interface

3.6. Visualization

After enabling the network context, capturing the social network elements and establishing the nodes and the edges between corresponding nodes, a display of the network projection need to be established in order to show the agents as nodes and the relationships between those agents as edges.

During the visualization, the network will be updated dynamically when a node is attached to the network or removed from the network, as well as when a connection is established between pairs of nodes or a connection is removed, the network should be updated accordingly. Furthermore the measures values for the whole nodes in the network will be recalculated. Figure (3.6) shows the sequence diagram of updating a network display.

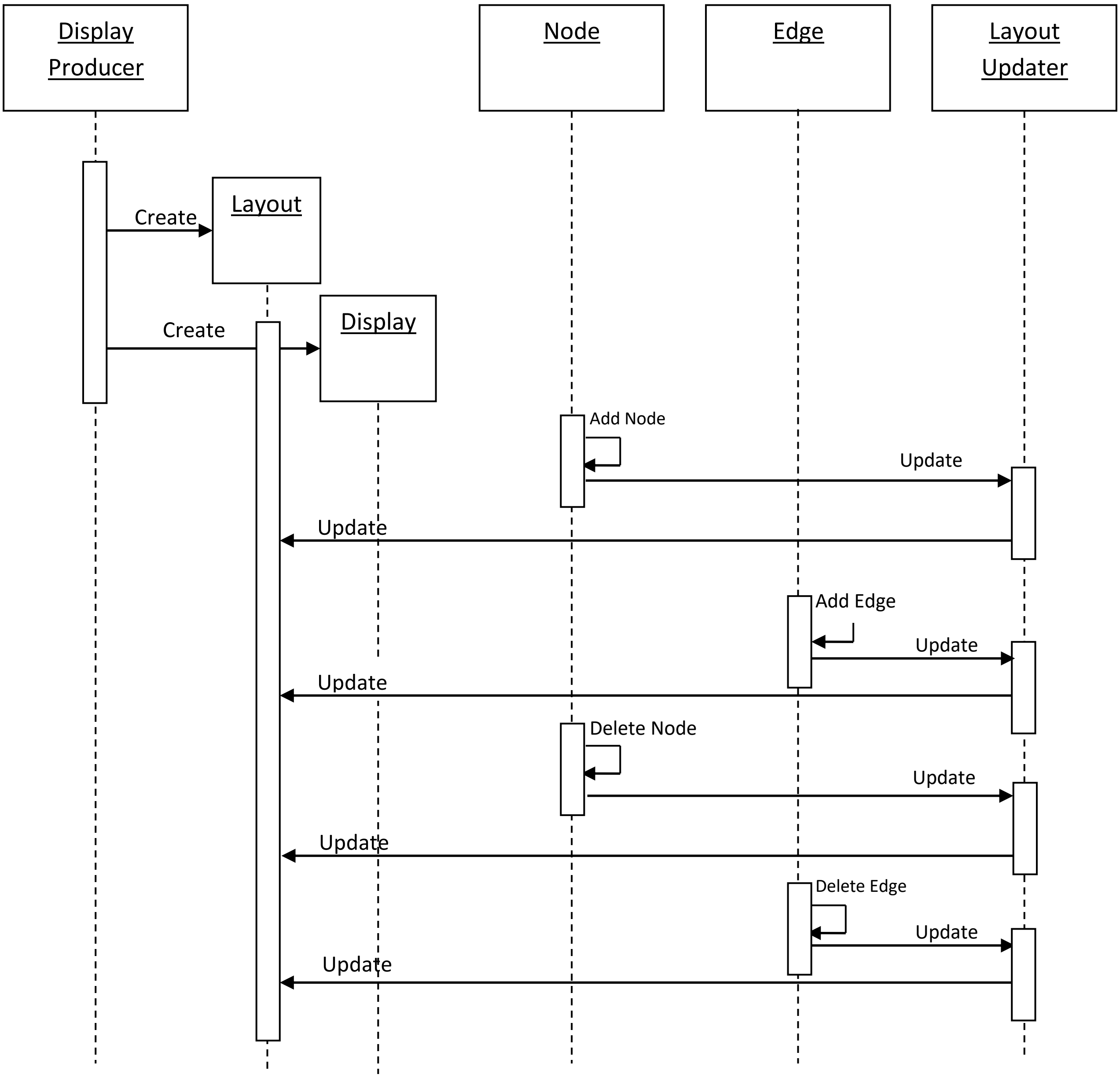


Figure (3.6): Sequence diagram: updating network visualization

Figure (3.7) illustrates the stages from capturing the network elements to the display of the corresponding network.

In the case of huge number of nodes as in the disease model, there will be a problem when trying to display all nodes in one display. It gives such a cluttered view so that it is very difficult to see the network elements. Different criteria could be applied on the nodes in order to get an appropriate view. One could to use different type of layouts. In table (3.2), it can be shown the available layouts for network visualization and the advantages of each layout according to the network size and the complexity of the applied algorithm to display the layout (Holzhauer, 2009). Another criterion is to choose only the nodes that attain certain threshold, such as the nodes that have degree of centrality of 4 or more, display only the nodes which belong to certain neighborhood, or for example as is shown in figure (3.7) display the male agents and female agents with different colors and the diseased agent with another color.

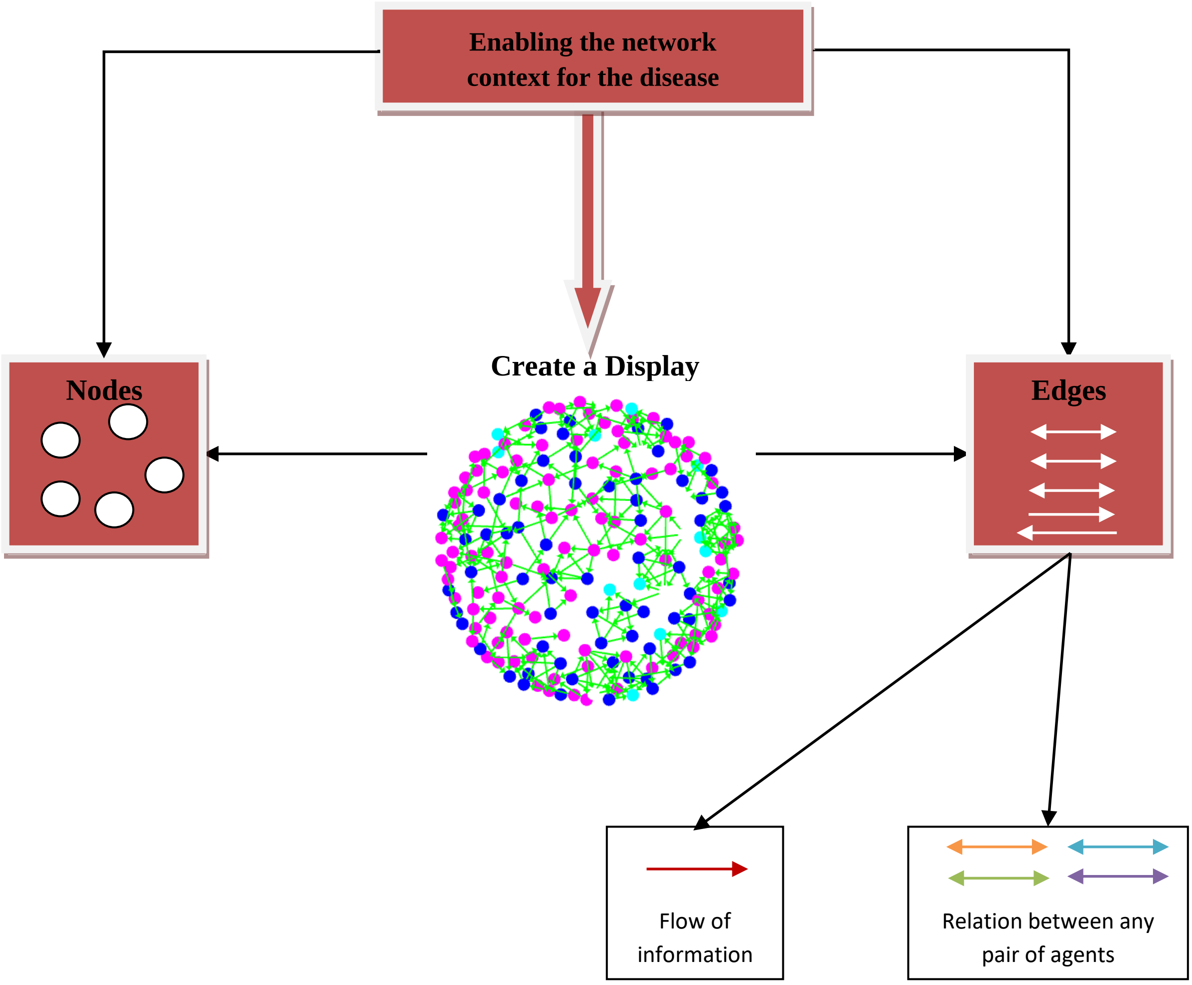
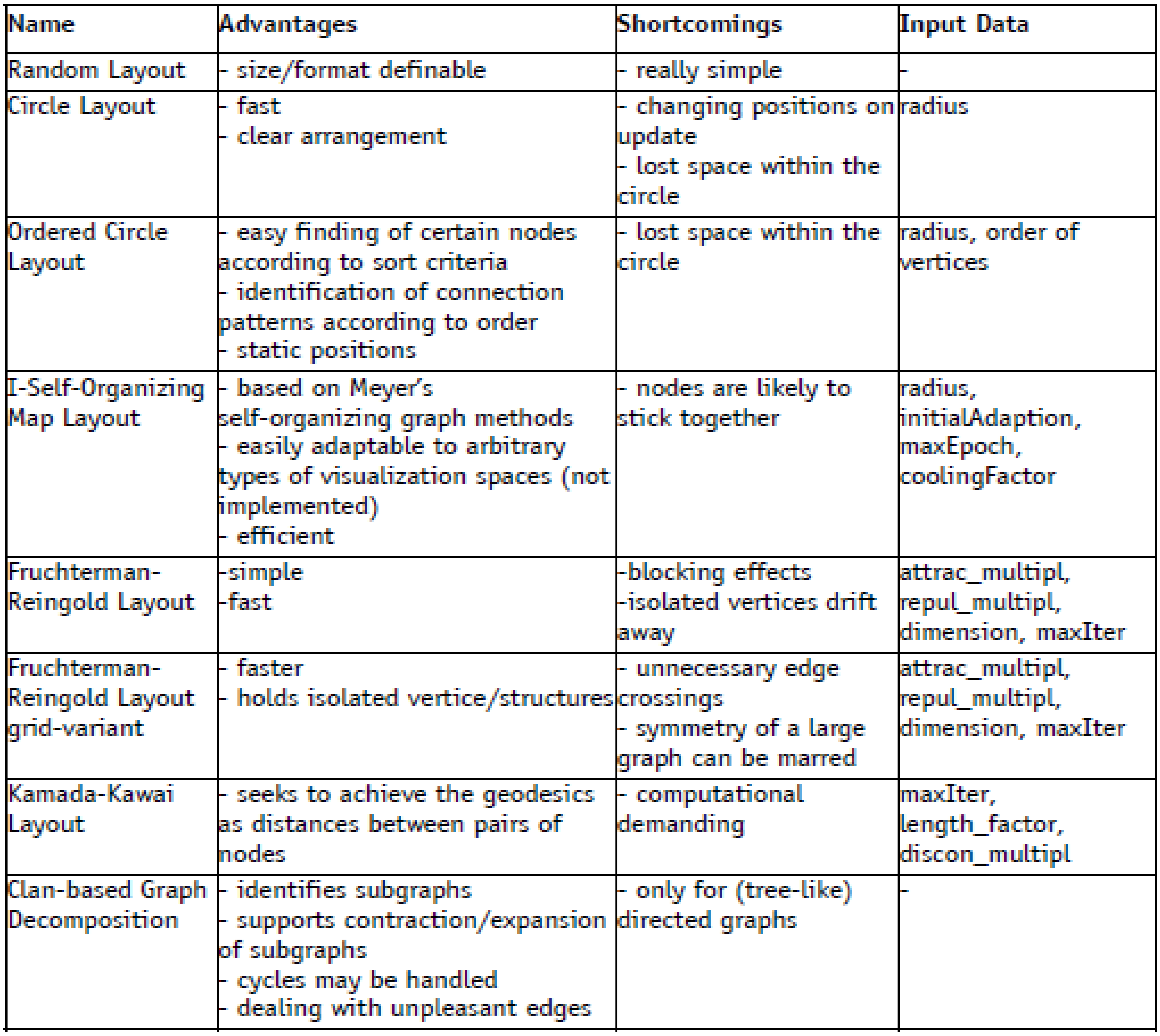

Enabling the network context for the disease
Create a Display
Nodes
Edges
Flow of information
Relation between any pair of agents

Figure (3.7): flowchart describe the steps of stage (1)



Name	Advantages	Shortcomings	Input Data
Random Layout	- size/format definable	- really simple	-
Circle Layout	- fast - clear arrangement	- changing positions on update - lost space within the circle	radius
Ordered Circle Layout	- easy finding of certain nodes according to sort criteria - identification of connection patterns according to order - static positions	- lost space within the circle	radius, order of vertices
I-Self-Organizing Map Layout	- based on Meyer's self-organizing graph methods - easily adaptable to arbitrary types of visualization spaces (not implemented) - efficient	- nodes are likely to stick together	radius, initialAdaption, maxEpoch, coolingFactor
Fruchterman-Reingold Layout	-simple -fast	-blocking effects -isolated vertices drift away	attrac_multipl, repul_multipl, dimension, maxIter
Fruchterman-Reingold Layout grid-variant	- faster - holds isolated vertice/structures	- unnecessary edge crossings - symmetry of a large graph can be marred	attrac_multipl, repul_multipl, dimension, maxIter
Kamada-Kawai Layout	- seeks to achieve the geodesics as distances between pairs of nodes	- computational demanding	maxIter, length_factor, discon_multipl
Clan-based Graph Decomposition	- identifies subgraphs - supports contraction/expansion of subgraphs - cycles may be handled - dealing with unpleasant edges	- only for (tree-like) directed graphs	-

Table (3.2): layouts for network visualization

4. Evaluation of Social Network analysis software

Libraries in general are software collections that implement features for specific uses and topics. Libraries support the modular assembly concept, since they provide code that is invoked partly or as a whole by other applications. For network analysis, there are libraries that display and layout graphs, calculate measures constitute general graph frameworks, or provide classes for efficient data representation.

There are number of libraries that are useful to be incorporated in either stand-alone applications or frameworks. The functionality the libraries were developed for reaches from general visualization to network analysis. This chapter evaluates a number of those libraries which play a significant role in the analysis and visualization of social networks.

Moreover, many diverse stand-alone applications have emerged in the field of SNA. Nearly all of them provide basic network measure calculations and descriptive statistics. Import and export functionality of network data exists. The majority of tools also integrate means to visualize networks with varying extent. Apart from such basic features, the applications are mostly intended for specific purposes like aggregating parts of a network to super nodes, enhanced visualizations, or dealing with large networks.

The following sections review a number of software libraries which are common in network analysis and visualization.

4.1. Social network software - a closer look

In this section, the programs UCINET, PAJEK, VISONE, JUNG and MORe are investigated in more details. It focuses on three groups of procedures the software does or does not possess.

- Data model and structure.
- Visualization techniques.
- Social network analysis routines which consists of descriptive methods to calculate (simple) network statistics (e.g., centrality or transitivity).

4.1.1. UCINET

UCINET 6.0 (S.P Borgatti, Everett, & Freeman, 2002) is a comprehensive program for the analysis of social networks and other proximity data. It is probably the best known and most frequently used software package for the analysis of social network data and contains a large number of network analytic routines. Two kinds of output are generated: textual output saved in log files and displayed on the screen, and data sets that can be used as input for other procedures.

4.1.1.1. Data model and structure

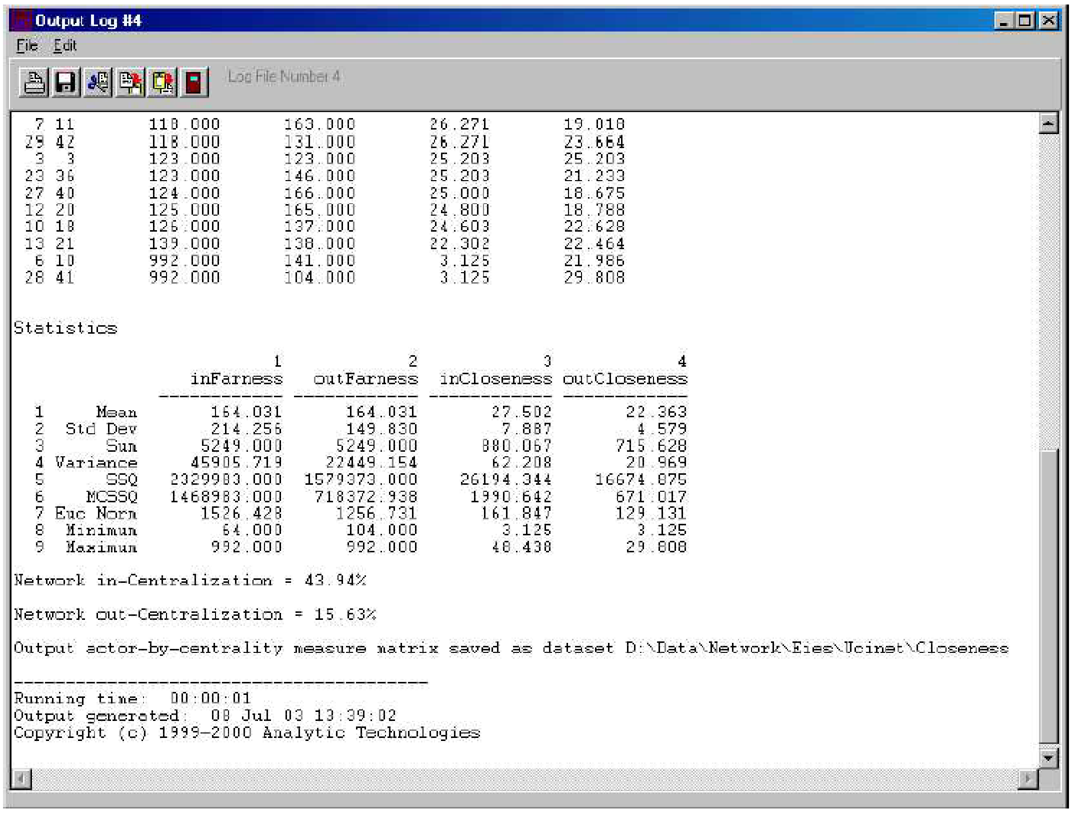
UCINET is matrix oriented, that is, data sets are collections of one or more matrices. A single UCINET data set consists of two files: one containing the actual data and one is containing information about the data. UCINET data sets can be created by importing data or by entering data directly via the built-in spreadsheet.

4.1.1.2. Visualization techniques

UCINET contains graphical tools to draw scatterplots, and tree diagrams, which can be saved as bitmap files (BMP). The program itself does not contain graphical procedures to visualize networks, but it has a speed button to execute the program NetDraw (S.P Borgatti, 2002), which reads UCINET files natively. NetDraw, developed for network visualization, has advanced graphical properties.

4.1.1.3. Descriptive methods

The program contains a large number of network analytic routines for the detection of cohesive subgroups, for centrality analysis, and for ego network analysis. As an example, the output of a centrality analysis is presented in Figure 4.1. For each node, it contains the in- and out-farness (the sum of the lengths of the geodesics to and from every other node), and the in- and out-closeness centrality (the reciprocal of farness times g - 1, with g the number of actors), some descriptive statistics, as well as Freeman's group closeness index (Freeman, 1979).



```
 7 11      118.000      163.000       26.271       19.018
29 42      118.000      131.000       26.271       23.664
 3  3      123.000      123.000       25.203       25.203
23 36      123.000      146.000       25.203       21.233
27 40      124.000      166.000       25.000       18.675
12 20      125.000      165.000       24.800       18.788
10 18      126.000      137.000       24.603       22.628
13 21      139.000      138.000       22.302       22.464
 6 10      992.000      141.000        3.125       21.986
28 41      992.000      104.000        3.125       29.808


Statistics

                        1            2            3            4
                inFarness   outFarness  inCloseness outCloseness
             ------------ ------------ ------------ ------------
  1     Mean      164.031      164.031       27.502       22.363
  2  Std Dev      214.256      149.830        7.887        4.579
  3      Sum     5249.000     5249.000      880.067      715.628
  4 Variance    45905.719    22449.154       62.208       20.969
  5      SSQ  2329983.000  1579373.000    26194.344    16674.875
  6    MCSSQ  1468983.000   718372.938     1990.642      671.017
  7 Euc Norm     1526.428     1256.731      161.847      129.131
  8  Minimum       64.000      104.000        3.125        3.125
  9  Maximum      992.000      992.000       48.438       29.808

Network in-Centralization = 43.94%

Network out-Centralization = 15.63%

Output actor-by-centrality measure matrix saved as dataset D:\Data\Network\Eies\Ucinet\Closeness

----------------------------------------
Running time:  00:00:01
Output generated:  08 Jul 03 13:39:02
Copyright (c) 1999-2000 Analytic Technologies
```

4.1.2. PAJEK

PAJEK (Batagelj & Mrvar, 2003) is a network analysis and visualization program, specifically designed to handle large data sets. The main goals in the design of Pajek are:

- to facilitate the reduction of a large network into several smaller networks that can be treated further using more sophisticated methods,
- to provide the user with powerful visualization tools, and
- to implement a selection of effcient network algorithms (Batagelj & Mrvar, 1999).

PAJEK can handle multiple networks simultaneously, as well as very large networks can be analyzed, with more than one million nodes. (The available memory on the computer sets the actual limit.

Large networks are hard to visualize in a single view. Therefore meaningful substructures have to be identified, which can be visualized separately. The algorithms implemented in Pajek are especially designed for this purpose (Batagelj & Mrvar, 2003). Pajek uses six different data structures:

1) networks (nodes and arcs/edges),
2) partitions (classifications of nodes, where each node is assigned exclusively to one class),
3) permutations (reordering of nodes),
4) clusters (subsets of nodes),
5) hierarchies (hierarchically ordered clusters and nodes), and
6) vectors (properties of nodes).

Partitions contain discrete attributes of nodes, whereas vectors contain continuous attributes. The structure of the program is entirely based on these six data structures and on transitions among these structures. The results generated by the procedures are usually presented using the data structures (instead of graphical or tabular output), and can be used as input in other procedures such as visualization methods.

4.1.2.1. Data model and structure

Network data can be entered in four ways:

1) by defining a (small) network inside the program,
2) by importing ASCII network data from network files (extension NET),
3) by importing data from software packages with other formats (e.g., UCINET DL- files and formats of some visualization programs), and
4) by opening a Pajek project file (PAJ), which combines all different data structures into a single file.

The NET- files consist of a node list and arcs/edges list, aimed at entering large networks more efficiently, specifying only the existing ties. For small networks the link list can be replaced by an adjacency matrix.

4.1.2.2. Visualization techniques

The graphical properties of Pajek are advanced. The Draw window gives the user many options to manipulate the graphs (layout, size, color, spin, etc.). Moreover, graphical representations of partitions, vectors, and combinations of partitions and vectors can be obtained. The network drawing is based on the principle that distances between nodes should reveal the structural patterning of the network

(Freeman, 2004). Besides simple layouts (circle, random), Pajek has several automatic procedures to find optimal layouts: procedures using eigenvectors, special procedures for layer drawing of acyclical networks, and spring embedders. The latter procedures are called so, because in those algorithms it is assumed that the nodes are connected by springs, whose stress is to be minimized.

Pajek uses two spring-embedding algorithms to visualize network data: the Kamada-Kawai and the Fruchterman-Reingold algorithms. The former one produces more stable results, but is slower and less suited for large networks. The latter algorithm is faster and can handle large networks. Both are optimization procedures that do not yield the same mapping each time they are run. The graphs, however, should resemble each other largely.

4.1.2.3. Descriptive methods

Each data object in Pajek has its own descriptive methods. The largest number of methods is available for networks, for instance, computation of degrees, depths, cores, or cliques (output is a partition), centrality (closeness, betweenness), detection of components (weak, strong, biconnected, symmetric), paths, or flows, structural holes, and some binary operations on two networks. The menu Info gives general characteristics of each data structure.

Computing closeness centrality with Pajek is straightforward. The network has to be dichotomized before calculating the closeness. For directed graphs, the in- or out-closeness can be calculated as well as the closeness for the symmetrized network (Huisman & Duijn, 2003).

4.1.3. VISONE

The visone project team (Brandes & Wagner, 2003) is developing models and algorithms to integrate and advance the analysis and visualization of social networks and is intended to be used in research and teaching.

Visone contains several different algorithms for drawing graphs and representing results of analysis. It uses uniform layouts, spring embedders, spectral layouts, layered layouts, and radial layouts to present networks. It has many options to improve the (layout of the) graphs and visualizations can be exported in SVG or postscript format. The analysis methods include local measures (degree, indegree, outdegree), distance measures (e.g., betweenness, closeness, eccentricity, radiality), and feedback measures (e.g., status, eigenvector, pagerank, authority, hub) (Brandes & Wagner, 2003).

4.1.4. JUNG

The Java Universal Network/Graph (JUNG) framework is a software library that provides a common and extendible language for the modeling, analysis, and visualization of data that can be represented as a graph or network. JUNG supports a variety of representations of graphs (e.g., directed, undirected) and the current version includes algorithms for clustering, decomposition, random graph generation, statistical analysis, and calculating of network distances, ows, and importance measures (O'Madadhain et al., 2007).

It also provides a visualization framework to construct tools for data exploration. JUNG is a framework on which applications and tools for manipulating graph and network data can be built. It can be used in simple snippets of code to test ideas, or to aid in the development of a sophisticated tool with a graphic user interface. JUNG is not itself a standalone tool, but rather a library that can be used to support the construction of specific tools. Building a tool that uses JUNG requires some knowledge of Java programming. The JUNG distribution does include samples of small applications that use JUNG to accomplish certain tasks, but they are intended to be examples of how one might use JUNG, not tools in their own right.

As a library, JUNG can be used both to build network-oriented tools, and to provide network capabilities to existing systems.

The major features of JUNG include the following:

- Support for a variety of representations of entities and their relations, including directed and undirected graphs, multi-modal graphs (graphs which contain more than one type of vertex or edge), graphs with parallel edges (also known as multigraphs), and hypergraphs (which contain hyperedges, each of which may connect any number of vertices).
- Mechanisms for annotating graphs, entities, and relations with metadata. These capabilities facilitate the creation of analytic tools for complex data sets that can examine the relations between entities, as well as the metadata attached to each entity and relation.
- Implementations of a number of algorithms from graph theory, exploratory data analysis, social network analysis, and machine learning. These include routines for clustering, decomposition, optimization, random graph generation, statistical analysis, and calculation of network distances, flows, and ranking measures (centrality, PageRank, HITS, etc.).
- A visualization framework that makes it easy to construct tools for the interactive exploration of network data. Users can choose among the provided layout and rendering algorithms, or use the framework to create their own custom algorithms.
- Filtering mechanisms which extract subsets of a network; this allows users to focus their attention, or their algorithms, on specific portions of a network.

4.1.4.1. Data model and structure

in many cases, network analysts will have text files, or databases, which contain the data to be analyzed; in addition, analysts may wish to create a permanent record of a network that they have generated or modified. To these ends, JUNG supports several input and output formats, and provides general mechanisms that allow users to create their own formats.

- Text Files

Pajek (Batagelj & Mrvar, 2003) is a tool that is popular in the social network analysis community, which has its own (complex) file format. JUNG can currently interpret the portions of the Pajek format that define directed, undirected, and mixed-type graphs; string (text) labels for vertices; and numeric edge weights. The format can also represent time-series and labeled vertex partition information, but JUNG does not as yet interpret these parts of the format.

Data is often provided in text files that are not in a common format. In such cases, developers have two choices: convert the text files into a common format, or write a parser for the existing format. Because there are many different possible text data formats, JUNG cannot provide a parser that works for all text data formats.

- Databases

Network data may sometimes be stored in a database; this facilitates certain kinds of “flat” statistical analysis and retrieval, but is not conducive to most methods for network analysis.

4.1.4.2. Visualization techniques

JUNG provides mechanisms for laying out and rendering graphs. The current renderer implementations use the Java Swing API to display graphs, but they may be implemented using other toolkits (such as SWT). In general, a visualization requires one of each of the following:

- A Layout, which takes a graph and determines the location at which each of its vertices will be drawn.
- A (Swing) Component, which provides a “drawing area” upon which the data is rendered.

JUNG provides a VisualizationViewer class for this purpose, which is an extension of the Swing JPanel class. A currently available experimental version of VisualizationViewer allows the user to create a “window” on the graph visualization, which can be used to magnify (zoom in on) portions of the graph, and to select different areas for magnification (panning).

- A Renderer, which takes the data provided by the Layout and paints the vertices and edges into the provided Component.

4.1.4.3. Descriptive methods

JUNG provides several different categories of network algorithms, such as ranking, Clustering and Blockmodelling, Topology, Paths, and Flows, etc…

Ranking algorithms assign values to each vertex (or edge) according to a set of criteria that reflect structural properties of the network. These criteria are generally intended to measure the “influence”, “authority”, or “centrality” of a given vertex/edge.

Several of these algorithms interpret their input as a Markov network: a directed weighted graph in which the vertices represent states, the edges represent possible state transitions, and the edge weights represent transition probabilities. (A directed graph with non-negative weights can be converted into a Markov network by normalizing the weights on the outgoing edges from each vertex to sum to 1.)

Ranking algorithms that JUNG provides include BetweennessCentrality (Brandes, 2001), which labels each vertex and edge in a graph with a value that is derived from the number of shortest paths that pass through it.

4.1.5. MORe

MORe stands for (Managing Ongoing Relations). MORe is a newly developed software library which adds network functionality to Repast J framework. It uses several features of the recently released Repast Simphony framework, and proposes an extensible network measure framework, which enables the user to compute and analyze any network measure through GUI (graphical user interface) elements.

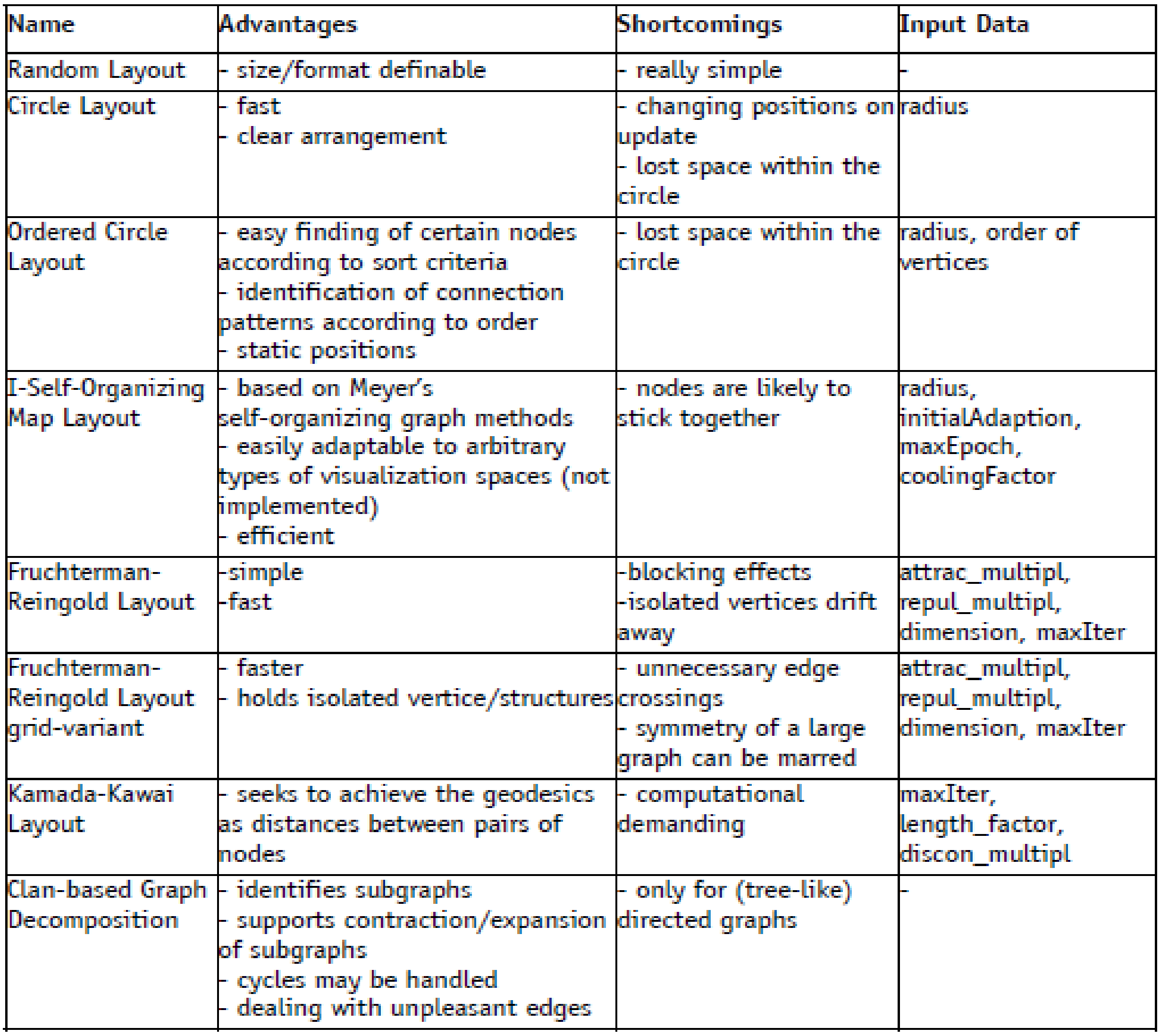
4.1.5.1. Data model and structure

As mentioned in shapter 3, the nodes stored Repast J in an array list which would include data about each node such as nominal data, color and size values that may optionally be adjusted by node properties.

While the edges between pairs of nodes are represented by adjacency matrix, in which nodes are listed both in the columns and rows. If there is a relation between two nodes, the value 1 is assigned to the field that represents the row node and the column node, and zero indicates that there is no edge between the particular nodes. For the case of undirected networks, the matrix is symmetric.

4.1.5.2. Visualization techniques

In MORe, number of network drawing techniques are available. Table (4.1) describes the different layout characteristics in order to decide which one is the most appropriate for a certain purpose, in addition to description of each visualization layout advantages, shortcomings and inpit data.



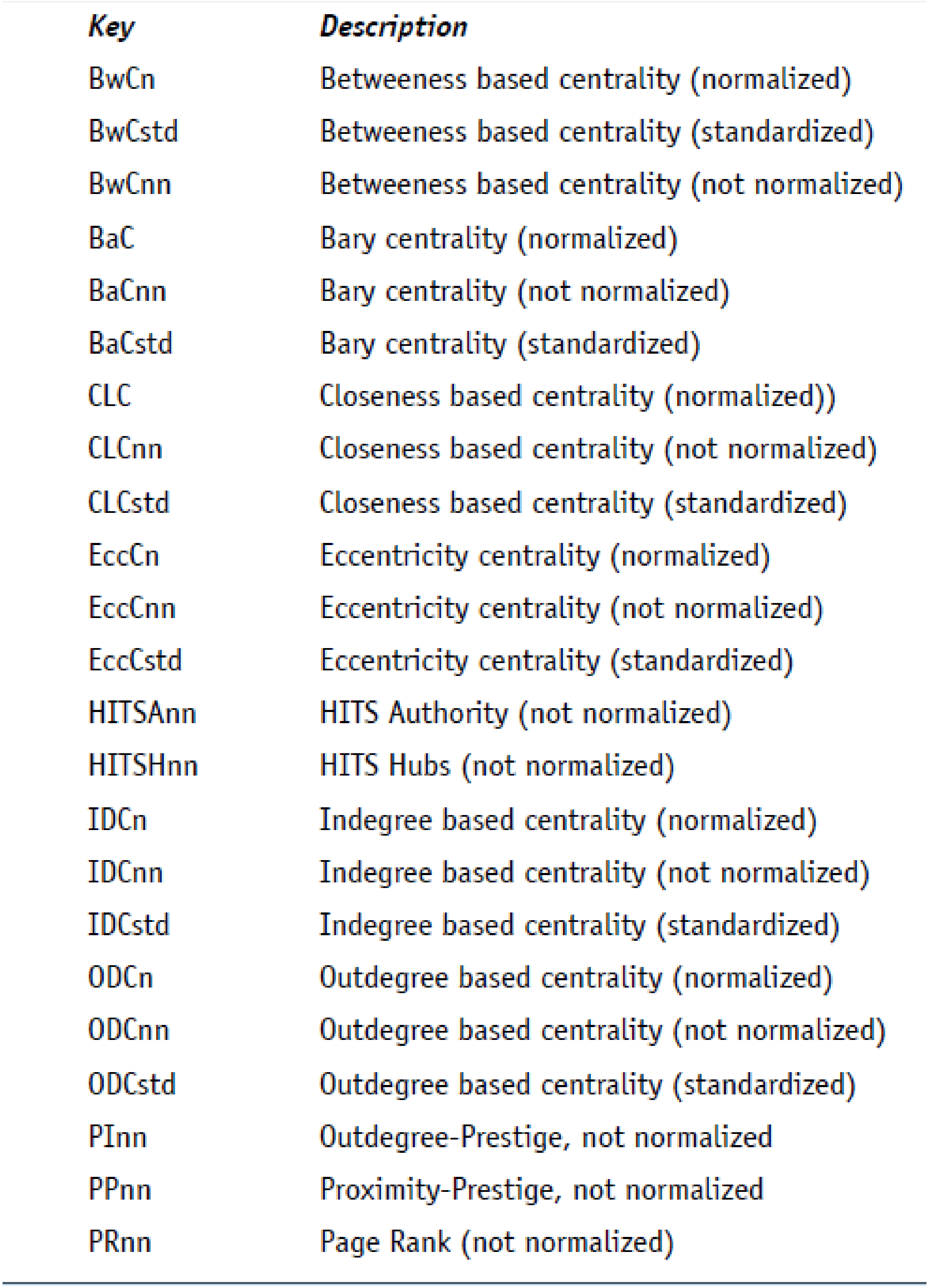
Name	Advantages	Shortcomings	Input Data
Random Layout	- size/format definable	- really simple	-
Circle Layout	- fast - clear arrangement	- changing positions on update - lost space within the circle	radius
Ordered Circle Layout	- easy finding of certain nodes according to sort criteria - identification of connection patterns according to order - static positions	- lost space within the circle	radius, order of vertices
I-Self-Organizing Map Layout	- based on Meyer's self-organizing graph methods - easily adaptable to arbitrary types of visualization spaces (not implemented) - efficient	- nodes are likely to stick together	radius, initialAdaption, maxEpoch, coolingFactor
Fruchterman-Reingold Layout	-simple -fast	-blocking effects -isolated vertices drift away	attrac_multipl, repul_multipl, dimension, maxIter
Fruchterman-Reingold Layout grid-variant	- faster - holds isolated vertice/structures	- unnecessary edge crossings - symmetry of a large graph can be marred	attrac_multipl, repul_multipl, dimension, maxIter
Kamada-Kawai Layout	- seeks to achieve the geodesics as distances between pairs of nodes	- computational demanding	maxIter, length_factor, discon_multipl
Clan-based Graph Decomposition	- identifies subgraphs - supports contraction/expansion of subgraphs - cycles may be handled - dealing with unpleasant edges	- only for (tree-like) directed graphs	-

Table (4.1): Network visualization layouts in MORe

4.1.5.3. Social network analysis methods

MORe provides number of network measures, those measures are catalogued into the main categories centrality, prestige and authority.

MORe usually provides three kinds of each measure: the natural value normalized and standardized data as described in chapter 2. Table (4.2) lists the available network measures in MORe.



Key	*Description*
BwCn	Betweeness based centrality (normalized)
BwCstd	Betweeness based centrality (standardized)
BwCnn	Betweeness based centrality (not normalized)
BaC	Bary centrality (normalized)
BaCnn	Bary centrality (not normalized)
BaCstd	Bary centrality (standardized)
CLC	Closeness based centrality (normalized))
CLCnn	Closeness based centrality (not normalized)
CLCstd	Closeness based centrality (standardized)
EccCn	Eccentricity centrality (normalized)
EccCnn	Eccentricity centrality (not normalized)
EccCstd	Eccentricity centrality (standardized)
HITSAnn	HITS Authority (not normalized)
HITSHnn	HITS Hubs (not normalized)
IDCn	Indegree based centrality (normalized)
IDCnn	Indegree based centrality (not normalized)
IDCstd	Indegree based centrality (standardized)
ODCn	Outdegree based centrality (normalized)
ODCnn	Outdegree based centrality (not normalized)
ODCstd	Outdegree based centrality (standardized)
PInn	Outdegree-Prestige, not normalized
PPnn	Proximity-Prestige, not normalized
PRnn	Page Rank (not normalized)

Table (4.2): Synopsis of all measures in MORe

4.2. Summary of evaluated libraries

Finally, in this section a summarization of the different libraries that have been introduced in this chapter.

The evaluation will depend on the same points have been discussed earlier in this chapter, data entry and data manipulation, visualization techniques and social network analysis features. Table (4.3) recaps the five libraries have been introduced.

Library	data entry and data manipulation	visualization techniques	SNA features	Library feature
UCINET	Handle small network sizes	does not contain graphical procedures to visualize networks	large number of network analytic routines	module
PAJEK	Use data structure to facilitate the reduction of a large network into several smaller networks that can be treated further And it can handle multiple networks simultaneously	Advanced visualization approach. Many options to manipulate the graphs graphical representations of partitions, vectors, and combinations of partitions and vectors can be obtained.	Computation of degrees, depths, cores, or cliques centrality detection of components paths, or flows	Stand alone package
VISONE	single, unified network model (a labeled digraph)	Several different algorithms for drawing graphs	local measures distance measures feedback measures	Stand alone package
JUNG	Text files or data base files	Variety mechanisms for laying out and rendering graphs	Several different categories of network algorithms, such as ranking, Clustering and Blockmodelling, Topology, Paths, and Flows	module
MORe	array list which would include data about each node adjacency matrix which would include all possible edges in the network	Number of network drawing techniques are available (table 4.1)	Centrality, prestige and authority.	module

Table (4.3): summary of evaluation of different libraries

In this chapter, different techniques and libraries have been discussed, because the current disease model was developed using Repast Simphony, so that the appropriate library to use for the analysis of the social relationships network is MORe. It can be used with adjustment in the data structure used to represent network elements, by using the techniques and approaches used in other modules and software such as PAJEK. Next chapter introduce another approach to represent the data structure using the sparse matrix.

5. Implementation plan

In Chapter (3), the conceptual framework of the work was discussed. After the introduction and evaluation of different libraries for the analysis and visualization of social interaction networks in chapter (4), this chapter put the implementation plan in order to realize the conceptual framework in order to incorporate the social interaction network library with the pertussis model.

The implementation consists of a number of steps; the following steps are:

1. Enabling the social interaction network,
2. Capturing the network elements,
3. Initialize the network analysis library,
4. Network measure calculation,
5. Network display configuration,
6. Behavioral change.

5.1. Enabling the social interaction network

The Context is the core concept and object in Repast Simphony. It provides a data structure to organize agents from both a modeling perspective as well as a software perspective. Fundamentally, a context is just a bucket full of agents, but they provide more richness (Howe, Collier, North, Parker, & Vos, 2006).

Contexts provide model designers with a container that can maintain a localized state for agents. A Context state can maintain multiple interaction spaces called Projections, as well as more typical state information. Projections are designed such that they can be used to represent a wide range of abstract spaces, from graphs to grids to realistic geographic spaces. Importantly, projections and agents or individuals are independent of one another. Agents can be independent from the type of projection in which they are interacting, as well as projections can be independent from the type of agents whose relationships they maintain. Finally, the context provides a logical location to maintain agent behaviors that are dependent on localized agent interactions and environment. Figure (5.1) shows how the contexts are structures in Repast.

Many models use abstract relationships between agents using Networks or Graphs. Networks may represent social relationships, physical infrastructure connections or some other abstract connections. Repast provides tools to work with networks or graphs easily as a projection.

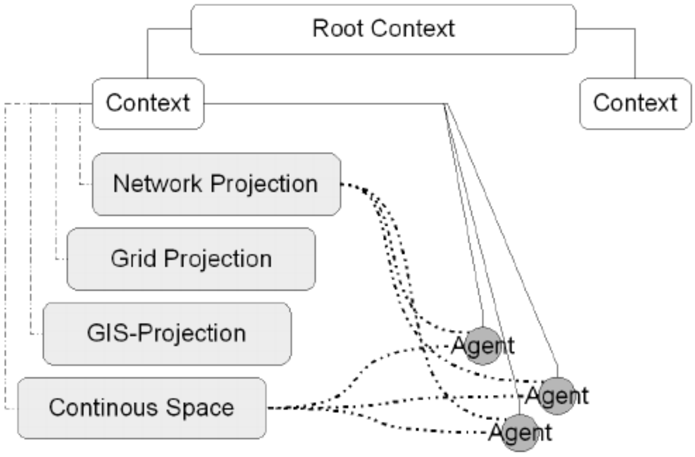


Figure (5.1) context and projections structure in repast

- **Building a network**

The first thing to remember when working with networks is that, like most projections, in order to participate in a network an agent must already exist in the context. A side effect of this is that any agent in a context is automatically a member of any networks associated with that context. The agent might not have any edges, but they will be in the network for all intents and purposes.

Network context is created using *ContextCreator* class which is responsible of creating all the contexts and projections in the pertussis model.

private static Context<Object> mainContext;

First, it creates an instant from the general object *Context*. Then creates a new class called *NetworkContext* which will hold all operations will be executed in the network elements.

NetworkContext networkcontext = new NetworkContext();

networkcontext.CreateNetwork();

The network elements will represent the individuals of the pertussis model, who are responsible for the flow of information over the network, for that reason the *network context* should be created as a sub context from the context *Individual context* with a separate projection called *Network Projection*. Figure (5.2) shows the structure of the model context after adding the network context and its projection.

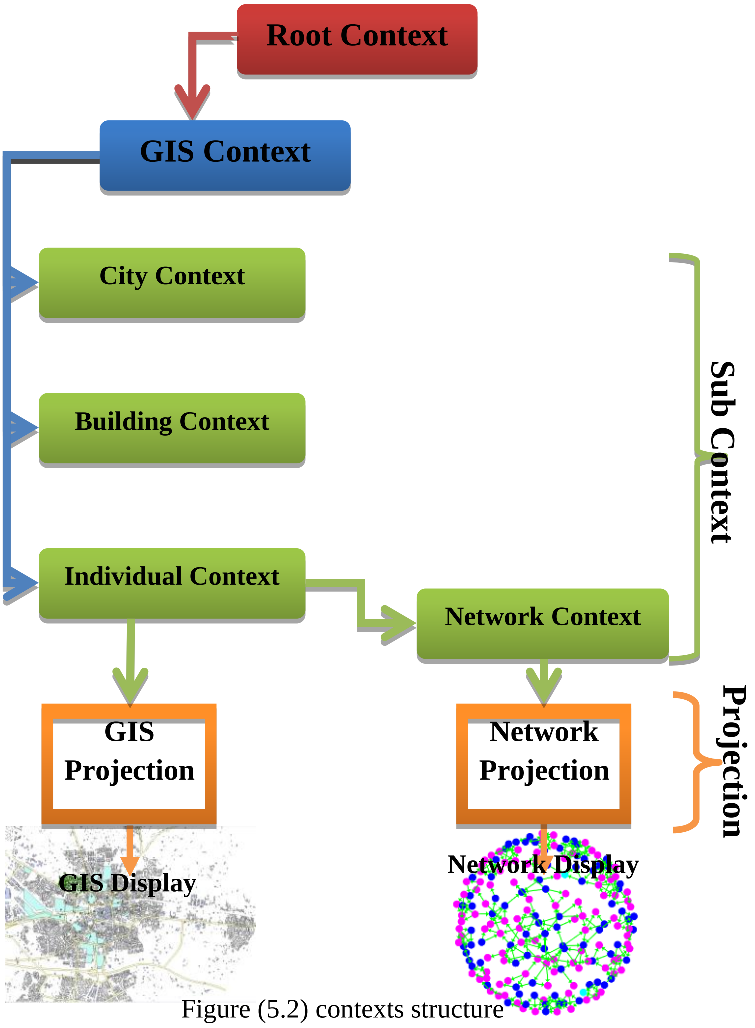


Figure (5.2) contexts structure

The method *CreateNetwork* will use the *NetworkBuilder*. The *NetworkBuilder* class allows the user to tune the network creation to create specific kind of network. This "tuning" is done by calling various methods on the builder.

```
private NetworkBuilder<Individual> socialNetwork = new NetworkBuilder<Individual>("SocialNetwork", this, true);

socialNetwork.buildNetwork();
```

The first argument to the builder constructor is the name of the network; the second is the context which the network "projects" and the last are whether or not the network is directed.

Final step in the enabling of the network context is to assign an appropriate projection, in order to give the elements the space to interact.

```
public static Network<Individual> getSocialNetowrk(){

return (Network<Individual>) ContextCreator.getNetworkContext().getProjection("SocialNetwork");

        }
```

Once the network context is built, other implementation steps would be able to be implemented in the created context.

Once the network is built, relationships can always be added, but that's not really useful until queries can be based on the relationships, for example to visualize the nodes that have certain criteria such as the degree centrality or the closeness centrality. The Network library provides all of the methods you would expect to have to access the objects and relationships in it. The particular methods which would be interested are:

- *getAdjacent(agent)* - Retrieves all of the objects that share a relationship with this agent.
- *getPredecessors(agent)* - Retrieves all of the objects that have a relationship to this agent.
- *getSuccessors(agent)* - Retrieves all of the objects that have a relationship from this agent.
- *getEdges(agent)* - Retrieves all of the edges which have this agent as an endpoint.
- *getOutEdges(agent)* - Retrieves all of the edges which have this agent as a source.
- *getInEdges(agent)* - Retrieves all of the edges which have this agent as a target.
- *addEdge(agent, agent)* - Adds an edge between the two agents.
- *addEdge(agent, agent, weight)* - Adds an edge of the specified weight between the two agents.
- *addEdge(RepastEdge edge)* - Adds the specified edge. Assumes that the source and target of the edge are part of the network.
- *getEdge(agent, agent)* - Get the edge, if any, between source and target agents.

5.2. Capturing the network elements

As mentioned in chapter (3), the network consists of two main elements: the nodes which will represent the agents of the pertussis model, and the links or relationships between nodes which reflect the interaction between different agents during the simulation, this interaction would be physically by spreading the disease from person to other, or un-physically by passing information about the disease by different medias.

In order to capture the nodes of the social network elements from the disease model, the network context represents the agents as nodes stored in an array list which would include data about each node such as nominal data, health status, vaccination level, age group, current activity.

In order to capture the edges to represent the interaction between agents, the edges between pairs of nodes are represented by an adjacency matrix, in which nodes are listed both in the columns and rows. If there is a relation between two nodes, the value “1” is assigned to the field that represents the row node and the column node, and “0” indicates that there is no edge between the particular nodes. For the case of undirected networks, the matrix is symmetric.

In order to represent the large number of nodes in an efficient data structure that is appropriate to characterize 60000 agents, different scenarios could be used to convert the adjacency matrix. One of these scenarios is the use of the data structure suggested by PAJEK. In PAJEK, the main goal in the design of its data structure is to support abstraction by decomposition of large network into several smaller networks that can be treated further, which makes the analysis and visualization more efficient and powerful. The idea of decomposition is to find clusters (cut out of components, neighborhoods of important nodes, etc…) in the network, and shows relations among clusters in a hierarchical manner as shows in figure (5.3).

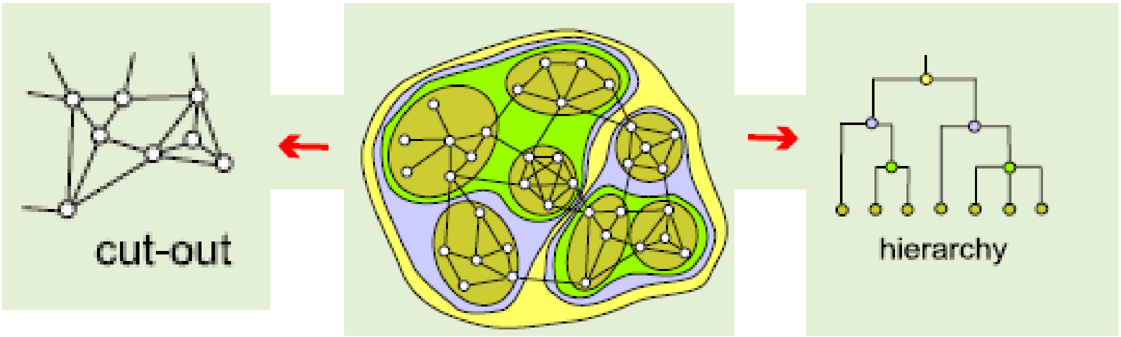
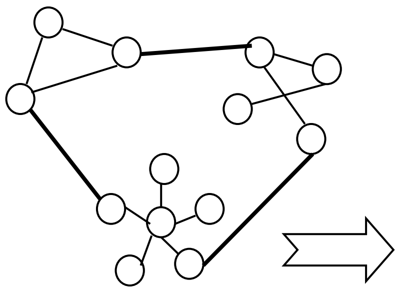


Figure (5.3) Decomposition in PAJEK

Figure (5.4.a) shows a small part of a large network with number of nodes, so that the adjacency matrix which represents the relationships between the nodes is shown in figure (5.4.b).

Figure (5.5.a) shows the decomposition of the network in figure (5.4.a) with one level of clustering into three clusters illustrated in the figure. Decomposition of the network will also split the adjacency matrix into three matrices each one reflects the corresponding clusters, figure (5.5.b).

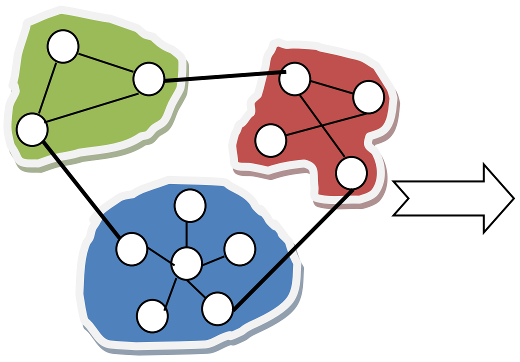


	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	1	0	0	0	0	0	0	0	0	0	0
2	1	0	1	1	0	0	0	0	0	0	0	0	0
3	1	1	0	0	0	0	0	1	0	0	0	0	0
4	0	1	0	0	1	1	0	0	0	0	0	0	0
5	0	0	0	1	0	0	1	0	0	0	0	0	0
6	0	0	0	1	0	0	0	0	0	0	1	0	0
7	0	0	0	0	1	0	0	0	0	0	0	0	0
8	0	0	1	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0	0	0	1
11	0	0	0	0	0	1	0	0	0	0	0	0	1
12	0	0	0	0	0	0	0	0	0	0	0	0	1
13	0	0	0	0	0	0	0	1	1	1	1	1	0

(a)

Figure (5.4) snippet of large network

(b)



(a)

	1	2	3
1	0	1	1
2	1	0	1
3	1	1	0

	1	2	3	4
1	0	1	1	0
2	1	0	0	1
3	1	0	0	0
4	0	1	0	0

	1	2	3	4	5	6
1	0	0	0	0	0	1
2	0	0	0	0	0	1
3	0	0	0	0	0	1
4	0	0	0	0	0	1
5	0	0	0	0	0	1
6	1	1	1	1	1	0

(b)

Figure (5.5) Clustering

According to the example above and the concept of PAJEK data structure in representing large networks, the large number of agents in the disease model, could be represented in the same way by the use of number of clusters by represent the adjacency neighborhoods which their households send their children to the same school in one cluster as illustrated in figure (5.6).

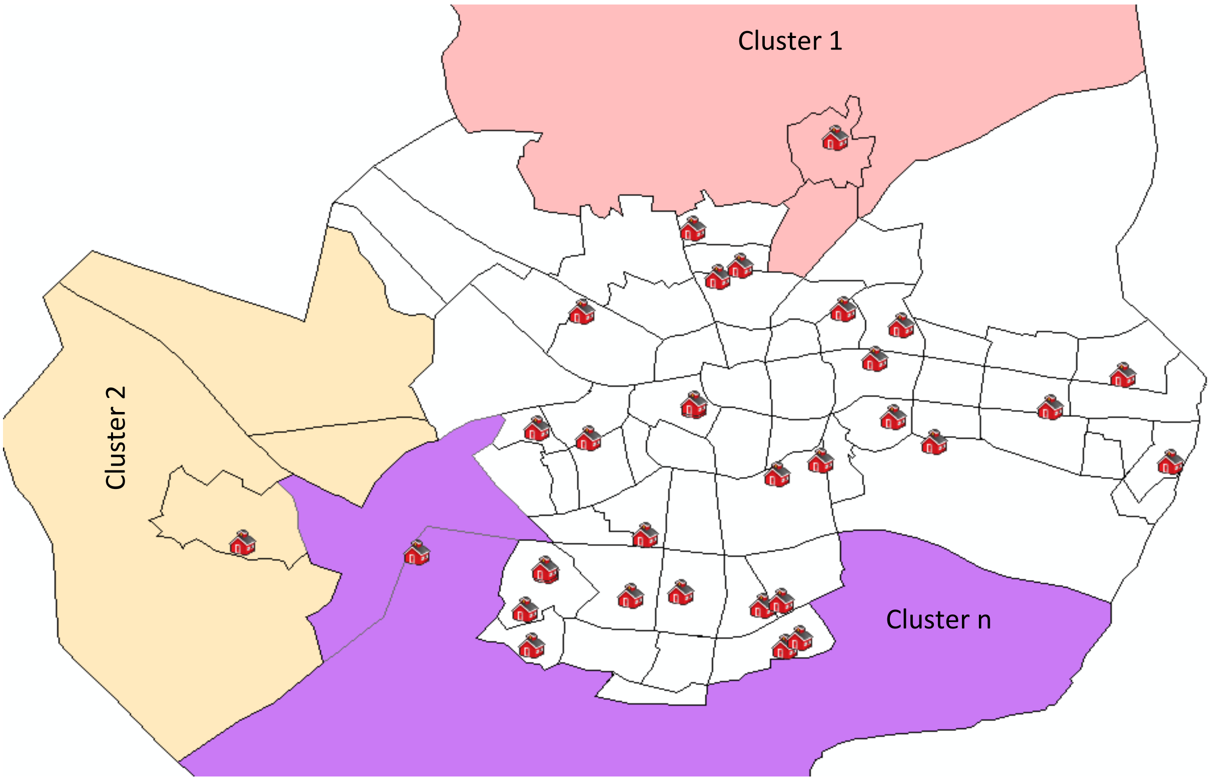
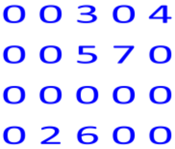


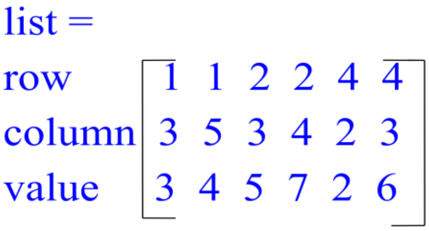
Figure (5.6) clustering of agents in the disease model

Another scenario to solve the problem of large network representation in Repast is the use of sparse matrices. The native data structure for a matrix is a two-dimensional array. Each entry in the array represents an element $a_{i,j}$ of the matrix and can be accessed by the two indices *i* and *j*. For an $m \times n$ matrix, enough memory to store at least $(m \times n)$ entries to represent the matrix is needed. Substantial memory requirement reductions can be realized by storing only the non-zero entries. Depending on the number and distribution of the non-zero entries, different data structures can be used and yield huge savings in memory when compared to a native approach. One of those data structures to use “single linear list in row major order” which is stand for scan the nonzero elements of the sparse matrix in row-major order. each nonzero element is represented by a triple (row, column, value), in which the list of triples may be an array list or a linked list (chain). For example, suppose the following matrix:



$$\begin{matrix} 0 & 0 & 3 & 0 & 4 \\ 0 & 0 & 5 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 6 & 0 & 0 \end{matrix}$$

By scanning the non zero elements, the new data structure will be as the following matrix, so that the single linear list has 6 elements each list element is a triple (row, column, value):



$$\text{list} =$$
$$\begin{array}{l} \text{row} \\ \text{column} \\ \text{value} \end{array} \begin{bmatrix} 1 & 1 & 2 & 2 & 4 & 4 \\ 3 & 5 & 3 & 4 & 2 & 3 \\ 3 & 4 & 5 & 7 & 2 & 6 \end{bmatrix}$$

Returning to the adjacency matrix of the example in figure (5.4), by emplimenting the sparse matrix data structure, we get the matrix in figure (5.7), in which the single linear list has 28 elements each list element is a triple (row, column, value), instead of 26x26 matrix.

row	1	1	2	2	2	3	3	3	4	4	4	5	...	11	12	13	13	13	13	13
column	2	3	1	3	4	1	2	8	2	5	6	4	...	13	13	8	9	10	11	12
value	1	1	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1

Figure (5.7) use of sparse matrix

For the case of the disease model when the adjacency matrix is *60000 x 60000* agents, and because during the simulation there are only few number of relationships between agents which means few nonzero elements in the corresponfing matrix, so that it is obvious that the use of sparse matrix to represent the interaction between agents would give up huge savings in memory when compared to the adjacency matrix of Repast and even to the data structure provided by PAJEK.

When the agents are created in the disease model before starting the simulation, they are created according to the following assumptions:

- Families with children, the maximum number of children are four.
- Single fathers with children, the maximum number of children are two.
- Single mothers with children, the maximum number of children are three.

The matrix which contains the relations between agents will be established in such a way that the number of rows is equal to the total number of agents as well as for the number of columns and with value of zero for all cells at the beginning.

The disease model starts by creating the household head first and assign to her/him her/his partner and children to create the household family, with consideration of the maximum size of each type of families. After creation of the families, the adjacency matrix will be filled with the new relations between the created family members. For example the first created household head will get the id = 1, his partner will get id = 2, and they have 4 children, so that all members has full interactions with each other, and the matrix will look like the following, figure (5.8).

	1	2	3	4	5	6	…
1	1	1	1	1	1	1	0
2	1	1	1	1	1	1	0
3	1	1	1	1	1	1	0
4	1	1	1	1	1	1	0
5	1	1	1	1	1	1	0
6	1	1	1	1	1	1	0
…	0	0	0	0	0	0	0

Figure (5.8) establishment of adjacency matrix in the beginning of the disease model simulation

With the following assumptions of the values which represent the type of relations between agents (edges between nodes in the network):

0: no interactions

1: unbreakable family members interactions (one directinal relations between parents and children – bidirectional relations between children).

2: breakable relations that are considered to be permanent in simulation (teacher-her students' relations in a class - children in a school class, people within a department at work or adults in high school).

3: non-permanent relations (people that meet each other by accident).

After creating the household members, activity patterns will be assigned to the agents in order to start the simulation, and to let each agent knows its next place to go, for example children to schools, parents to workplaces or marketplaces, adults to friends…

According to the type of relation, the corresponding nodes in the adjacency matrix will get the corresponding value in the cell. For example in figure (5.9), suppose the following interactions in the beginning of the simulation with assigned activities to certain agents:

	1	50	547	1000	1050	20000	…
3	1	0	0	0	0	3	…
55	2	1	0	0	0	0	…
550	0	0	2	0	0	0	…
990	0	0	0	3	0	0	…
1051	0	0	0	0	3	0	…
20010	3	0	0	0	0	2	…
…	…	…	…	…	…	…	…

Figure (5.9) assigning activities to the agents and node in the adjacency matrix

During the simulation, when there are new interactions between random agents or an existing relation is detached, that should be reflected in the adjacency matrix accordingly. From the previous example, the relation between the two agents 20000 and 3 is non-permanent and will be detached after a while during the simulation. In the same time, in a new workplace, agent 1000 could meet agent 55 so that an interaction would be established between them.

In the same time the adjacency matrix is established, the appropriate arry linear list (sparse matrix) is established in oreder to scan the nonzero cells in the adjacency matrix and fill them in the array. Returning to the example in figure (5.9), the array list created in the beginning of the simulation will be as the following:

row	3	3	55	55	550	990	1051	20010	20010
column	1	20000	1	50	547	1000	1050	1	20000
value	1	3	2	1	2	3	3	3	2

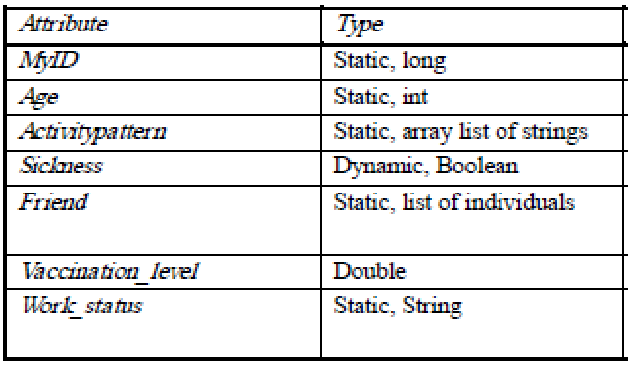
Figure (5.10) the corresponding sparse matrix of the beginning of simulation

During the simulation, when the edges between nodes are updates by adding new edges or detaching existing edges, the sparse matrix will be updated accordingly. for example, the relation between the two agents 20000 and 3 in figure (5,10) is non-permanent and will be detached after a while during the simulation. In the same time, in a new workplace, agent 1000 could meet agent 55 so that an interaction would be established between them, the updated sparse matrix during the same period of simulation is illustrated in fugure (5.11).

row	3	55	55	55	550	990	1051	20010	20010
column	1	1	50	1000	547	1000	1050	1	20000
value	1	2	1	2	2	3	3	3	2

Figure (5.11) updated sparse matrix during the simulation

In the pertussis model, each agent is represented with number of attributes such as the agent ID, age, family ID, and other attributes as illustrated in table (6.1).



Attribute	*Type*
MyID	Static, long
Age	Static, int
Activitypattern	Static, array list of strings
Sickness	Dynamic, Boolean
Friend	Static, list of individuals
Vaccination_level	Double
Work_status	Static, String

Table (5.1) agent basic attributes

The network elements were stored in the "single linear list" which is represented by a triple (row, column, value), in which the list of triples may be an array list.

The row will represent the first agent as a node, the column will represent the second agent, and the value will represent the relationship type between the two agents. In the row and column, the *AgentID* will be stored to present the agents, because from the *AgentID*, it easy to have access to all other agents' attributes.

private ArrayList<sparse> capture = new ArrayList<sparse>();

where *sparse* is a class to present the triple (row, column, value).

The disease model start by creating the household head first and assign to her/him her/his partner and create children and assign them randomly to the household, with consideration of the maximum size of each type of families. Each pair of agents created, it is added to the arraylist simultaneously. For example, when the first househould head is created with his partner, it is added to the sparse matrix:

this.capture.add(new sparse((int)p.getMyID(), (int)p2.getMyID(), 1));

with: *p* the first agent, *p2* the partner, *1* is the type of relationship which means family relation.

When the children is created and assigned randomly to families, the model check the type of the family, whether it is family, single father or single mother.

this.capture.add(new sparse((int)randomFamilyMember.getMyID(), (int)child.getMyID(), 1));

if (randomFamilyMember.getPartner() != null)

this.capture.add(new sparse((int)randomFamilyMember.getPartner().getMyID(), (int)child.getMyID(), 1));

with *randomFamilyMember* is a variable look randomly for a household head, *child* is the created agent and *getPartner* is looking for the partner of the household head.

After creating the children, the model add also the relationship between the siblings. And after that, each child and adult look for friends to be assigned and the activity patterns which each agent will start the simulation with.

p.addFriends(friend);

friend.addFriends(p);

this.capture.add(new sparse((int)p.getMyID(), (int)friend.getMyID(), 2));

During the simulation, when the edges between nodes are updated by adding new edges or detaching existing edges, the sparse matrix will be updated accordingly.

5.3. Initialize the network analysis library

Import the software library that contains all java interfaces and classes that help to analyze and visualize the social network model during the simulation and apply certain measures for the analysis of the network components in order to implement the most appropriate measures that are relevant to the case of disease diffusion.

- The two libraries need to be used in the research are MoRe (Managing Ongoing Relationships) and ReSoNetA. As mentioned in chapter (2), the purpose of MoRe is to support representing and analyzing social networks in agent-based modeling (Holzhauer, 2009). The usage of MoRe consists of:

  - Installation of the library by making the java code available to the existing model code, initializing moRe.
  - Configuration of network measures and adding Measure Definitions.
  - Managing node measures.
- Setting Parameters for Node Measure Calculation.
- However, moRe currently doesn’t contain many network measure calculations. But it is straight forwards to convert the java code from ReSoNetA into MoRe, by adding the missing measures packages to MoRe.
- In the other hand, ReSoNetA is a software library that adds network analysis and visualization features for Repast J models. It comes as an ordinary jar-file.
- In order to provide the library with an entry point into the existing model some minor changes are necessary. The idea is to interchange some classes of the model with ReSoNetA-versions of these classes that add or alter some functionality.
  - Change the agent super class.
  - Adapt the model class.
  - Initialize the node table accordingly in order to apply the measures manager which contains all measures as provided by any network measure supplier. The node table lists the nodes of the displayed networks with their measure values.

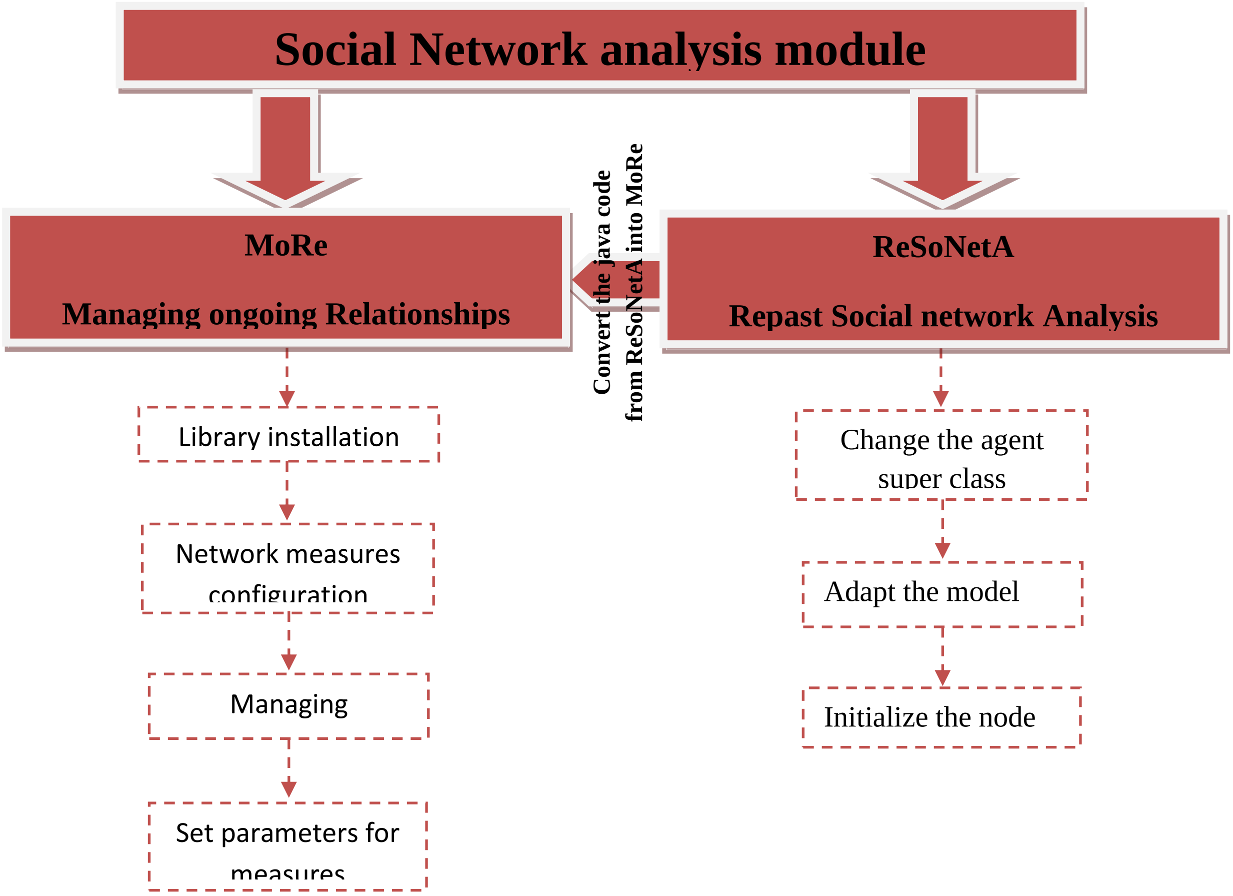

Social Network analysis module
MoRe
Managing ongoing Relationships
ReSoNetA
Repast Social network Analysis
Convert the java code from ReSoNetA into MoRe
Library installation
Network measures configuration
Managing
Set parameters for measures
Change the agent super class
Adapt the model
Initialize the node

Figure (5.12): adding the analysis library to apply relevant measures

The network analysis library which is used to analyze the social interaction between the disease model agents, is called MoRe, The purpose of MoRe- Managing ongoing Relationships is to support representing and analyzing social networks in agent-based modeling. Some of the features and code is based on ReSoNetA, but more directed to Repast Simphony (RS).

the main MoRe features are:

- ✓ Network generators.
- ✓ Network Manipulation.
  - Manipulation Agent Components.
    - Manages the agents' links according to differences between partners regarding some node property.
- ✓ Calculation and management of network and node measures.
- ✓ Input/Ouput.
  - GraphML import / export (including node/network attributes).

Considering the following steps to use MoRe properly:

- ✓ Installation.
  - Make the java code available to the project code.
- ✓ Initializing MoRe.
  - Call MManager.getInstance().init() to initialize MoRe.

MORe is conists of number of packages and classes to deal with network analysis features. The Figure (5.13) illustrates the main packages of MORe.

1. Basic: contains the main classes that are responsible for the initialization of MORe functionality to deal with networks and edges.
2. Param: contains Basic parameter definitions for MORe and main parameter definitions used for MORe's network building.
3. Building: Initial creation of network and edges.
4. Manipulate: manipulation of networks, agent links and edges. It is contain class responsible for Agent component that manages the agents links according to differences between partners regarding some node property, and provides methods to add agents to and remove agents from an existing network.
5. Measures: the main package that contains the network measures. It described in details in section5.4.

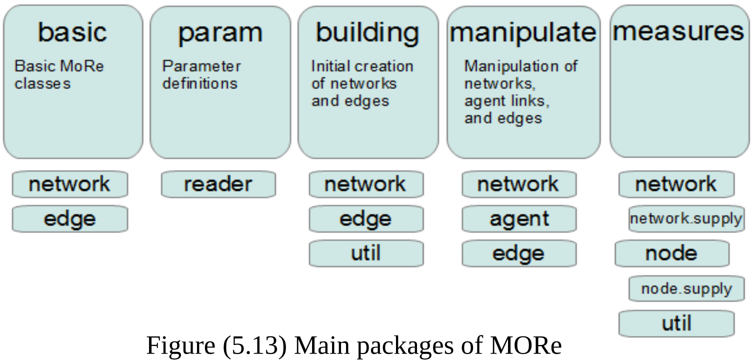


Figure (5.13) Main packages of MORe

Initialize MORe by calling the class *MManager* from the package *de.cesr.more.basic*.

MManager.init();

MManager.setSchedule(new MRsSchedule(RunEnvironment.getInstance().getCurrentSchedule()));

The above class are called in order to initialize parameter framework and schedule network measure calculations.

Register network at MoRe: *MNetworkManager.setNetwork(network, "SocialNetwork");*

Where the *network* is implements the interface *MoreNetwork* from the package *de.cesr.more.basic.network*, which Specifies demands on network implementations such as agent type the common type (of agents) that is contained as nodes in the network and edge type. *"SocialNetwork"* the name of the network.

5.4. Network Measure Calculation

MORe provides some code that makes is quite easy to calculate a number of network measures for networks. Only a few steps are required to anchor the network measure support to your model. All code needs to be added to your models setup() method:

1. Tell the NetworkMeasureUtilities the current schedule.
2. Add the measures you want to the calculation. The first argument is a JungNetwork that may be fetched from the NetworkProjektion.*getNet().getJungNetwork()*. The key for the desired measure follows before a HashMap is given that contains optional parameters.
3. Tell the display descriptor to output the table.
4. Add the networks the table shall comprise.
5. Add one or more measures to be displayed in the table.

Table (5.3) contains all currently supported measures in MORe.

Key	Description	Return Type
BWCn	Betweeness based centrality (normalized)	Double
BWCstd	Betweeness based centrality (standardized)	Double
BwCun	Betweeness based centrality (unnormalized)	Double
CLC	Closeness based centrality (normalized))	Double
CLCstd	Closeness based centrality (standardized)	Double
CLCun	Closeness based centrality (not normalized)	Double
EccCn	Eccentricity centrality (normalized)	Double
EccCstd	Eccentricity centrality (standardized)	Double
EccCun	Eccentricity centrality (not normalized)	Double
IDCn	Indegree based centrality (normalized)	Double
IDCstd	Indegree based centrality (standardized)	Double
IDCun	Indegree based centrality (not normalized)	Double
ODCn	Outdegree based centrality (normalized)	Double
ODCstd	Outdegree based centrality (standardized)	Double
ODCun	Outdegree based centrality (not normalized)	Double
PIun	Indegree-Prestige, unnormalized	Double

Table (5.3) Supported network measures

The package *de.cesr.more.measures* is responsible for calculation of different measures. It contains *MoreMeasure* interface the basic measure class that holds the measure type, a map of parameters, the *MeasureDescription* (provide information about *MoreMeasure* objects) and the *MoreAction* responsible for measure calculation. Also this package contains *MoreMeasureSupplier* which is an interface for classes that provide Measures.

5.5. Network Display Configuration

It is convenient to give a hint on the layout for the particular displays since this is the issue that mostly differs among the initialized displays. Further settings may be specified and comprise the following areas:

- Updating the display:

Regarding the updating of network layouts the frequency and interval may be specified. For frequency there are four opportunities:

  - ON_NEW: The layout is updated every time an element is added or removed.
  - AT_UPDATE: Every time the display is updated also the layout is updated.
  - AT_INTERVAL: The process is executed according to the given interval. If the interval is 2, every second call on the display's update() method also leads to an update of the layout. However, the updater is also active when an element is moved, added or removed no matter if the interval is matched.
  - ON_MOVE: The layout is updated every time an element is moved.

- Layout:

The layout that is used to determine the coordinates of node is specified by its classname. The layout projection is the one that is passed to the layout class and used to determine the agent's position. Table (5.4) shows available layout classes and their advantages.

The user may also specifiy whether the nodes that are filtered out at the node table shall disappear in the network display. If so, all edges that belong to a filtered node are also set invisible.

Classname	Advantages
RepastS.CircleLayout2D	Simple; useful for nodes that are located along a line
RepastS-Fix.ISOMLayout2D_Fixed	Easily adaptable to arbitrary types of visualization; efficient
RepastS-Fix.FRLayout2D_Fixed	Fast
RepastS-Fix.KKLayout2D_Fixed	Seek to achieve the geodesics as distances between pairs of nodes

Table (5.4) Layouts that may be used in MORe

6. Results

In chapter (3), the conceptual framework of the work was presented. Chapter (4) presented a comprehensive introduction and evaluation of different libraries for the analysis and visualization of social interaction networks. Chapter (5) discussed the realization of the conceptual framework into the implementation process. In this chapter experiments are conducted on the model in order to get the results of each step in the implementation plan. The purpose of these experiments is to proof that the implementation is functioning correctly.

The experiments will be done to check the validity of the following implementation steps:

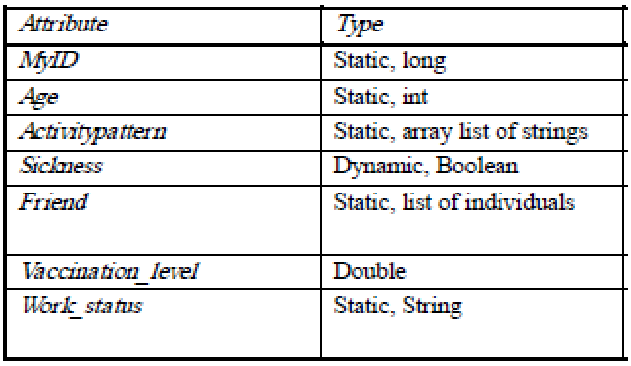
- Capturing the network elements,
- Network measure calculation,
- Network display configuration,

6.1. Capturing the network elements

The fact that the network capturing works is already confirmed by the fact that measures can be performed on it (section 6.2) and that it can be displayed (section 6.3). However, it is also necessary to confirm that all agents and all edges are captured and it should be tested if the implementation via the sparse matrix indeed has the ability to work with large numbers of agents.

According to the discussion in the implementation plan about the suitable solution to represent the network elements, it was decided to implement the sparse matrix in a “single linear list in row major order” to be able to present large networks.

In the pertussis model, each agent is represented with a number of attributes such as the agent ID, age, family ID, and other attributes as illustrated in table (6.1).



Attribute	*Type*
MyID	Static, long
Age	Static, int
Activitypattern	Static, array list of strings
Sickness	Dynamic, Boolean
Friend	Static, list of individuals
Vaccination_level	Double
Work_status	Static, String

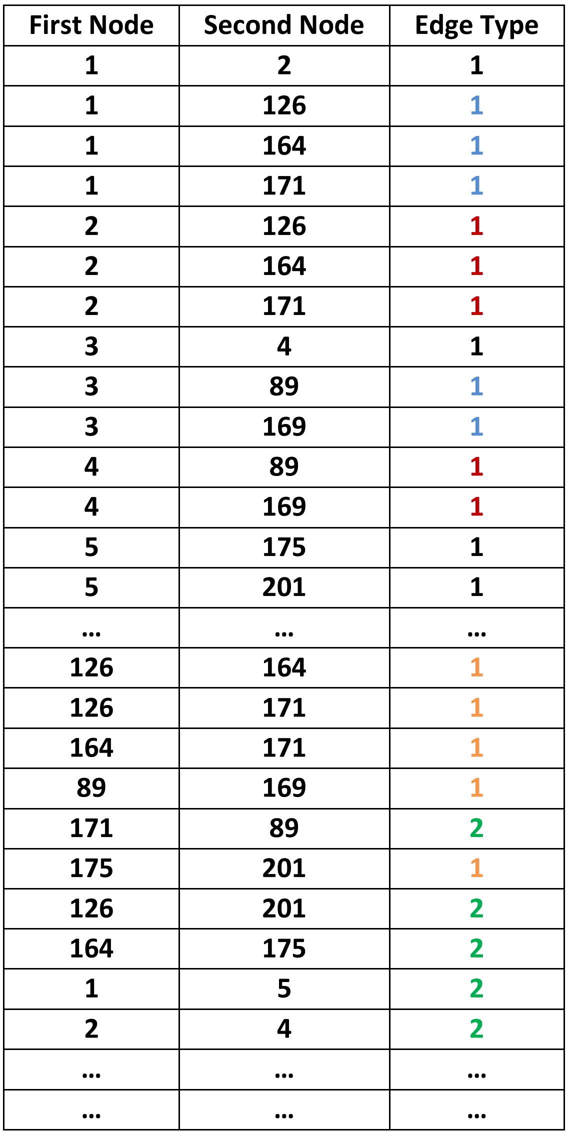
Table (6.1) agent basic attributes

As described in chapter (5), the elements in the “single linear list” are represented by a triple (row, column, value), in which the list of triples may be an array list.

The row will represent the first agent as a node, the column will represent the second agent, and the value will represent the relationship type between the two agents.

In order to proof that the agents and edges are captured correctly, experiments were conducted on a subset of the model consisting of two neighborhoods. Objective of this exerpeiment is to find out if indeed the network contains the correct number of agents, the correct number of male and female and the correct number of infants, children, teenagers and adults.

Two neighborhoods, ”City” and “Lasonder 't Zeggelt”, (total population=576) were selected for this experiment because they contains number of agents that are possible to be analyzed and displayed. After the simulation created the synthetic population and the sparse matrix is filled with all possible relationships between different created agents, the result was captured and analyzed. A snapshot of the created sparse matrix is shown in figure (6.1). The nodes with the relationships are created with the consideration of the family types and and number of children. Figure (6.2) how the created nodes and relationships (from the snapshot) would be linked in order to shape the network in the beginning of the simulation.



First Node	Second Node	Edge Type
1	2	1
1	126	1
1	164	1
1	171	1
2	126	1
2	164	1
2	171	1
3	4	1
3	89	1
3	169	1
4	89	1
4	169	1
5	175	1
5	201	1
…	…	…
126	164	1
126	171	1
164	171	1
89	169	1
171	89	2
175	201	1
126	201	2
164	175	2
1	5	2
2	4	2
…	…	…
…	…	…

Figure (6.1) created sparse matrix in the beginning of the simulation

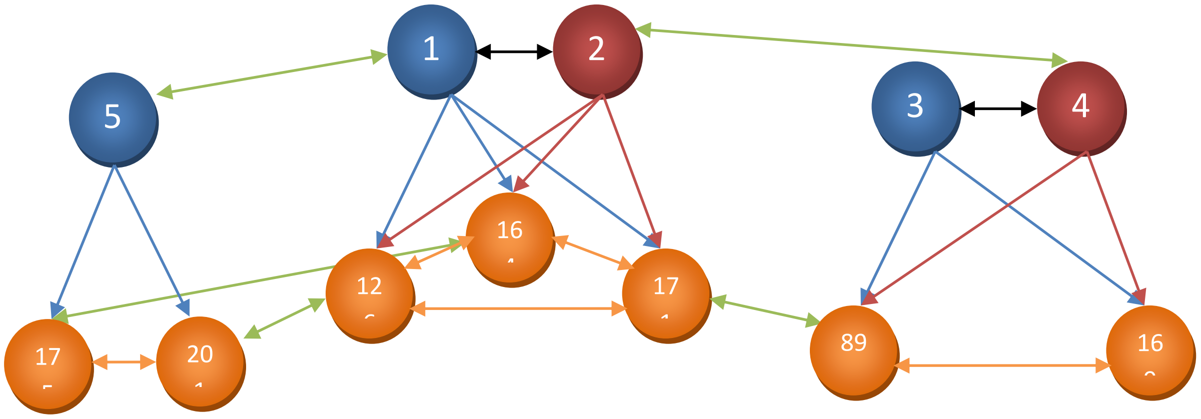


Figure (6.2) the created network after capturing its elements from the sparse matrix

With the consideration of the type of relations, where: "1" is representing partnership relations, "2" is representing parents - children relations, "3" is representing sibling relations and "4" is representing children friendship relations.

In order to verify that the agents and their relations are captured correctly, and converted to nodes and edges into the sparse matrix, the model performs transferring the sparse matrix into a CSV (Comma Separated Values) file. Manual check can be performed on that file to verify that the capturing is correct. First the numbers of agents: the sparse matrix contains relationships between the whole population found in the two neighborhoods.

In order to check if the model captures the different types of edges in the correct way, the following experiments were set up:

edge type	1	2	3	4
Count	83	481	172	116

- When only looking at the edge type 2, it is found that:

Count	total
1	110
2	88
3	65

This means that of agents/nodes that have a type 2 edge, 110 have only one connection, 88 have 2 and 65 have 3 connections.

- When only looking at the edge type 3, it is found that:

Count	total
1	98
2	37

This means that of the agents with a type 3 connection, 98 have only 1 connection, and 37 have 2 connections.

- When only looking at the edge type 4, it is found that:

Count	total
1	54
2	31

This means that of the agents with a type 4 connection, 54 have only 1 connection, and 31 have 2 connections.

By combining all edges' types together:

	edge 1	**edge 2**	**edge 3**	**edge 4**
1	83	110	98	54
2	-	88	37	31
3	-	65	-	-
total	83	481	172	116

From the table, there are 83 partners that mean there are at least 83 families of different sizes. From the statistics of edge type 2, it could be concluded that there are 110 parents including single father and single mother families have only one child, 88 parents have 2 children and 65 have 3 children. These results can also be concluded from the edge 3 which represents the siblings' relations. From edge type 3, only 98 families have 2 children, in other words 98 families of size three and four (with consideration of families with single fathers or mothers) and 37 families have 3 children or 37 families of size four and five. In total there are 172 families with size 3, 4, or 5. From the model, total number of families of three types created for the two neighborhoods is 179, so that it can be derived 7 families of size 2 which means single father or single mother families with only one child. These findings are met with the findings in the existing model of (Abdulkareem, 2010), when she discussed the difference between the families' sizes in the simulation and census data, as what is show in figure (6.3).

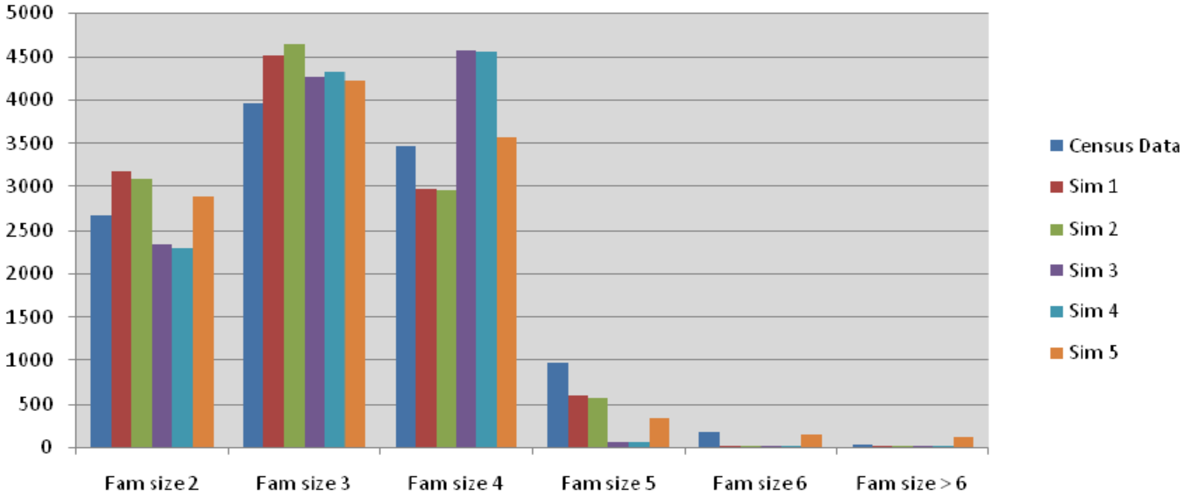


Figure (6.3) the difference between the families' sizes in the simulation and census data

According to the analysis of figures (6.1) and (6.2), it can be drawn the following:

- Agents can have "0" or "1" partner. "0" in the case of agent ID 5 which is a single father for two children and "1" in the case of agent ID 1 and 5.

- Agents can have 1, 2, or 3 children, but according to the model assumptions discussed in chapter (5) families can have 4 children in maximum, that also indicated in figure (6.3) the family size 6 is near or equal to zero.
- The number of children of an agent should be equal to the number of children of its partner. For example agent ID 1 has three children and his partner agent ID 2 has the same children, as well as for the agent ID 3 and his partner agent ID 4.
- All siblings of a family should have an equal number of family edges.

From table (6.2) node with *AgentID* = “1” has a family relation with node “2” where node “2” is the node “1”’s partner. Nodes “1” and “2” have three children, nodes “126”, “164” and “171”, and these nodes have sibling relations between all of them. In the generated sparse matrix, families of type single father or single mothers are also represented such as the node “5” and it is children “175” and “201”. The friendships relations between children as well as between adults are represented such as the relation between nodes “126” , “201” and nodes “2”, “4”. In the generated sparse matrix there are no nodes without any relations. By implementing one of the network measures on the captured elements, for example the degree centrality, each node will get a certain value according to the number of connections each node has.

By repeating the experiment with different neighborhoods, the model start with choosing the most populated 5 neighborhoods which are: “Wesselerbrink NW”, “Stroinkslanden Zuid”, “Helmerhoek-Zuid”, “Helmerhoek-Noord” and “Stevenfenne” with total number of all individuals Created: 12233 individuals. We get different results as illustrated in the snapshot in figure (6.4).

First Node	Second Node	Edge Type	First Node	Second Node	Edge Type
1	2	1	5	2673	2
1	1557	2	6	1564	2
1	2275	2	6	2581	2
2	1557	2	6	2673	2
2	2275	2	7	8	1
3	4	1	7	2046	2
3	1816	2	7	2464	2
4	1816	2	7	2738	2
5	6	1	8	2046	2
5	1564	2	8	2464	2
5	2581	2	8	2738	2

Figure (6.4) created sparse matrix in the beginning of the simulation for 5 neighborhoods

In order to check if the model captures the different types of edges in the correct way, the following experiments were set up:

edge type	1	2	3	4
Count	2412	9769	2123	3206

- When only looking at the edge type 2, it is found that:

Count	total
1	3178
2	2886
3	273

This means that of agents/nodes that have a type 2 edge, 6336 have only one connection, 1307 have 2 and 273 have 3 connections.

- When only looking at the edge type 3, it is found that:

Count	total
1	1819
2	152

This means that of the agents with a type 3 connection, 1971 have only 1 connection, and 81 have 2 connections.

- When only looking at the edge type 3, it is found that:

Count	total
1	1624
2	791

This means that of the agents with a type 3 connection, 2416 have only 1 connection, and 395 have 2 connections.

By combining all edges' types together:

	edge 1	**edge 2**	**edge 3**	**edge 4**
1	2412	3178	1819	1624
2	-	2886	152	791
3	-	273	-	-
total	2412	9769	2123	3206

From the table, there are 2412 partners that mean there are at least 2412 families of different sizes. From the statistics of edge type 2, it could be concluded that there are 3178 parents including single father and single mother families have only one child, 2886 parents have 2 children and 273 have 3 children. These results can also be concluded from the edge 3 which represents the siblings' relations. From edge type 3, 1819 families have 2 children, in other words 1819 families of size three and four (with consideration of families with single fathers or mothers) and 152 families have 3 children or 152 families of size four and five. In total there are 2123 families with size 3, 4, or 5. From the model, total number of families of three types created for the five neighborhoods is 3925. These results also met with the findings in the model in which the family size 2,3 or 4 are more than the family size 5.

With different experiments, it is noticed that the extended model is capturing the nodes and the relationships correctly and consistency. It is noticed also that the number of relationships each agent has is not more than 4, and with the observation of the way the model is run and created the population, it is seen that the model has a problem when it tries to assign friendship relationship to adults and teenagers.

6.2. Network Measure Calculation

The MORe library contains a number of features in order to analyze the network elements, and as discussed in chapter (3) , the most important measures are probably: the number of links an agent has in order to pass the message (degree of centrality) and closeness centrality (maximum number of links people are apart in the network).

The available source code provided with the MORe library contains the interfaces and classes that are responsible for the initialization of the nodes measure manager and centrality measure supplier. These classes contains the needed methods in order to check and calculate the network measures. For the time being, MORe library only contains the method for calculation the degree centralty.

Manually, the degree centrality of all created agents stored in the generated sparse matrix can be calculated.

By implementing the degree centrality measure on the generated population for the two neiborhoods ”City” and “Lasonder 't Zeggelt”, we get the following results:

- The Total number of all individuals Created: 576.
- The pair (Agent ID, Degree)
- for the whole created agents are shown in figure (6.5) which shows a chart of the calculated degree centrality for number of nodes from node with *AgentID* = 1 to node “82”. As seen from the chart, all nodes have degree centrality with values range from 1-4 according to the consideration of number of children per family and number of friendship per individual.

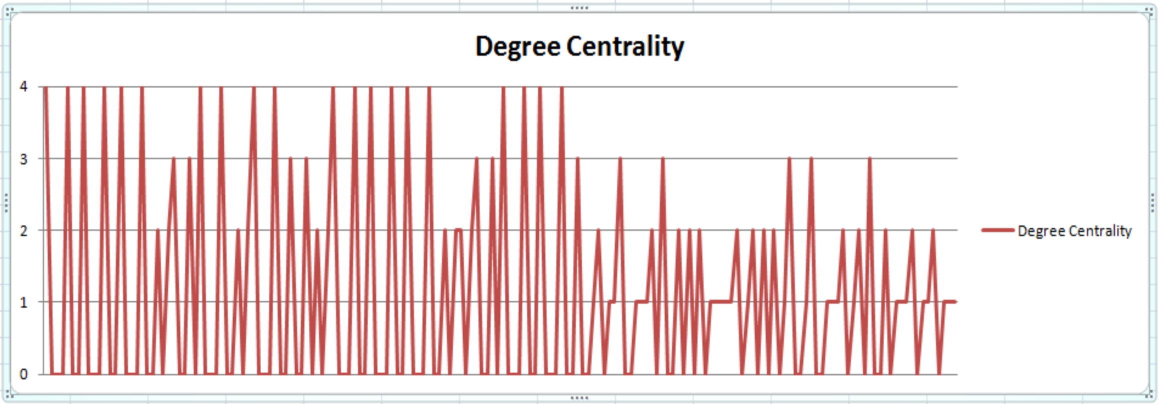


Figure (6.5) the degree centrality of created network

The degree of centrality values for all created agents are ranging between 1-4. The same problem mentioned in section (6.1), which is related to the function of “adults look for friends”, which doesn’t work proporly during the simulation, this problem influences the results when caculating the degree of centrality for agents. In that case only infant and children can have more than one friend according to the pertussis model but adults would not have any friend, so that the degree of centrality is between 1-4 for all agents.

- In the extended model, there are four categories of individuals according to their age. Those groups are infant, child, teenager and adult. By caclculating the average degree of centrality for each age group, and obtaining the minimum and miximum degree, the following values were obtained:

Age Group	Infant	Child	Teenager	Adult
Average	3.0233	2.1176	0.7222	6.2317
Minimum	1	1	2	2
Maximum	4	4	3	4

These values are calculated in the beginning of the simulation, and they reflect the type of relationships in the model.

- By calculating the average degree of centrality for males as well as females and obtaining the minimum and miximum degree, the following values were obtained:

Gender	Male	Female
Average	3.8953	3.6383
Minimum	1	1
maximum	4	4

By repeating the experiment with different neighborhoods, the model start with choosing the most populated 5 neighborhoods which are: “Wesselerbrink NW”, “Stroinkslanden Zuid”, “Helmerhoek-Zuid”, “Helmerhoek-Noord” and “Stevenfenne”.

By implementing the degree centrality measure on the generated population for the five neiborhoods, we get the following results:

- The Total number of all individuals Created: 12233 individuals.
- The pair (Agent ID, Degree) for the whole created agents are shown in figure (6.6) which shows a chart of the calculated degree centrality for number of nodes. As seen from the chart, all nodes have degree centrality with values range from 2-4 according to the consideration of number of children per family and number of friendship per individual.

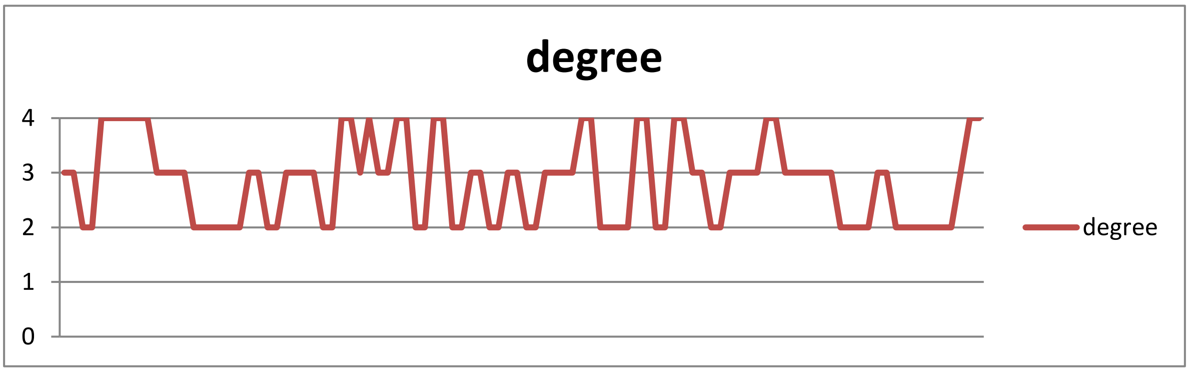


Figure (6.6) the degree centrality of created network

The degree of centrality values for all created agents in this example when more neighborhoods were used, the values are ranging between 2-4. The same problem mentioned in section (6.1), which is related to the function of “adults look for friends”, which doesn’t work proporly during the simulation, this problem influences the results when caculating the degree of centrality for agents. In that case only infant and children can have more than one friend according to the pertussis model but adults would not have any friend, so that the degree of centrality is between 2-4 for all agents.

- In the extended model, there are for categories of individuals according to their age. Those groups are infant, child, teenager and adult. By caclcualting the average degree of centrality for each age group, and obtaining the minimum and miximum degree, the following values were obtained:

Age Group	Infant	Child	Teenager	Adult
Average	3.1736	2.333	0.5146	6.2547
Minimum	2	2	2	2
Maximum	4	4	3	4

- By calculating the average degree of centrality for males as well as females and obtaining the minimum and miximum degree, the following values were obtained:

Gender	Male	Female
Average	3.1422	2.9646
Minimum	2	2
maximum	4	4

6.3. Network Display Configuration

After capturing the network elements and applying relevant measures on the nodes in order to analyze the number of links an agent has, it is the time to display the network elements, as nodes and edges representing the relationships between those agents.

Different conditions could be applied on the elements in order to control the display. In the following examples, the network is displayed by differentiating between males and females agents by color and by giving the infected persons different color. It is also possible to differentiate between age groups by the size of the node. Different layouts were used to display the network elements. These layouts were discussed in chapters (3, 4 and 5).

Depending on the number of agents will be represented in the network, different layouts could be applicable. Starting with the five most populated neighborhoods, with 12233 individuals is the total number of population. Because of the huge number of created nodes, the simplest layout was chosen is the "Random Layout" which is fast and suitable to visualize big network size, but the problem that it gives such a cluttered view in which it is difficult to distinguish between different nodes and different edges as we see in figure (6.7).

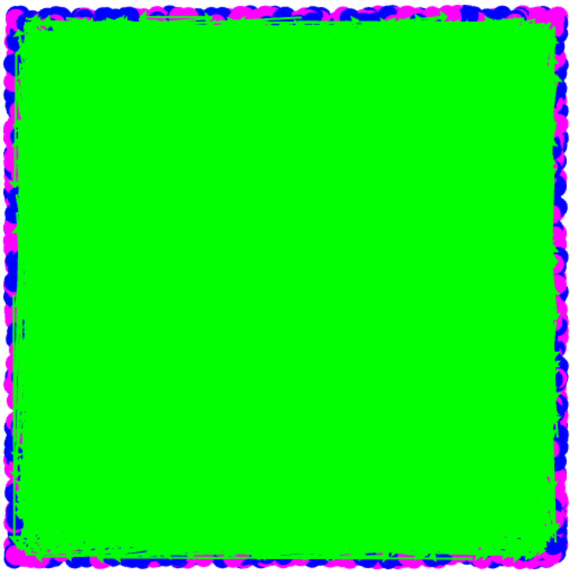


Figure (6.7) display part of the network with Random Layout

Where the blue and pink colors represent the nodes and the green represent the edges. In that case it is very difficult to check the results.

By running the simulation for only two neighborhoods ”City” and “Lasonder 't Zeggelt”, with the same “Random Layout”, we get the following view as illustrated in figure (6.8).

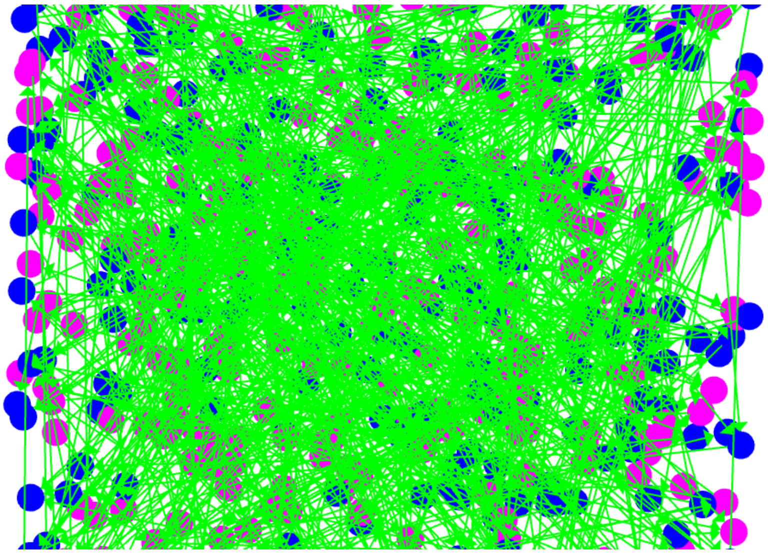


Figure (6.8) display part of the network with Random Layout

By trying another layout such as “Kamad-Kawai Layout” which seeks to achieve the geodesics as distances between pairs of nodes, but this layout is comptational demanding. It gives the following network viewed in figure (6.9), when it is implemented for only two neighborhoods ”City” and “Lasonder 't Zeggelt”, with total number of all individuals Created is 576.

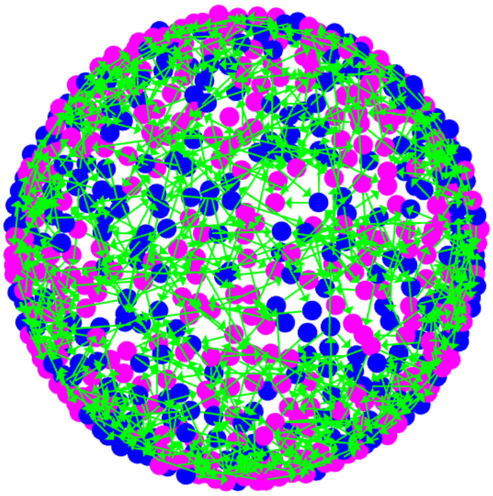


Figure (6.9) display part of the network for two neighborhoods with KK Layout

The following example in figure (6.10) is implemented for only one neighborhood with total number of all individuals created is 195 individuals. The display is dynamically changing to show the increasing number of diseased persons in the network during the simulation. In the display, the “Kamad-Kawai Layout” is used to with the following legend:

● Male individuals.

● Female indivdiuals.

● Diseased persons.

The display is running till the tick number 4800 which means 31 days. The reason of choosing the KK layout because it shows clearly the edges between the nodes.

The reason of using the KK layout is that it aims to map the geodesic distance between vertices in the display to the abstract distance of the graph represented by the geodesics. The algorithm which KK layout depends on is reducing the number of edge crossings, as well as it distributes the vertices and edges uniformly during the update (Kamada & Kawai, 1989).

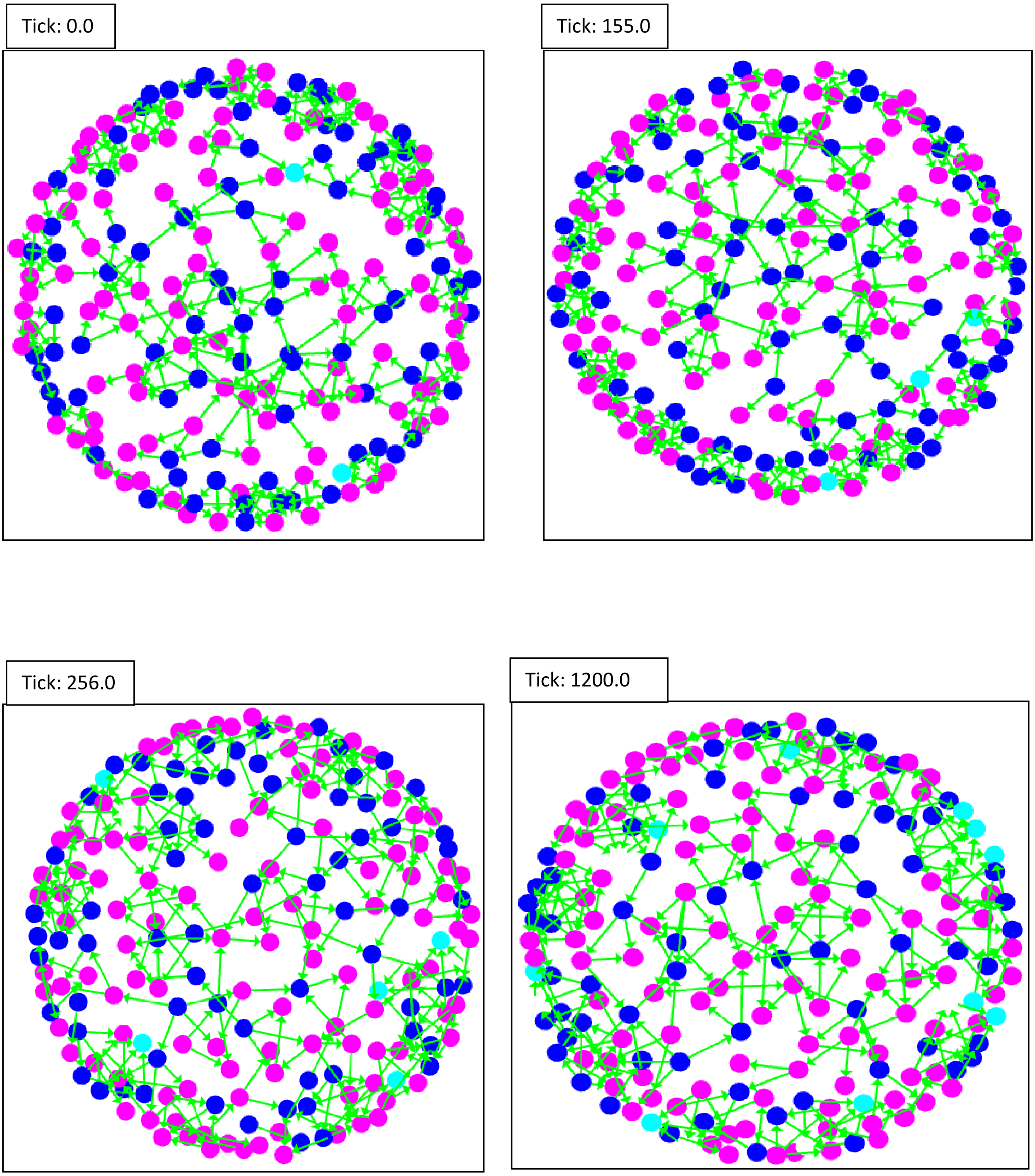


Figure (6.10) display is updated when new person becomes infected

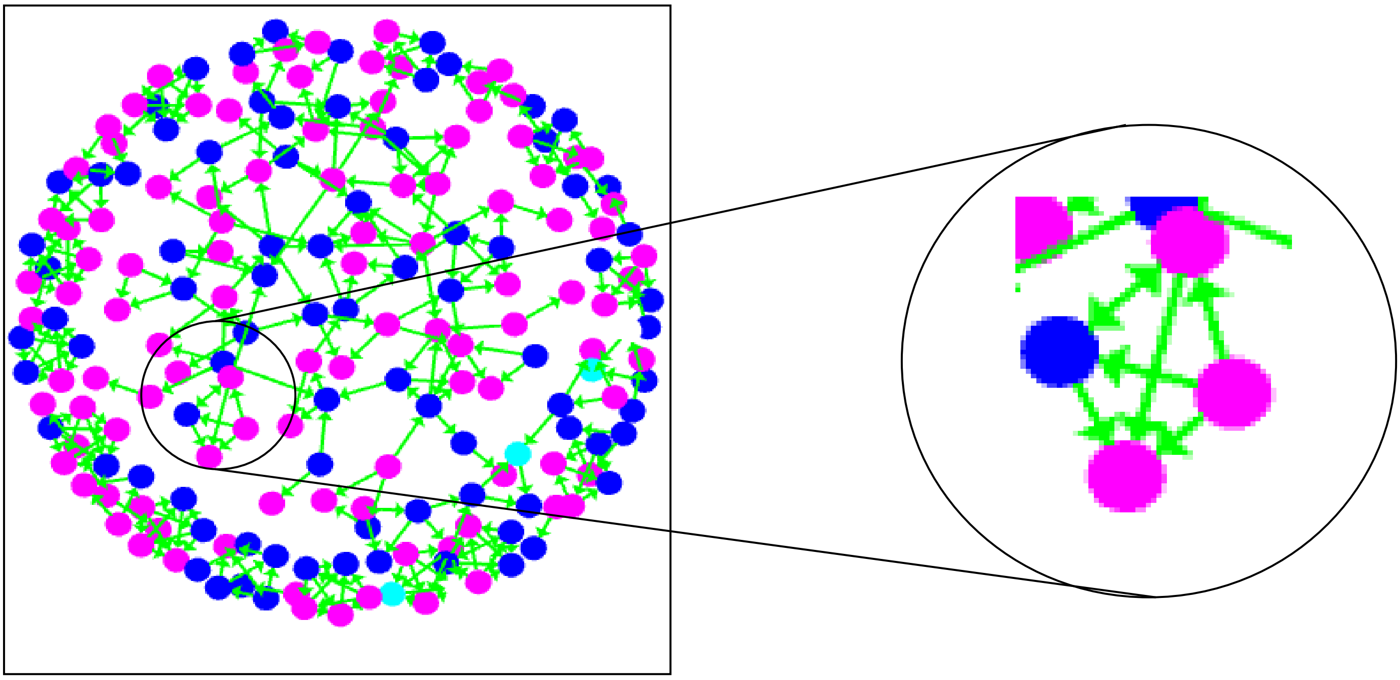


Figure (6.11) display is updated when new person becomes infected

By zooming in to a random set of agents as shown in figure (6.11), the different types of relations between agents are clarified, between partners, between parents and their children and freindship relationships.

7. Conclusion & Recommendations

This chapter seeks to summarize what has been discussed in this research. The first section concludes the findings of this research and the second section recommends what should be done in order to improve the objective of this research and what further improvement could be added to the work.

7.1. Conclusion

The objective of this research was to find and develop a methodology for linking an existing geographically enabled agent based model for disease diffusion with a social relationships network module. The social interaction model was created in order to analyze the agent relations which could influence the spread of infection. When the social interaction network is enabled, agents can send messages over the network that contain information about the disease and change the behavior accordingly. In addition to the visualization of the agents from a social network perspective the aim is also to display the relations between the agents and the analysis results.

In order to create the social interaction network of the disease model, it was proven that it is possible to capture the social interaction network of the spatially explicit simulation model and display it and perform analytical measures on it.

A problem in capturing the social interaction network is the large number of agents/relatiosnhips. A number of approaches to represent the agents with a data structure capable to tackle that large number of agents and their relations have been evaluated. In this research sparse matrices have been used to capture the networks and it has been proven that sthis approach is suitable to store with a large number of agents and links.

The method has been developed to capture the agents and their relationships, incldudes a diffferentiation between different types of relationships between agents. Four types of network relationships have been captured including partner relationship, parent-child relationships, relationships between siblings of the same household and friendship relationships It has introduced number of relations between the model individuals such as unbreakable family members interactions, breakable relations that are considered to be permanent during the simulating and non-permanent relations.

Evaluation of the captured network per type of relationship has proven to be a good way to evaluate and verify that the created synthetic population is realistic.

Variety of network features has been introduced and argued in the literature review that is useful to analyze and measure different interaction networks approaches. Some of these measures are relevant for the case of disease diffusion and the spread of information about diseases. It has been shown that most important measures for the case of information about disease broadcasting are the degree of centrality to determine the number of links an agent has in order to broadcast the messages and closeness centrality which measures the maximum number of links people are apart.

Different social interaction network approaches has been introduced and evaluated in order to evaluate which approach is the better solution to apply from different perspectives, data model and structure to capture the netwrok elements, visualization techniques, SNA routines and the ability of handling large netwrok elements, and it has shown that MORe library is the most promising library to be incorporated with existing models.

Its has shown that the degree of centrality is a good measure to verify the capturing of the network and even the code itself. For the degree of centrality it was shown that some of the relationships are not functioning correctly.

An attempt has been made to display the social interaction network for further validation. The large number of agents and their relationships is a problem when displaying the network as the graphs become over cluttered. Three ways to deal with this problem have been evaluated including: using colour and size, different layout types and displaying subpopulations (queries) Variation in size and colour (nodes) has been very useful in making the graphs more informative. The type of diplay can help to improve this. A number of experiments has been conducted on different layouts and it was seems that Kamada- Kawai is the most promising approach to display the netwrok nodes and the links between them clearly.

It has also conducted a number of displays of results using different neighborhood combinations with various populations, with using different colours and node sizes in order to make the display more readable.

7.2. Recommendations

In scientific researches, everyone tries to put a block on the wall of science, so that the research work still improved continuously.

A. Improvements

The extended model has been developed in this research by incorporating the social network library with the disease model. However, the work still needs further steps to improve it.

In order to improve the reasearch, it is promising to implements other measures that are relevant for the case of disease difusion to achieve more analysis for the network elements. It is also possible to improve the social network library by adding new measures that are relevant also such the measures described in the research.

Because of the problem of displaying large number if network elements in one display, it would be also advisable to use certain queries to display only subsets of agents, such as to split the network into groups according to age, gender, agents’ locations, activities…

In the work, the created network was static, it means that the number of nodes and the edges between the nodes are constant during the simulation, so that it could be a good improvement if the network made network dynamic in order to add new edges when two individuals meet in the model, remove edges, add new nodes or remove nodes.

The extended model and results had been analyzed using synthetic population generated from the disease model during the simulation. It would be a very worthwhile develoment, if can use emperical population data about the social relationships for Enschede region in order to validate and calibrate the analysis results.

B. Further developments

Finally, it would be good if further develoments could be added to the model, by implimenting the messages broadcasing system over the network in order to study how the information about diseases would influence the spread of disease. This implementation would also be possible by implementing a risk awarness system with change of behavior system to let individuals get awarness level depending on their distance from the sourse of information, so that they would change their activities to avoid getting infected. In addition to add the government agent and school which are the most improtant sources of information about the diseases outbreaks.

References

Abdulkareem, S. A. (2010). *Simulating the spread of pertussis in Enschede region using agent - based modelling.* University of Twente Faculty of Geo-Information and Earth Observation ITC, Enschede. Retrieved from http://www.itc.nl/library/papers_2010/msc/gfm/abdulkareem.pdf

Alam, S. J. (2005). Complex Networks Theory and Agent-based Social Networks: A Synopsis. Retrieved from http://www.cfpm.org/~shah/pub/cpm_report_168.htm website:

American Academy of Pediatrics. (2006). 11. Pertussis Introduced in 1952/3 (DTP) DTaP introduced 1996 NOTIFIABLE.

Antia, R., Regoes, R. R., Koella, J. C., & Bergstrom, C. T. (2003). The role of evolution in the emergence of infectious diseases. *NATURE, 426*.

Augustijn, P. W. M., Abdulkareem, S., Huisman, O., & Flacke, J. (2010). Simulating diffusion of pertussis in the Netherlands. *In: Workshop on agent - based simulation of diffusion processes, 8-9 April 2010, Vienna, Austria. - Wien : Universität Wien, 2010. ISBN 978-3-200-01835-8. pp. 31-33*.

Badrow, W., Helfert, M., & Steinicke, M. (2010). *influence of interpersonel perception on information and knowledge sharing in E-learning networks*. Paper presented at the International conference on computer systems and technologies - ComSysTech10, Sofia, Bulgaria.

Batagelj, V., & Mrvar, A. (1999). Pajek – Program for Large Network Analysis.

Batagelj, V., & Mrvar, A. (2003). Pajek—Program for Analysis and Visualization of Large Networks.

Bonacich, P. (1987). Power and centrality: a family of measures. *American Journal of Sociology*, 1170–1182.

Borgatti, S. P. (2002). NetDraw 1.0: Network visualization software. Version 1.0.0.21.

Borgatti, S. P. (2005). Centrality and network flow. *Social Networks*.

Borgatti, S. P., Everett, M. G., & Freeman, L. C. (2002). UCINET 6 for Windows: Software for social network analysis.

Boven, M. V., Melker, H. E., Schellekens, J. F. P., & Kretzschmar, M. (2000). Waning immunity and sub-clinical infection in an epidemic model: implications for pertussis in The Netherlands. *Mathematical Biosciences, 164*(2), 161-182. doi: 10.1016/s0025-5564(00)00009-2

Brandes, U. (2001). A Faster Algorithm for Betweenness Centrality.

Brandes, U., & Wagner, D. (2003). visone - Analysis and Visualization of Social Networks. *Graph Drawing Software*, 321–340.

Caschera, M. C., Ferri, F., & Grifoni, P. (2008). *A dynamic multidimensional visualization method for social networks.* Institute of Research on Population and Social Policies, Rome.

Christley, R. M., Pinchbeck, G. L., Bowers, R. G., Clancy, D., French, N. P., Bennett, R., & Turner, J. (2005). Infection in social networks: Using network analysis to identify high-risk individuals. *American Journal of Epidemiology, 162*(GEOBASE), 1024-1031.

Cliff, A. D. (Ed.). (1981). *Spatial Diffusion: An Historical Geography of Epidemics in an Island community*. New Yourk: Press Sydicate of the University of Cambridge.

Cohen, S., & Brissette, L. (2004). Social Integration and Health: The Case of the Common Cold. *social structure, 1*.

Colizza, V., Barthélemy, M., Barrat, A., & Vespignani, A. (2007). Epidemic modeling in complex realities. *Comptes Rendus Biologies, 330*(4), 364-374. doi: DOI: 10.1016/j.crvi.2007.02.014

Crooks, A., Castle, C., & Batty, M. (2008). Key challenges in agent-based modelling for geo-spatial simulation. *Computers, Environment and Urban Systems*.

De Melker, H. E., Conyn-van Spaendonck, M. A. E., Rümke, H. C., Van Wijngaarden, J. K., Mooi, F. R., & Schellekens, J. F. P. (1997). Pertussis in the Netherlands: an OutbreakDespite High Levels of Immunization with Whole-Cell Vaccine.

Fent, T., Scheffran, J., Prskawetz, A., & C., B. F. (2006). *Agent-Based Computational Modelling: Applications in Demography, Social, Economic and Environmental Sciences*. DE: Physica Verlag.

Freeman, L. C. (1979). Centrality in networks: I. Conceptual clarification. *Social Networks*, 215–239.

Freeman, L. C. (2004). Graphical techniques for exploring social network data.

Freeman, L. C. (2005). graphic techniques for exploring social network data. In P. J. Carrington, J. Scott & S. Wasserman (Eds.), *Models and Methods in Social Network Analysis* (pp. 248-269).

Freeman, L. C., Borgatti, S. P., & White, D. R. (1991). Centrality in valued graphs: a measure of betweenness based on network flow. *Social Networks*, 141–154.

Funk, S., Salathe, M., & Jansen, V. A. A. (2010). Modelling the influence of human behaviour on the spread of infectious diseases: a review. *The Royal Society*.

Glaser, A. (2007). Dynamics and control of Infectious diseases. Retrieved from

Health, W. S. D. o. (2011). Pertussis, Reporting and Surveillance Guidelines.

Hethcote, H. W. (1999). Simulations of pertussis epidemiology in the United States: effects of adult booster vaccinations. *Mathematical Biosciences, 158*, 47-73.

Hethcote, H. W., Horby, P., & McIntyre, P. (2004). Using computer simulations to compare pertussis vaccination strategies in Australia. *Vaccine, 22*.

Hill, A. L., Rand, D. G., Nowak, M. A., & Christakis, N. A. (2010). Infectious Disease Modeling of Social Contagion in Networks. [Article]. Plos Computational Biology, 6(11). doi: e100096810.1371/journal.pcbi.1000968

Holzhauer, S. (2009). Developing a Social Network Analysis and Visualization Module for Repast Models.

Howe, T. R., Collier, N. T., North, M. J., Parker, M. T., & Vos, J. R. (2006). Containing Agents: Contexts, Projections*, and Agents.* Paper presented at the the Agent 2006 Conference on Social Agents, The University of Chicago.

Huisman, M., & Duijn, M. A. J. V. (2003). Software for Social Network Analysis.

Kamada, T., & Kawai, S. (1989). AN ALGORITHM FOR DRAWING GENERAL UNDIRECTED GRAPHS. *Information Processing Letters*.

Kermack, W. O., & McKendrick, A. G. (1927). *A contribution to the mathematical theory of epidemics.* Paper presented at the Royal Society of London.

Klovdahl, A. S. (1985). Social networks and the spread of infectious diseases: The AIDS example.

Lert, F., & Dray-Spira, R. (2002). Social health inequalities during the course of chronic HIV disease in the era of highly active antiretroviral therapy.

Lin, N. (1976). *foundations of social research*.

Meyer, G. G., & Szirbik, N. B. (2007). *Agent Behavior Alignment: a Mechanism to Overcome Problems in Agent Interactions During Runtime.* Paper presented at the the cooperative information agents, Delft.

Nakamya, J., Moons, E., & Wets, G. (2009). how real are synthetic population.

Newman, M. (2010). *Networks: An Introduction*. Oxford, UK: Oxford University Press.

Newman, M. E. J. (2007). The mathematics of networks.

O'Madadhain, J. O., Fisher, D., Smyth, P., White, S., & Boey, Y. B. (2007). Analysis and Visualization of Network Data using JUNG. *Journal of Statistical Software*.

Perez, L., & Dragicevic, S. (2009). An agent-based approach for modeling dynamics of contagious disease spread. *International Journal of Health Geographics, 8*(1), 1-17. doi: 10.1186/1476-072x-8-50

Piunti, M., Castelfranchi, C., & Falcone, R. (2007). Anticipatory coordination through action observation and behavior adaptation *AISB 2007 Symposium Mindful Environments.*

Rummel, R. J. (1976). Social Behavior and Interaction The Conflict Helix (Vol. 2).

Ryan, J., Maoh, H., & Kanaroglou, P. (2007). Population synthesis: comparing the major techniques using a small, complete population of firms. Geographical Analysis.

Sabater, J., & Sierra, C. (2002). Reputation and Social Network Analysis in Multi-Agent Systems.

Scheinkman, J. A. (2009). Social Interactions.

Scott, J. (2000). Social Network Analysis: A Handbook.

Tatara, E. (2007). Repast Simphony reference.

Valente, T. W., & Foreman, R. K. (1998). Integration and radiality: measuring the extent of an individual's connectedness and reachability in a network. *Social Networks*, 89-105.

Wallinga, J., Edmunds, W. J., & Kretzschmar, M. (1999). Perspective: human contact patterns and the spread of airborne infectious diseases. *Trends in Microbiology, 7*(9), 372-377.

Wasserman, S., & Faust, K. (1994). *Social network analysis: methods and applications*.

world Health Organization. (2010). Pertussis Retrieved September 3, 2011, from http://www.who.int/topics/pertussis/en/

Xu, B., & Liu, L. (2010). *Information diffusion through online social networks*. Paper presented at the Emergency Management and Management Sciences (ICEMMS).

Yang, Y., & Atkinson, P. M. (2007). Individual space - time activity-based model: a model for the simulation of airborne infectious-disease transmission by activity-bundle simulation. *Environment and Planning B: Planning and Design, 35*, 80 - 99.

Zhou, Y. (2006). *Explorative research on methods for discrete space/time simulation integreated with the event-based approch and agent concept.* MSc, ITC.