Fall 2014

Discretized Agent-Based Model of Infectious Disease Spread That Uses Contact Probability

Tyrell L. Gardner
Old Dominion University

DISCRETIZED AGENT-BASED MODEL OF INFECTIOUS DISEASE SPREAD THAT USES CONTACT PROBABILITY

by

Tyrell L. Gardner
B.S. Modeling and Simulation Engineering, May 2013, Old Dominion University

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

MODELING AND SIMULATION

OLD DOMINION UNIVERSITY
December 2014

Approved by:

James F. Leathrum, Jr. (Director)

Frederic D. McKenzie (Member)

Yuzhong Shen (Member)

ABSTRACT

DISCRETIZED AGENT-BASED MODEL OF INFECTIOUS DISEASE SPREAD THAT USES CONTACT PROBABILITY

Tyrell L. Gardner
Old Dominion University, 2014
Director: Dr. James Leathrum

This study uses contact probability in an agent-based model to simulate the spread of an infectious disease. In order to perform the study, the agent-based model must first be discretized into events. Each agent in the model is given its own infectious disease state machine taken from the Susceptible-Exposed-Infected-Recovered (SEIR) model. The agents move between squares in a grid environment where each square represents a group. Groups have a contact probability as an attribute that is used to predict whether an agent comes in close contact with another agent. The transitions between the states in the SEIR model are easily translated into Expose, Infect, and Recover events. The Infect and Recover events use the latency period and the infectious period of the disease being modeled. For the Expose event, the contact probability of the group is used to determine what agents the disease is spread to. Due to a lack of available data on the probability of individuals infecting each other, the model output is calibrated to match that of an existing agent-based model as a proof of concept.

This thesis is dedicated to my lovely wife, Shana Gardner, who has helped me in more ways than she knows.

ACKNOWLEDGEMENTS

First and foremost, I would like to extend a special thanks to my advisor, Dr. Leathrum, for contributing to the core idea and helping me get out of the weeds. I would also like to thank former ODU student, Jesse Caldwell, for implementing the first pass of the simulation architecture which served as a great starting point for the remainder of the project.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1
INTRODUCTION

Issues of disease epidemiology have become exceptionally important in contemporary society. Particularly, the spread of viral infections through populated areas has emerged as a crucial public concern, one that has unique and great implication for the modern society. Notably, ever since the start of the new millennium, various epidemics such as pandemic influenza (swine flu), avian influenza (bird flu), and severe acute respiratory syndrome (SARS) have occurred recurrently amongst the human population fostering increased public concerns, mostly over the risk of disease in future, and underlying the need for a better understanding of disease propagation processes and the effectiveness of alternate control methods [1][1].

Agent Based Modeling (ABM) is still a fairly new form of modeling that has grown in popularity due to increasing computational power. This form of modeling has proven useful for many social systems, one of which is epidemiology [2]. ABM captures the local interactions of individual entities in order to produce global phenomena. Many diseases are spread through close contact, or interaction, between infected individuals and susceptible individuals; therefore, ABM has been used extensively to study the spread of diseases.

Another method of studying and predicting disease spread has been using differential equations [3]. This method takes a global view of the system and predicts the spread of disease based on set of equations, but does not take into account the local interactions between entities. Both ABM and differential equations provide useful insight

[1] IEEE Transactions and Journals style is used in this thesis for formatting figures, tables, and references.

into epidemiology, and the decision of which method to use depends on the purpose of the model [4]. Nonlinear differential equation models usually separate agents into a relatively small number of states and assume that the individuals within each state are both well mixed and homogenous [4]. ABMs on the other hand can easily include heterogeneity in both the structure of individual interactions and attributes [4].

This paper takes an ABM approach, but uses a different method than most ABMs to determine when an interaction between agents occurs. In the traditional ABM of infectious disease spread, the agent uses some form of random motion, such as Brownian, to move around the environment, and after an agent moves the agent checks to see if there are any other agents within close proximity. If an infected agent is determined to be close enough to a susceptible agent for an interaction to take place, then the susceptible agent can start its traversal through the states of the disease. The states of the disease are usually borrowed from some previously studied infectious disease model such as the Susceptible-Exposed-Infected-Recovered model. The choice of infectious disease model depends on the disease being studied as well as added factors such as immunity and vaccinations.

Instead of using random motion of the agent to determine interactions, this paper uses the probability of contact between two agents to determine interactions. This approach still gives the agent the ability to move around the environment, but cuts down on extraneous movements where there are no interactions. The contact probability then becomes an attribute of a specific type of group that the agent can travel to. Different groups can have different probabilities of contact between agents, therefore, the only movement that matters to the agent is the movement between groups.

1.1 Purpose

The purpose of this study is to explore an alternative approach of determining when two agents come into contact. This alternative approach may reveal some unique phenomena as well as help to decrease the computational load required by ABM when studying infectious disease spread. The results from this study are compared against a traditional ABM for similarities and differences. Hopefully, the added benefit of being able to simulate an increased number of agents due to the decreased computational load will push for data acquisition of contact frequencies in a variety of settings.

1.2 Problem

The main problem in this study is the danger that infectious diseases present. Although, the danger may seem less real in the United States, this danger still exists in many countries across the world. There is always opportunity for new understanding and a different perspective. With increasing computational power and constantly improving methods of modeling and simulation, it is necessary and beneficial to look for alternative methods of study. In this fashion, better approaches and new ideas are found.

That being said, there are no available data on the probability of contact between two individuals. Although these data would be difficult to gather, the danger imposed by infectious diseases presents motivation enough since most infectious diseases require close contact with an infected individual. It would be beneficial to have data that help to determine when this contact occurs.

1.3 Theoretical Formulations

This study uses the probability of contact between two agents within a group to predict if an interaction will take place. This concept is founded on basic probability theory and random number generation. The theory is shown below.

Let: A = contact event between two agents

Let: P (A) = probability that event A will occur where $0 \leq P(A) \leq 1$

Let: $x \sim U\ (0,\ 1)$

$F(x) = P(X \leq x)\ where\ 0 \leq F(x) \leq 1$

$\therefore If\ \left(x \leq P(A)\right)$

$A\ occurs$

Using the theory above, all that is needed is a U (0, 1) random variable, which is easily obtainable using a linear congruential generator. The randomly generated variate is then compared against the provided contact probability in order to determine if the event takes place.

1.4 Method and Procedure

The first step of this study was developing the model, which can be thought of in two parts, the infectious disease model and the ABM. There were several important steps in this process that should be mentioned. The attributes and methods for an agent and group were defined first, and the control of the infectious disease model was given to the agent. Second, the environment that the agents interact in needed to be defined so that the key events in the model could be determined. Once the events were characterized, the simulation architecture was designed.

The simulation architecture was designed hierarchically for two reasons. First, the hierarchical design is very easy to interface with a variety of applications. This way, if a different application needs to be studied the simulation architecture is reusable. Second, the computational speed of the simulation is improved by using a hierarchical design. One of the most important goals of this study was making the execution of the model as fast as it could be without compromising the results.

Once the simulation architecture design was finished, the implementation of the simulation was completed. This included implementation of the simulation architecture, test harness, and after-action viewer. Testing of the simulation was done for verification purposes.

Since there were no data available to calculate what the contact probability would be, calibration was used to fit the model output to that of another study. The study used for calibration is discussed in [3]. Calibrating the model output to match that of another study demonstrates that the approach used in this study is valid.

In order to accurately calibrate the model, the same scenarios used in the calibration model's study are used for this study. That includes the infectious disease being modeled. The only input parameter that is altered for calibration is the contact probability. All other parameters such as latency period and infectious period are taken from the calibration model study.

The goal is to get the output from this study to match the output from the study used for calibration. This is accomplished by performing the study in three stages with each stage increasing in complexity. The first stage of the study uses a single group that contains all of the agents. There is no movement of the agents within the group, and the

initial number of infected agents is varied according to the scenarios used in the calibration model. Once the output for this stage is calibrated the second stage of the study is initiated. During this stage there are multiple groups each containing approximately equal numbers of agents. There is no movement of the agents between groups or within groups; however, infected agents can expose susceptible agents in neighboring groups. The output for this stage is again calibrated before moving on to the final stage. For the final stage of the study, there are multiple groups each containing approximately equal numbers of agents. This stage enables the agents to move to neighboring groups. When an infected agent moves to a neighboring group the agent can expose susceptible agents to the disease. Once the final stage of the study is calibrated the scope of the study has been reached.

CHAPTER 2

BACKGROUND AND RELATED WORK

In [3], the transmission of a communicable disease within a population is a complex process, which involves a variety of factors [3]. This plethora of factors, objects, and processes exists both at the micro (for a single infection) and macro levels (the infection network) and work together to generate complex disease transmission scenarios, which vary from setting to setting. Yang et al. [1] identify some of these factors to include the infection probability, time lines of contagious diseases, contacts amongst individuals, and the demographic dynamics determining the contact configuration.

Infectious disease transmission may occur through a number of pathways ranging from water, contact with infected individuals, airborne inhalation, to a vector-borne transmission. However, this study assumes that the main pathway of transmission of infectious diseases is human contact; that is direct contact of susceptible individuals with an infected person. For this reason, it is presumed that contagious diseases are spread from person to person following a web of contact amongst them. As this contact typically occurs within a geographical space, it is automatic that the space also has a significant role in infectious disease dynamics. The objective of the current study is to discretize an agent-based modeling approach for the spread of an infectious disease using human contact probabilities.

2.1 Understanding Epidemics: Approaches

Numerous studies have investigated the disease infection or transmission process through experiments that directly infect healthy volunteers with a virus. However, these scientific studies, because of the complexity in the disease transmission process and ethical reasons, often do not elicit feasible solutions for the study of infectious disease propagation. Moreover, these studies generally do not provide practical solutions for the testing of infection control approaches, outside of the experimental setting, particularly across cities or communities [1]. As a result, it is challenging to gain sufficient insight on how the various factors interact in order to predict the occurrence and spread of epidemics and develop effective mitigation approaches. Consequently, in the absence of reliable systems of detecting epidemics, computer modelling and simulation systems have been identified as important tools for both policy-makers and the public as a whole.

Computer models and simulations are significant in providing a universal understanding of the behavior of infectious disease epidemics by investigating the spread of infectious diseases in a given population, with varied demographic and geographic features [5]. Notably, computer simulations promise an enhancement in the representation and conceptualization of the complex social structure alongside the heterogeneous patterns in the contact networks of human populations that determine the dynamics of transmission. Agent-based modeling is a recent approach of sophisticated modeling which has been categorized as a powerful means of understanding the propagation of contagious diseases.

2.2 Agent-Based Modelling

Agent-based models (ABMs) - also referred to as multiagent simulations or multi-agent-based simulation - are computational models that are used to simulate the interactions and actions of autonomous individuals in a heterogeneous population [6]. Fundamentally, agent-based models recreate the whole population and its dynamics by integrating heterogeneous connectivity patterns, social structures, and meta-population grouping at the scale of the single individual [6]. ABMs are bottom-up models whereby, instead of describing the overall universal phenomenon, the phenomenon may be understood by observing the interactions and actions of the multiagent system. In other words, ABMs generate the macro-level behavior of the entire population by simulating the behavior of agents at the micro-level [1]. This phenomenon is labeled as emergence and is one of the unique benefits of using ABMs. Accordingly, ABMs, are the preferred methods of analyzing complex adaptive systems as well as emergent phenomena in various disciplines such as computer science, geography, epidemiology, and other interdisciplinary fields.

2.2.1 Creating an Agent-Based Model

Three factors must be considered in order to create an ABM [6]. The first element is the set of agents which must be autonomous in relation to the other entities included in the simulated setting. The second crucial factor is specification of the interactions of the agents within their group/population and with the shared environment. In this instance, design of all connected aspects is crucial, because these interactions are necessary to yield the overall result. Interactions do not require explicit representation such as within organizational structures; instead they may be represented implicitly, such as in

stigmergic interactions. Conversely, in the applied ABM, it is important to consider the explicit interactions such as organizational structures, even though they may not be obvious [6]. Notably, interactions in this sense may also be the set of rules. The third factor is the simulated environment, which is comprised of all the other components. These may range from other entities without active behavior, resources to universal properties. In addition to the above elements, the simulation infrastructure is crucial for the implementation of the model. Not to be confused with the simulated environment, the simulation infrastructure has functions such as visualization, agent data generation and exportation, and showing the time advance. It is important to ensure that the simulation infrastructure do not interfere with the end outcomes of the ABM [6].

In this simple predator-prey ABM model, wolves act as predators, while sheep are the prey and grass is the resource, in this case, as food for the prey. The set of agents here are wolves and sheep. Agents' behavior includes performing random walks whereby interactions occur when the agents come into contact. TABLE 1 illustrates the possible interactions that can occur in the simulation. Note that grass is not affected by the interactions between prey and predator. Here, the simulated environment could consist of spatial representation of grass objects, global variables such as humidity and temperature that may influence the availability of grass.

TABLE 1

INTERACTIONS IN A PREDATOR-PREY MODEL

	Wolf	Sheep	Grass
Wolf	reproduce	feed-on	-
Sheep	being-eaten	reproduce	feed-on
Grass	-	-	-

2.2.2 Understanding the Agent

While the term "agent" in the context of ABMs has no universally accepted definition, there are certain characteristics that can make up an agent. An agent in an ABM can have its own traits, memory, rules of behaviors, and decision making ability [7]. Moreover, the agent could have the ability to modify and adapt its behavior to the environment.

Some of the key characteristics of an agent include (1) identifiability; (2) existence in an environment with which it interacts alongside other agents; (3) autonomy and self-directedness (4) goals to accomplish with regard to behavior; and (5) flexibility that it can learn and adapt behavior according to experiences [7]. It is not necessary that an agent have all of these characteristics in ABM, however, the agent must have at least one of these characteristics. Figure 1 gives a pictorial representation using UML of the agents in their environment.

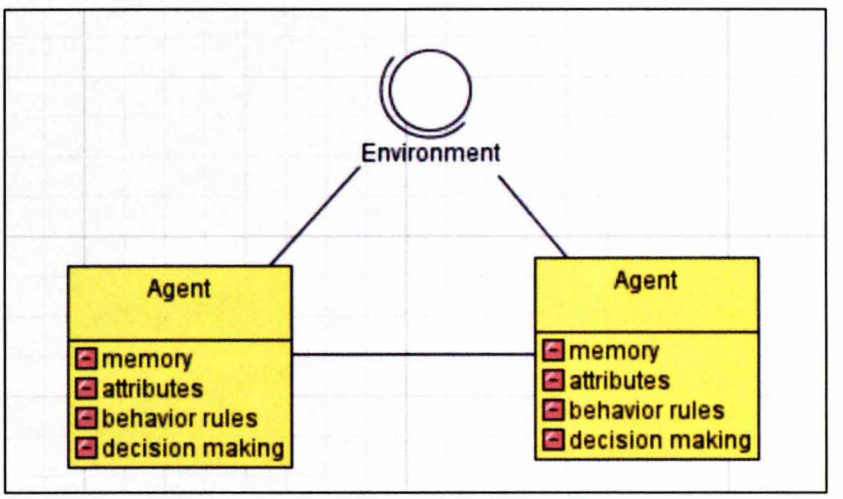

Fig. 1. Agent Characteristics in UML and a Pictorial Representation of the Agents in their Environment

2.2.3 Significance of Agent-Based Models

A key advantage of ABMs is that they are able to factor in the stochastic nature of infectious disease transmission as well as the heterogeneity of both individuals and environment [8]. Further, ABMs are able to demonstrate explicitly, the variations between individuals with regard to the factors that influence the disease transmission process including economic, social, physical, and environmental attributes [1]. For instance, occupation, gender, age and lifestyle characteristics all add to an individual's successive disease experience and their likelihood of infecting other individuals. Accordingly, ABMs enable the interaction among individuals, which is a major factor for predicting the transmission of an infectious disease, as well as overcome the challenge faced by other approaches such as classical epidemic and cellular automata models [3, 9]. Notably, the key difference between agent-based methodologies and cellular automata is entrenched in the fact that in cellular automata, the dynamics of disease transmission are uniform, spatially bound, and based on a permanent neighborhood of each cell [6]. However, in ABMs, even though an agent is also an object that exists in a particular spatial environment, its relationships and interactions to the neighbors are not essentially hard wired [6].

Additionally, ABMs allow for freedom of design meaning that they allow the researcher to consider the heterogeneity of the infection probability, infectious diseases' time lines, and demographic dynamics that define contact patterns [1]. In essence, ABMs and their capability to yield evolving macro-impacts from micro-rules continue to serve as a basis for the advancement of diverse procedural approaches in epidemiology. In most cases, epidemiologic applications that use ABM methodology are created with the

intention of enabling epidemiological researchers to conduct preliminary "what-if" analyses in order to assess the behavior of systems under different conditions as well as determining which alternate control strategies to implement so as to fight epidemics [1, 5]. Another key epidemiological application of ABMs is the simulation of vector-borne pathogens and the variations in their prevalence that may be attributed to climatic changes. Such models have been created to allow epidemiologists to evaluate the climate change effects on vector borne infections such as malaria, and the a priori appraisal of interventions that are environmental management-based [3].

2.3 Infectious Disease Modeling

The transmission of contagious diseases is essentially contingent on the configuration of contacts among individuals [10]. Primarily, factors such as the topological configuration of the population's contact network, the existence of individuals with a much larger contact network than the average number, the clustering and existence of well-demarcated societies of people, and the duration and frequency of contacts have crucial effects for the transmission and control of epidemics [11]. Thus, knowledge of these patterns is particularly vital to inform simulations and computational efforts of understanding the transmission process of infectious diseases. Consequently, the need to conceptualize the behavior of contagious disease spread has led to numerous attempts to simulate and forecast the pattern of various different infectious diseases through a population.

2.3.1 Classic Infectious disease models

The SIR model is the first and simplest model of infectious disease designed by Kermack and McKendrick in 1927 and has been used extensively to advance other more sophisticated infectious disease models [12]. The SIR model is based on commonplace differential equations and was initiated on the assumption that the entire population was susceptible to infections and that total immunity would only be acquired following an infection [13]. In this model, the population is categorized into three categories (see Fig. 2.): susceptibles (S) who are healthy individuals vulnerable to infection by the diseased; infected (I) individuals who are diseased and can spread it; and the recovered (R), containing individuals who have been infected and recovered and are now immune to the disease or if not, removed from the population.

In the simplest SIR model, the evolutions between these categories are all presumed to happen at a ratio proportional to the sum of individuals in each category, and to fixed rates of infection and recovery. In this way, the prevalence rate is proportionate to βSI, whereby β represents the infection rate, while S and I denote the sum of susceptible and infected, respectively. These dynamics of the system are demonstrated using the following differential equations.

$$\frac{dS}{dt} = -\frac{1}{N}\beta SI \tag{1}$$

$$\frac{dI}{dt} = \frac{1}{N}\beta SI - \gamma I \tag{2}$$

$$\frac{dR}{dt} = \gamma I \tag{3}$$

The last equation for the number of recovered is redundant due to the fixed number of individuals in the population N and N = S + I + R at any point in time [14].

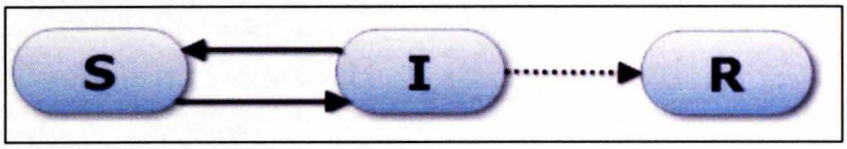

Fig. 2. The Three Categories used in the Susceptible-Infected-Recovered Infectious Disease Model and the Transitions between Them.

Advanced studies have further modified the SIR model to include other categories and vital dynamics. These include SER (Susceptible-Exposed-Recovered); SEIR (Susceptible-Exposed-Infected-Recovered) and MSEIR (Immunized–Susceptible–Exposed–Infectious–Recovered) models. These deterministic models are founded on the assumption that populations are totally mixed and fail to consider spatial effects, and interaction among individuals, since they model populations as continuous entities [3]. Moreover, these models neglect factors such as individual contact process, and the impacts of individual behaviors; hence the need for new computational methodologies.

2.3.2 Complex Systems Approaches for Infectious Disease Modelling

Cellular automata based models are the next level of modeling for infectious diseases, integrating spatial considerations to better represent the natural heterogeneous environment [3]. Cellular automata based models use a two-dimensional cellular automaton to simulate the location of specific attributes of the susceptible group alongside other stochastic parameters that adequately represents the probabilistic nature of disease spread [14]. Characteristically, a cellular automaton is made of a two-dimensional lattice graph in which each node represents a cell. However, the main

problem with cellular automata based models is that they disregard the social behavior and dynamic interactions amongst individuals within the simulated population. Since this is an important element in modelling contagious infections, the cellular automata model gave rise to the new approach, the agent-based models described in Section 1.2.

2.4 Related Work

Over the years, researchers have acknowledged that human behavior has a significant impact in how infectious diseases spread within populations. The need to understand this impact and role of human behavior in contagious disease transmission and control has resulted in a growing body of research evidence, documenting various approaches towards modeling human behavior probabilities and their contribution to infectious disease transmission [13]. In this section, a number of studies are discussed that have been conducted in the areas of infectious disease modeling to build on the understanding of the concept and inform the current study.

The most notable study that has served as a reference for modelling infectious disease spread through populations was the one conducted in [3]. Specifically, the researchers developed a new multi-agent based infectious disease modelling approach (referred as the Perez-Dragicevic model) that sought to explore and better understand the spatial temporal diffusion dynamics of infectious disease spread through a system of human contacts. In their model, Perez and Dragicevic incorporated geographic information systems to simulate the spread of infectious disease in an urban environment using a measles outbreak in a city [3]. The model used the SEIR (Susceptible–Exposed–Infectious–Recovered) disease model and was based on georeferenced land use data from

Metro Vancouver and used census data from Statistics Canada for the municipality of Burnaby, BC, Canada [3]. The objective of this model was to illustrate the progression of the disease based on people's interactions by calculating ratios of susceptible/infected at particular time frames and urban environments. The results of the study showed that dynamic spatial interactions among the populations resulting in huge numbers of exposed people who conduct stationary actions in different areas following commuting. Consequently, the sick people congregate in geographical locations such as schools and universities thus enabling specialists to formulate effective prevention and control measures in the case of an outbreak [3].

In [12], an agent-based approach to simulate the spread of pandemic influenza in Egypt was used. The study's model simulated individuals' interactions in a spatial-time context. The model factored in various types of behavioral parameters including social agent attributes, patterns of agents' interactions, and the distribution of Egypt's population. Specifically, this study extended the SIR model by adding new classes simulating the real pandemic behavior and control states including (in contact, quarantined, not quarantined, dead, and immunized). Figure 3 depicts the model used in this study.

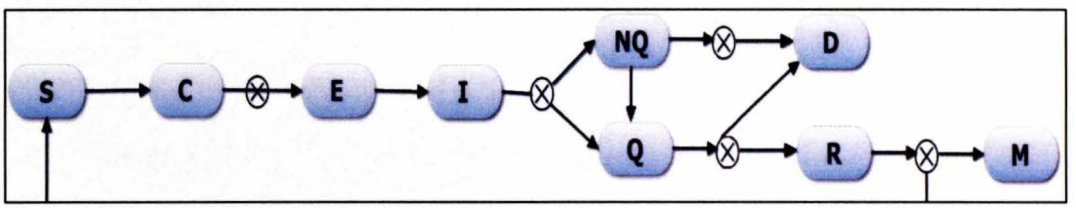

Fig. 3. State Chart of Khalil et al.'s Extension to SIR Model. (S) Susceptible, (C) in Contact, (E) Exposed,(I) Infectious, (Q) Quarantined, (NQ) Not Quarantined, (D) Dead, (R) Recovered, and (M) Immunized.

The findings of this model were significant to the understanding of the particular features of the modeled pandemic, the pandemic's transmission patterns, and the circumstances under which an outbreak could happen. Additionally, the model was effective in measuring the efficacy of various control strategies to intervene the spread of the pandemic [12].

The study performed in [10] sought to examine the role of dynamic contact patterns among individuals, particularly their temporal aspects, in determining the transmission of an epidemic through a population. The researchers used the SEIR model to simulate the spread of an epidemic using the face-to face interactions of conference attendees. The study demonstrated that an agent based model that takes into consideration the aggregated network and duration of contacts among agents in the model provides an effective approximation of the human contact network during an epidemic. The study emphasized the impact of contact heterogeneity on the dynamics of infectious diseases which is significant in predicting the impact of control or preventative techniques [10].

In a different approach, [15] formulated a metapopulation model that integrated various scenarios of self-initiated behavioral variations in response to epidemic outbreaks into the movement patterns of individuals. Specifically, this model focused on exploring whether prevalence-based travel constraints impact the epidemic invasion threshold. This was accomplished using a mechanism that gave the simulated individual the proclivity to circumvent epidemic affected areas. According to the results of the study, prevalence-based behavioral modifications have no significant impact on the invasion threshold; even though the number of infected subpopulations relied significantly on the behavior of

the entire population [15]. Therefore, while the objective of such self-initiated behaviors among travelers is to avoid exposure to the infection, [15] shows that such behavior may result in an unexpected effect of facilitating and increasing the disease's spread to new locations and subpopulations. The study emphasized the inclusion of mobility and human behavior in this process as a key element in modelling the geographical transmission of epidemics [15].

In [1] the effectiveness of non-pharmaceutical infectious disease control measures such as school closure; refraining from social activities, and household quarantine was studied. The researchers used the Individual Space-Time Activity-based Model (ISTAM), which is a bottom-up ABM within which the disease transmission was developed on the simulated physical contacts among people as a fine space-time scale. The model factored in parameters such as human social behavior, transmission mode of the disease and the physical environmental conditions and relied on the activity bundle (AB) concept [1]. The AB concept was significant to this model since it aided in the construction of individual activity patterns as well as enhancing simulation amongst the ABs. According to the findings elicited from the various AB classes, it appears that infectious disease transmission first strengthens among children then among adults. Therefore, the results of the study showed that household quarantine alongside school closure served as the best method of controlling an epidemic [1].

Finally, [16] simulated the transmission of an infectious disease using the classic complex systems approach. The key objective of the study was to identify why government policy makers and health practitioners encounter numerous challenges in formulating effective interventions for epidemics. The study also intended to show the

complex nature of infectious disease transmission such as its dependence on individual's susceptibility. The study also investigated other factors that compounded the ability to understand the spread of infectious disease transmission such as the type of epidemic, feedback mechanisms and the rate of transmission [16]. The study's model was based on the classic infectious disease model and incorporated probabilistic simulation software to demonstrate rate of disease transmission. According to its findings, the study demonstrated the value of the classic infectious disease modelling approach, though there exists a chance that the model may only be effective in epidemics where the main mode of transmission is through local individual contacts [16].

CHAPTER 3

MODEL DESIGN

This chapter gives a detailed overview of the model used in this study. The majority of agent-based models of infectious disease propagation take a time-stepped approach. An agent may move some randomly determined direction and distance and then check to see if it is close enough to another agent for spread of the disease. Although this approach is believed to add a sense of realism to the model by moving the agents around the environment in some human fashion, there tends to be some events that are unnecessary. Are the five steps of the agent which brings it close enough to another agent in order to spread the disease really needed? Some would argue that they are, but this study takes a different approach by discretizing the model into events.

In order to do this, the events that take place in the model need to be clearly defined. It is possible that certain models are not easily discretized; however, the infectious disease model is not one of these. This chapter gives an overview of the model used for the agents as well as the mathematical model used to predict the spread of the disease. Finally, the environment model is explained.

3.1 Agent Model

This section gives a detailed description of the agent model used in this study. First and foremost, the type of infectious disease model is discussed. This model is used to clearly define each of the events that take can occur. For this stage in the discussion, the basic functionality of each event is given rather than the inside decision logic. In this fashion, the events can be described in a more realistic way that lends itself more easily to the real world.

3.1.1 Infectious Disease Model

For this study, the infectious disease model found in [3] as discussed in Section 2.3.1 is used. This model is the SEIR model and can be viewed as a finite-state machine. Figure 4 shows this state machine with the circles representing states and the arcs representing transitions between the states. There are two approaches that can be taken when merging this model with the agent. The first approach, employs only one state machine with all of the agents existing in one of the states at any given time during the simulation. The second approach gives each agent its own state machine, thereby, enabling the agent to determine when and if a transition is going to take place. The second approach places the decision logic on the agent, which is desirable when using agent-based modeling.

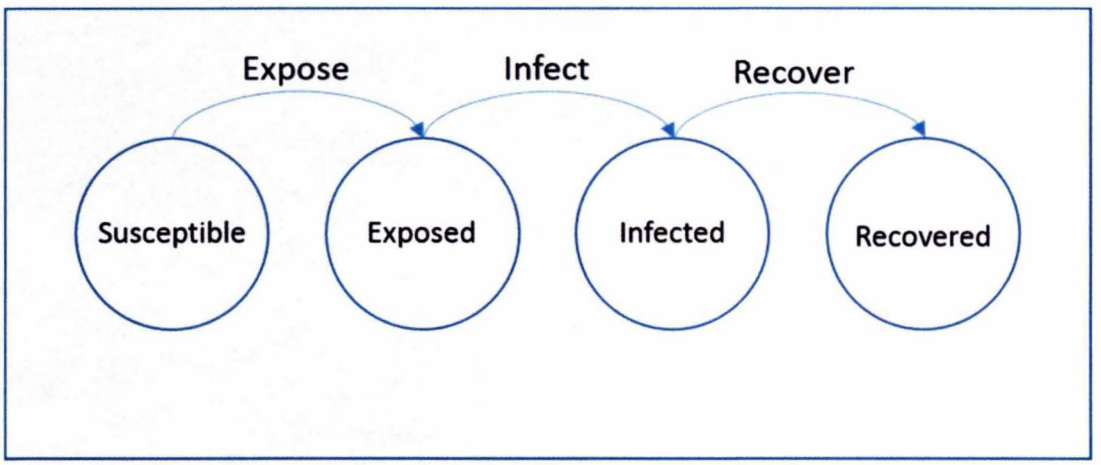

Fig. 4. Finite-state Machine Representation of the SEIR Infectious Disease Model.

All agents must be in one of the four states of the model. The agents that are in the susceptible state can be exposed to the disease being modeled. Only agents that are in the infected state can expose susceptible agents. Once an agent reaches the recovered

state, it is no longer susceptible to the disease and can be thought of as removed from the simulation. The events that occur in the model can be taken right from the state machine. These events are Expose, Infect, and Recover. The functionality of each of these events is discussed in the next section.

3.1.2 Event Functionality

In order to transition from the susceptible state to the exposed state an Expose event must take place. This is the event that occurs when an infected agent comes into close contact with a susceptible agent. The key words here are close contact. In order to discretize this infectious disease model an alternative method of determining when an agent comes into contact with another agent. In the traditional agent-based models, this is determined by having the agent continuously step in the environment until it reaches another agent. In order to remove these extraneous steps the notion of contact probability is used. Contact probability is the chance that an agent will come into close contact with another agent that is in the same group. By using this probability, no movement of the agent within a group is necessary.

When an infected agent is introduced to a new group, the agent uses the contact probability of the group to predict which susceptible agents are going to be exposed to the disease. This is the basic functionality of the Expose event.

Once an agent has been exposed to the disease, there is a disease specific period of time until the exposed agent becomes infectious. This leads into the functionality of the Infect event. Depending on the disease being modeled, the latency period of the disease is used to determine when the Infect event is going to occur. In addition to

transitioning the agent from the Exposed state to the Infected state, this event also must determine which susceptible agents are going to be exposed to the disease by the newly infected agent.

The final transition of the state machine is the Recover event. In addition to having a latency period, most disease also have a specific infectious period. This disease specific period is used to determine when the Recover event will take place. The basic functionality of this event is to move the agent from the Infected state to the Recovered state.

One final piece is missing from this model. In order for an infected agent to spread the disease, it must be able to move between the groups. This Move event should not be thought of as the same movement that takes place in the traditional agent-based models, but rather as the movement that matters. This movement matters because it is the move that makes it possible for either an infected agent to expose new susceptible agents or a susceptible agent to get exposed by some infected agent. With movement between groups enabled, the scope of this study is reached.

3.2 Mathematical Model

Now that the basic functionality of each event has been discussed, the mathematical model is introduced. This is the model used to predict when an agent comes in close contact with another agent. In addition to going over the contact probability model, the inside decision logic of each event is explained giving a more in-depth look at what happens when an event takes place.

3.2.1 Contact Probability

This unique approach to discretizing the infectious disease model used in this study employs the use of probability in order to predict if an event will take place. The contact probability can be thought of as group specific. There are certain groups that are more conducive to interaction than others. These groups would have a higher contact probability than others. By employing the use of contact probability this agent-based approach to infectious disease spread is discretized.

The contact probability is used in a very simple algorithm that predicts if an exposure is going to take place. Note, that this probability does not in any way determine when the event will occur. The algorithm for exposure is:

$For\ (each\ susceptible\ agent\ in\ the\ same\ group)$

$Generate\ x \sim U(0,1)$

$If\ (x \leq contact\ probability)$

$Determine\ the\ time\ of\ exposure$

$Schedule\ the\ exposure\ at\ this\ time$

This very simple algorithm is used in both the Infect event and the Move event. Before delving into the decision logic inside each event, the method of determining exposure time as well as the latency time and infectious time of the disease being study must be discussed. There are a multitude of data available on the latency period and the infectious period of many disease. Using input analysis, a probability density function can be fitted to the data and used to generate random variates for the time of infection and the time of recovery. In order to determine the time of exposure, an assumption must be made about the model. This assumption is that if an agent is predicted to come into close

contact with a susceptible agent, then the exposure will take place within one 24 hour period. The probability that the exposure will take place is uniformly distributed across this time period. This assumption is then used to fit a uniform distribution to the exposure time.

Lastly, the time of movement needs to be determined. This is a combination of the amount of time an agent spends in a group and the amount of time it will take the agent to travel to the next group. For the first piece, the assumption that an agent will stay in the group anywhere between 15 minutes and eight hours. For this study, this time span was assumed to be uniformly distributed as well. The travel time is assumed to be uniformly distributed between ten minutes and one hour. These two random variates are generated and then combined together to determine when the agent will actually arrive in the next group.

3.2.2 Event Decision Logic

This section gives the algorithms used in each of the events. These algorithms provide the framework for this study. First, the three events of the infectious disease model are reviewed. Lastly, the Move event is discussed, since this event employs a similar algorithm to the Infect event. The decision logic discussed next can easily be translated into any computer coding language.

3.2.2.1 Expose Event

The algorithm for the Expose event is:

Change current state to Expose

Determine latency period of disease

Schedule Infect event at current simulation time plus latency time

3.2.2.2 Infect Event

The algorithm for the Infect event is:

Change current state to Infect

Determine the infectious time of the disease

Schedule Recover event at current simulation time plus infectious time

For (each susceptible agent in the same group)

Generate $x \sim U(0,1)$

If $(x \leq$ *contact probability)*

Determine time of exposure

If (exposure time < next move time)

Schedule Expose event at current simulation time plus time of exposure

3.2.2.3 Recover Event

The algorithm for the Recover event is:

Change current state to Recovered

3.2.2.4 Move Event

The algorithm for the Move event is:

If (current state equals Infected)

For (each susceptible agent in the same group)

Generate $x \sim U(0,1)$

If ($x \leq$ *contact probability*)

Determine exposure time

If (expose time < last expose time)

If (expose time < next move time)

Schedule Expose event at current simulation time plus exposure time

Else If (current state equals Susceptible)

For (each infected agent in the same group)

Generate $x \sim U(0,1)$

If ($x \leq$ *contact probability*)

Determine exposure time

If (expose time < last expose time)

If (expose time < next move time)

Schedule expose event at current simulation time plus exposure time

Randomly choose neighbor group

Determine time spent in group

Determine travel time

Schedule Move event at current simulation time plus time in group plus travel time

3.3 Environment

The environment is the last piece needed to fully explain the model. For this study, the environment can be thought of as a set of groups in which the agent can travel between, and a group can be viewed as the context that the agent is in. Each individual group can represent anything from an agent's home or workplace to an entire college campus. What each group represents depends on the application and at what scale the application is going to run.

That being said, the groups need to be defined abstractly to start with in order to maximize the variety of applications that could be used. At the simplest level, a group can be thought of as one square in a grid and the environment can be thought of as the entire grid. An example of a 2 X 2 grid is shown in Figure 5 with the points representing agents. There are a couple desirable characteristics inherently applied when the environment is represented as a grid. First off, each square has easily identified other squares that share a side, which can be defined as neighboring squares. If an agent is in one group, then the neighboring groups represent the groups that the agent can travel to. Lastly, the grid is very easily scaled both up and down. If it is desirable to study a smaller group setting, then each of the groups can be divided into four more groups. The same is true for the reverse situation. If a larger group setting is needed, then each square group of four can be combined into a single group.

The only other thing needed to fully define this abstract sense of the environment is the agent. It is desirable for each group to have a list of the agents that are currently in the group. When an agent moves from its current group to one of the neighboring groups, the agent is removed from the current group's agent list and added to the neighboring

group's agent list. If a susceptible agent is scheduled to move to a neighboring group before an exposure is going to take place, then the exposure will not occur. In this fashion the movement of the agents between the groups has an effect on both spreading the disease and determining if an exposure is going to occur.

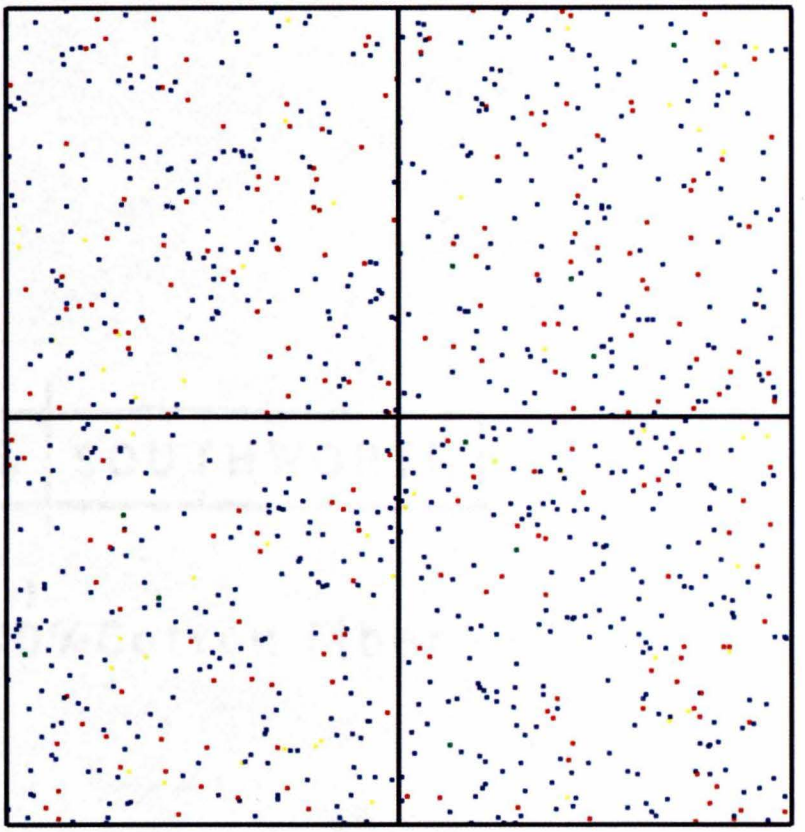

Fig. 5. Pictorial representation of the grid environment used for this study.

The model consists of the grid, which is populated by agents that each have their own infectious disease state machine. As the agents move between groups, new events occur that change the disease state of the agents.

CHAPTER 4

IMPLEMENTATION

This project is implemented using the C++ programing language. This language is chosen for its ease of object-oriented design implementation. Figure 6 depicts the simulation architecture and agent architecture used for implementation. Only the important attributes and methods are included in each class for simplification purposes. The classes can be divided into two main sections: simulation architecture and agent architecture.

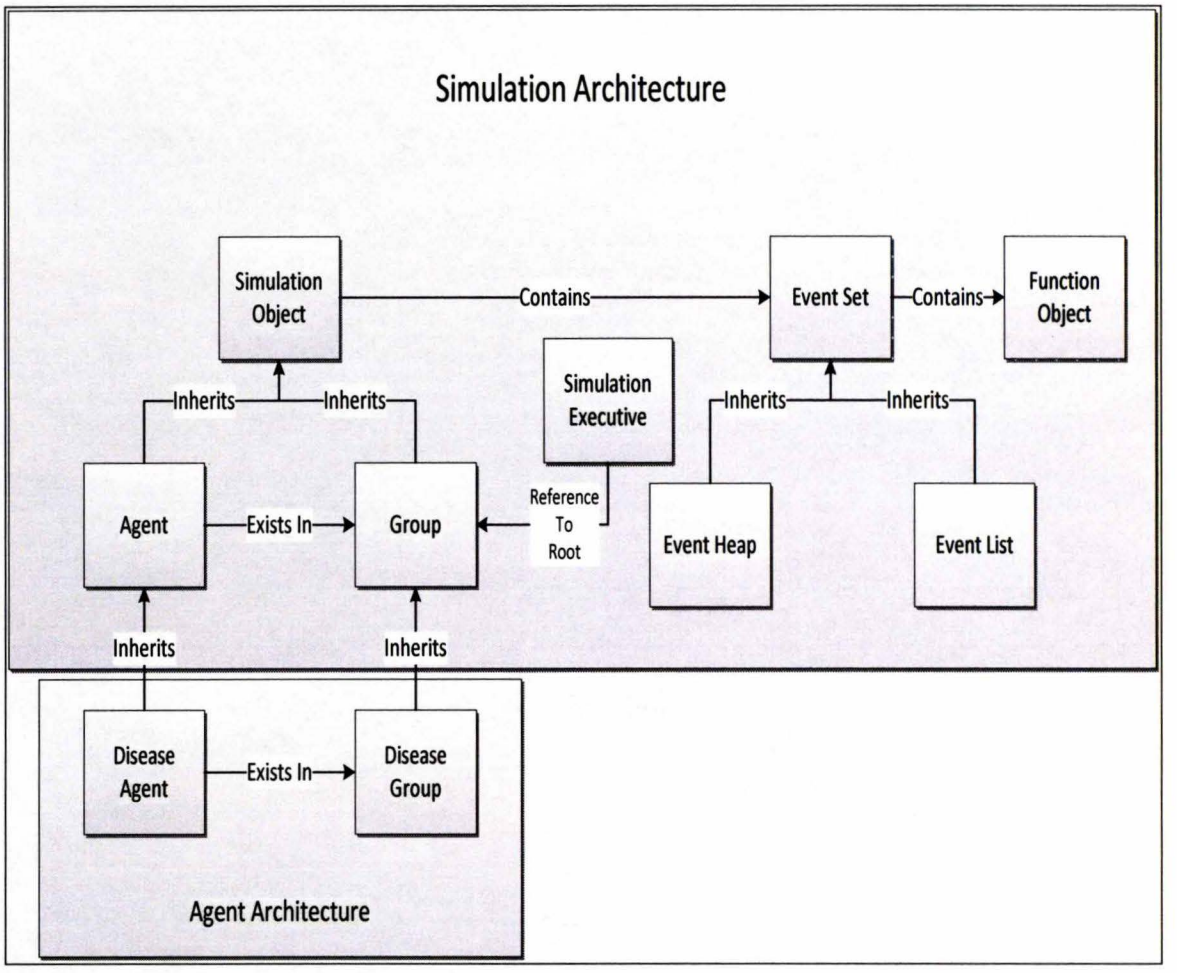

Fig. 6. Simulation Architecture and Agent Architecture Used for Implementation

The link between these sections is the Function Object, which enables the agents to create events that are recognized by the simulation. This will be explained in greater detail at the end of this chapter. The simulation architecture is now introduced.

4.1 Simulation Architecture

The simulation architecture is composed of the simulation executive, event set hierarchy, simulation object, agent, and group. The simulation executive acts as the engine in that it executes events in time-stamped order until there are either no more events or an end time or condition is reached. The event set stores all events in time-stamped order making the execution of events easier on the simulation executive. There are two inheriting classes of the event set: event heap and event list. An event heap is modeled after the heap data structure in that the root is always the next event to be executed, or in this case the event with the lowest execution time. An event list is just an ordered set of events with the next event to be executed being the first item in the list. Depending on the application a heap or list may be desirable. Figure 7 depicts the UML class diagram for the simulation architecture.

The functionality of the simulation executive is explained next as well as the functionality of the important methods. After that, the event set class and methods is introduced. Since the methods of the inheriting event heap and even list employ exactly the same functionally as the event set, the explanation is limited to that of just the event set.

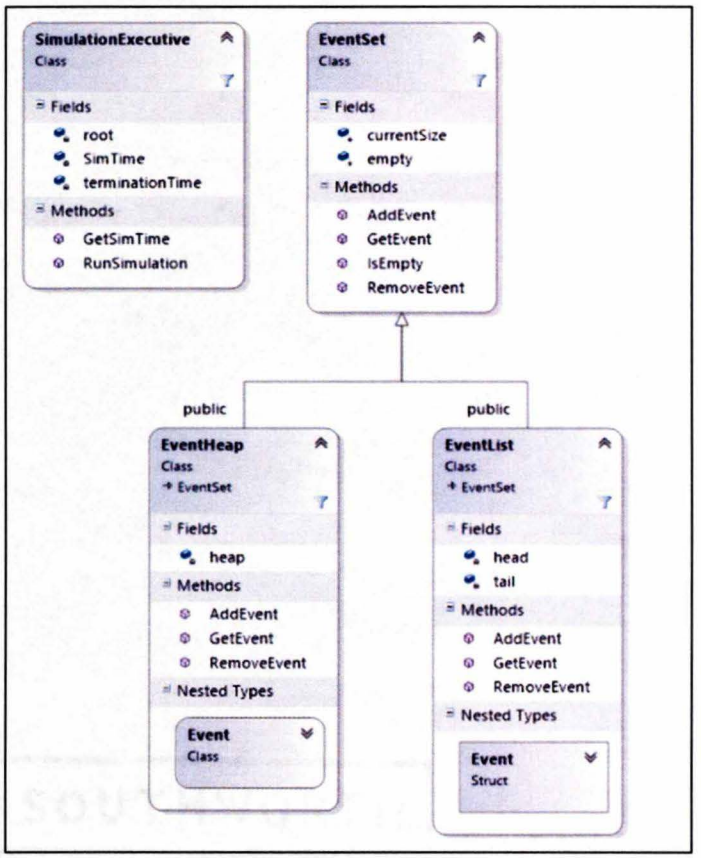

Fig. 7. UML Class Diagram of the Simulation Architecture.

4.1.1 Simulation Executive

The simulation executive ensures that all events are executed in time-stamped order. In addition, the simulation is stopped according to the desired stopping condition. This stopping condition could be when there are no more events or a specified end time is reached. This logic is handled in the run simulation method. Although most simulation executives would contain an event set, the simulation executive used in this study does not. Instead, the event set is stored with the agents and the groups, and the simulation executive has a reference to the root of the group heap.

4.1.1.1 Run Simulation

This method loops until the current simulation time is greater than the stopping time or there are no more events. In each loop the root group is executed and then the current simulation time is updated. The execution of the root group is explained later in the group class.

4.1.2 Event Set

The event set is used to store all events to be executed during the simulation. This class acts as a base class for the event list and the event heap. As mentioned earlier, whether or not a heap or a list is used depends on the application. In this study, the heap is beneficial for the groups and the list is beneficial for the agents. Heaps work great when there is a large number of events because the decreased time in adding events outweighs the increased space needed for memory allocation. This is desirable for the groups since they are expected to have a large number of events at any given simulation time. Lists work well when there are relatively few items to deal with. This is desirable for the agents because they only ever have at most two events to be executed. A heap would be overboard in this case because the added benefit of decreased adding time is lost when there are only two items in the heap.

There are four main methods used by the event set. These methods are "add event", "get event", "remove event", and "is empty". The functionality of these methods is explained next.

4.1.2.1 Add Event

In the case of an ordered event set, the event to be added must be added in such a way that the set retains its property of order. Although the heap and the list accomplish this in different ways, the end result is the same. For the event list, this is the most computationally intensive method because in a worst case scenario the entire list must be iterated over. The worst case scenario for the heap is the number of levels in the tree. This tends to make the heap much faster at adding as the set gets larger.

4.1.2.2 Get Event

This method gets the next event to be executed in the set. The next event is the event with the lowest time-stamp. This method tends to be very fast for the event list as the next event is the first item in the list. In the case of the heap, some manipulation of the heap must be performed in order to retain the heap property making this method more computationally intensive for the event heap. For this reason, it is only beneficial to use the heap if it is known that the set is going to be large.

4.1.2.3 Remove Event

Each event has a unique identifier, which is used to locate the event. This method iterates over the set until the specified identifier is found. Once the identifier is located, the event is removed from the set. This will be more computationally intensive for the heap because in a worst case scenario the entire tree will have to be iterated over and

once the event is removed some manipulation must be made in order to maintain the heap property. For the list, even though the entire list may need to be iterated over, when the event is removed the list can just be reconnected and no other manipulation needs to be performed. For this reason, the application developer needs to weigh the size and number of removals that are going to be made in order to determine whether a list or heap will be more beneficial.

4.1.2.4 Is Empty

This is a very simple method that checks to see if the set is empty. If the set is determined to be empty, then a Boolean value of True is returned. Otherwise, a Boolean value of False is returned.

4.2 Agent Architecture

The agent architecture is made up of an inheritance hierarchy of simulation objects. A simulation object is any object in the simulation that needs to execute an event. The hierarchy starts with the base class simulation object. For this project, the only two objects that need to execute events are agents and groups both of which inherit the base class simulation object. Before the core of the agent architecture is explained, the concept of command patterns must be introduced. Once this design concept has been explained at a high level, the implementation of the agent object and the group object is explained.

4.2.1 Command Pattern

The command pattern design concept makes it so that the simulation can store events of one type to be executed at some future time. For a more in-depth look at command patterns see [17]. In order to accomplish this the function object base class is used. This base class has one virtual function labeled execute that is implemented in each of the inheriting classes. By making the function virtual, each of the inheriting classes can provide a unique implementation of the execute method.

For each event in the simulation one of these function object sub-classes is implemented. These sub-classes enable the simulation executive to execute the desired method, also known as the event in this case. In order to execute a method a reference to the simulation object calling the method as well as any parameters used by the method is needed. These items are passed during instantiation of each of the function object sub-classes. In this fashion, when the Execute method for the function object sub-class is called, the reference to the simulation object and any parameters are used to execute the event. This elegant design concept makes it possible to create an event list that stores only function objects. Now that the link between the simulation architecture and the agent architecture has been introduced, the simulation object, agent and group can be explained.

4.2.2 Simulation Object

The simulation object is the base class for all objects in the simulation that have events that need to be executed by the simulation executive. The only attribute used by

the simulation object is an event set. This makes it so the inheriting classes can use either an event list or event heap since the event set is the base class for these two sub-classes. All simulation objects need to perform three tasks. These tasks are add event, remove event, and get event. A method has been implemented for each of these tasks and is explained next. Figure 8 shows the UML class diagram for the simulation object.

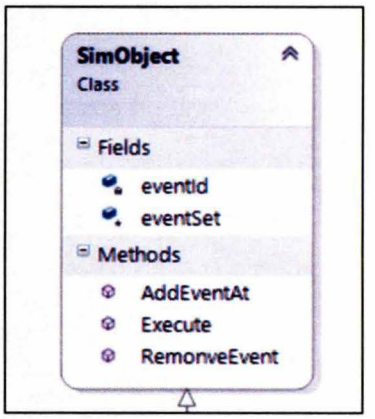

Fig. 8. UML Class Diagram of the Simulation Object.

4.2.2.1 Add Event

This method adds an event to the event set. In order to accomplish this, two parameters are needed, the function object and the time of execution. With these two parameters the function object can be added to the event set in order.

4.2.2.2 Remove Event

The remove event method plays an important role in this simulation. For example, if two events of the same type are to be executed on the same simulation object at two

different times, then only the event with the earliest time stamp matters. This method removes the event with the later time stamp so that the simulation executive only executes the event with the earliest time stamp. This is why each type of event has a unique identifier, which enables the simulation object to remove the event with the later time stamp.

4.2.2.3 Execute Event

This method gets the event in the event set with the earliest time stamp and executes it. Since all of the event sets in this simulation are ordered, this should be the first event in the set. By doing this, the event is removed from the event set, and simulation executive can update the current simulation time to the time of the executed event.

4.2.3 Agent

The agent inherits from the simulation object and acts as a base class for application specific agents. Each agent has a unique identifier so that it can be tracked as the simulation progresses. In order to make the agent independent of the desired application there is very little functionality built into the agent other than an implementation of the execute event method inherited from the simulation object. Each agent will have a reference to the current group it resides in. The functionality of the groups will be explained later in this chapter. For this project, a disease agent is used, which is a sub-class of the agent. Figure 9 shows the UML class diagram for the agent.

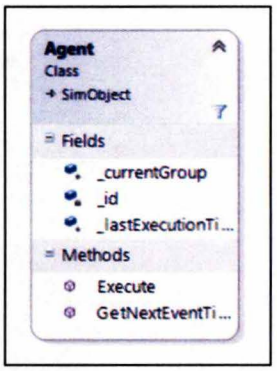

Fig. 9. UML Class Diagram of the Agent.

4.2.4 Disease Agent

This object provides the majority of the functionality for the simulation. Using the SEIR disease model previously discussed, each disease agent will exist in one of the four disease states. The disease agent will also reside in a disease group and have a current position within this disease group. This disease group is a sub-class of the group and will be explained later in this chapter.

There are several methods used by the disease agent. The first is the initialization method, which seeds the simulation with a specified number of infected agents. Without this method, the simulation would not be able to progress because in order to spread the disease a disease agent must first be infected. The remaining methods are Expose, Infect, Recover, and Move. Each of these methods, including the Initialize method, are the main events of the simulation and will have an implemented function object to provide the link between the agent architecture and the simulation architecture. These methods are explained in detail next. Figure 10 shows the UML class diagram for the disease agent.

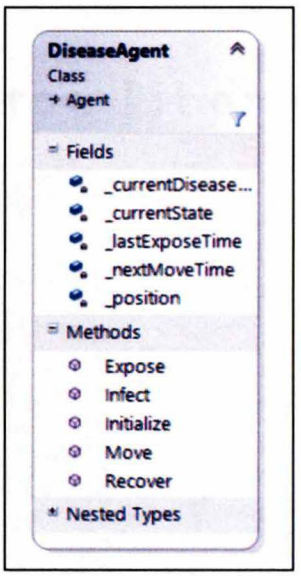

Fig. 10. UML Class Diagram of the Disease Agent.

4.2.4.1 Initialize

As previously mentioned, the simulation needs a specified number of infected disease agents in order to run. The exact number does not matter as long as there is at least one infected disease agent. Otherwise, every disease agent would be susceptible to infection, and there would be no spread of the disease. This method enables the user to provide a different number of initially infected disease agents and observe how this affects the spread of disease.

4.2.4.2 Expose

This event transitions the disease agent from the susceptible state to the Exposed state. The Exposed state acts as an incubation period for the disease being modeled.

During this period the disease agent is not yet infectious. The incubation period can be modeling using a probability density function. Using a stochastic incubation time each disease agent will become infectious at a random.

4.2.4.3 Infect

When a disease agent becomes infected with the disease they are able to pass the disease to other agents. The infect method provides this functionality. As soon as an agent becomes infected the agent will expose all agents in the same disease group that it comes in close contact with. In order to determine what close contact is the contact probability of the group is used. This is a unique method of modeling the spread of disease using a simple probability. All susceptible disease agents in the infected agent's current disease group are iterated over and if a generated uniform random number is less than the disease groups contact probability the susceptible agent is exposed to the disease.

Another fact must be taken into consideration here though. Not only are the disease agents moving through the SEIR model, they are also moving between the groups. In order for an infected disease agent to expose a susceptible agent, the susceptible agent must be in the same group at the time of exposure. In summary, the susceptible agent will be exposed if the generated contact probability is less than the group's contact probability and the agent is in the group at the randomly calculated time of exposure.

4.2.4.4 Recover

This method transitions the disease agent from the infected state to the Recovered state. Once in the recovered state the agent is no longer susceptible to exposure. Although the disease agent can still move between groups, they no longer play a role in the progression of the disease. Even though this project did not take into account agents that can be exposed multiple times, this architecture is set up to support such studies.

4.2.4.5 Move

All disease agents are constantly moving to different groups during the simulation. In order for an agent to move to another group, the desired group must be a neighbor of the current group. The group which the agent moves to is determined at random as well as the travel time of the agent. When an infected agent arrives at a new group they are able to expose susceptible agents in the new group to the disease. This process is exactly the same as when an agent becomes infected for the first time. This enables the disease to spread to different groups.

4.2.5 Group

The group inherits from the simulation object and acts as a base class for application specific groups. The groups in the agent architecture are set up using a tree design; therefore, each group has a reference to its parent and a reference to any children it may have. This elegant design makes execution of events extremely fast since the

number of groups can become exceedingly large depending on the application. Since the move method of the disease agent requires knowledge of neighboring groups, each group also has a list of its neighbors. The final thing a group needs is a list of agents that are in the group. Each of these attributes are independent of the application and are things that any group should have.

When setting up the tree hierarchy, methods such as add child group and add neighbor group are provided. These methods are used by the text file parser during the initialization phase of the simulation. Each group also needs to be able to add agents and remove agents. The final method used by the group is the execute method. Each of these methods is explained in more detail below. Figure 11 shows the UML class diagram for the group.

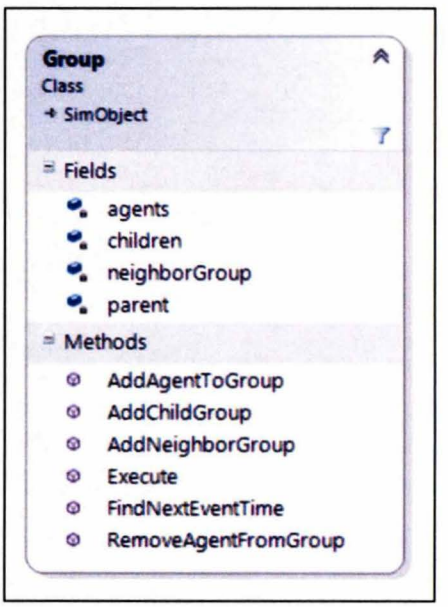

Fig. 11. UML Class Diagram of the Group.

4.2.5.1 Add Agent to Group

Each group has a list of all agents that reside in the group. This list of agents is ordered so that the agent with the lowest event execution time is at the head of the list. The add agent to group method adds the agent to the groups agent list in order and sets the current group of the agent to the group that the agent is added to.

4.2.5.2 Remove Agent from Group

This method is used when the agent needs to be removed from the group. The desired agent is removed from the group's agent list preserving the order of the list.

4.2.5.3 Execute

This is the method used by the simulation executive when executing events. The execute method first checks to see what the next execution time is for the simulation. Once this time is acquired, the root group checks to see if there is a group event that matches the next execution time, and if so it executes this event. Otherwise, the root searches through each of its children until the child group with the next execution time is found.

4.2.6 Disease Group

The disease group has only a few minor differences from the group in this application. The most important attribute of the disease group is the contact probability. This is the probability that an infected agent will come in close contact with a susceptible agent. The contact probability can be different for each group, but in this study it was kept constant across all groups. The main reason for this is that there is no data available that represents contact probabilities. Besides the contact probability, each disease group has a total of four agent sets that represent each of the four disease states. This makes locating the susceptible agents in each group much easier.

The two methods used by the disease group are add agent to set and remove agent from set. These methods make it so that each of the disease agents is in the set that represents its current disease state. These methods are explained next. Figure 12 shows the UML class diagram for the disease group.

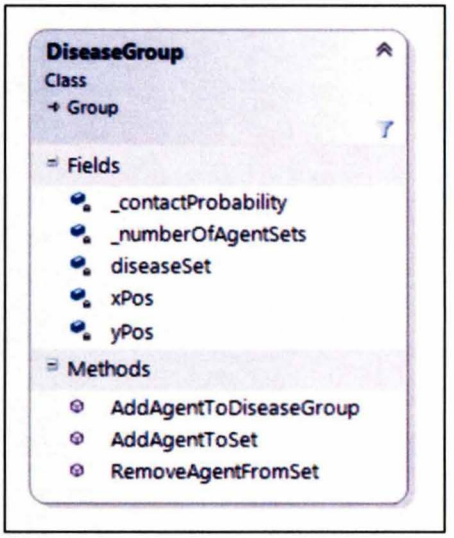

Fig. 12. UML Class Diagram of the Disease Group.

4.2.6.1 Add Agent to Set

This method adds an agent to the specified disease set. Each disease state is represented with an integer where 0 = Susceptible, 1 = Exposed, 2 = Infected, and 3 = Recovered. With these integer values, the disease agent can be added to the appropriate set if the agents current disease state changes.

4.2.6.2 Remove Agent from Set

This method is used when an agent's current disease state is going to change. When the agent's disease state changes due to an event, the agent is removed from its current set. Once this is done, the agent is added to its new set using the add agent to set method previously discussed.

CHAPTER 5

RESULTS

This chapter details the results obtained from the study. All results are produced using the implementation discussed in the previous chapter. This study is unique in that there are no data available that can be used to calculate the probability that an agent will come in close contact with another agent, or contact probability. Although these data would be difficult to gather, there is a need for such data. Hopefully, the results obtained in this study will help to substantiate this need. If the reader is interested in studies that employ unique methods to determine if an agent has come in close contact with another agent please see [18] or [19].

Since there is no contact probability data available, the model in this study is calibrated to match that of another model. The model used for calibration employed the use of the well-known agent-based simulation toolkit Repast Simphony to simulate the spread of a measles epidemic in an urban setting [3]. Measles, otherwise known as rubeola, is a viral illness that spreads through close contact; therefore, this disease is also a perfect candidate for the contact probability model used in this study [3].

5.1 Distributions Used for Measles Disease State Change

Measles is known to have a latency period of approximately eight days and an infectious period of approximately twelve days [3]. These approximations are used as the mode of a triangular distribution in order to make the simulation stochastic. The new periods become:

$$latency\ period \sim Triangular(7, 8, 9) \quad (4)$$

$$infectious\ period \sim Triangular(11, 12, 13) \quad (5)$$

The infect event uses the latency period distribution and the recover event uses the infectious period distribution. In this fashion, these two events occur at random times making it possible for each simulation replication to be unique.

The Expose event is much more difficult to fit to a distribution for a couple of reasons. When using a contact probability, the agents are not moving around freely in the group. Instead, when an agent becomes infectious they use the contact probability to predict whether or not they will come into contact with a susceptible agent. The problem with this is that the time of exposure is unknown even if it is known that there will be an exposure. It is only known that the infectious agent will come into contact with the susceptible agent at some time during the simulation.

An issue is when the exposure will take place. It seems logical that the infectious agent would expose the susceptible agent over the course of a single day. In this fashion, infectious agents are exposing a unique set of susceptible agents each day. This assumption is used to fit the exposure period to a uniform distribution. The exposure period becomes:

$$exposure\ period \sim U(0, 1) \quad (6)$$

Using the uniform distribution a time of exposure is computed between the current simulation time and current simulation time plus 24 hours. This is the final distribution needed for the model.

5.2 Scenarios

In the calibration model, four different scenarios, each with a different number of initially infected agents, are simulated [3]. In order to properly calibrate the contact probability model, the same four scenarios are used. This makes comparison and calibration between the model outputs much easier. An interesting note is the fact that Repast Simphony only allows for the use of 10,000 total agents in the calibration model's simulation, which is a fairly small number when compared to actual city sizes. The contact probability model used in this study is capable of running with 1,000,000 agents, a feat in itself. The initial conditions of the four scenarios are provided in Table 2.

TABLE 2

NUMBER OF INFECTED AGENTS FOR EACH SCENARIO

	Susceptible	Infected
Scenario One	999	1
Scenario Two	990	10
Scenario Three	950	50
Scenario Four	800	200

5.3 Single Group Study

The first set of scenarios is simulated using one large group. This is a logical starting point due to its simplicity, and calibrating the contact probability model is an

accomplishable task. The only parameter value that is altered in order to calibrate the model is the contact probability. After nearly 100 trials, a contact probability of 0.01 was found to produce output from the contact probability model that matches that of the calibration model.

There are two different methods of looking at the simulation output. The first method involves taking a global view of the model. The simulation output should consist of total number of susceptible, exposed, infected, and recovered agents in all the groups. Method two involves looking at the disease state totals for each group individually. When dealing with a single group, the two methods are one and the same; however, when there is more than one group it is beneficial to observe how the disease spreads between the groups. The simulation output provided by the calibration model deals with global values; therefore, these values are most beneficial when calibrating.

5.3.1 Scenario One

Using the first method, the total number of agents in each disease state is calculated every five hours during the 750 hour simulation runtime. This output is plotted as a multi-line graph for each of the four scenarios using the calibrated contact probability of 0.01. The output results from the contact probability model show similar trends to that of the calibration model. Figure 13 depicts the output from scenario one with the contact probability model on top and the calibration model underneath. The number of the susceptible agents is very similar between the two plots and so is the number of exposed agents; however, there is a dip at the end of the contact probability

output for exposed agents that is not easily removed. Other than that, the two outputs from scenario one are very similar.

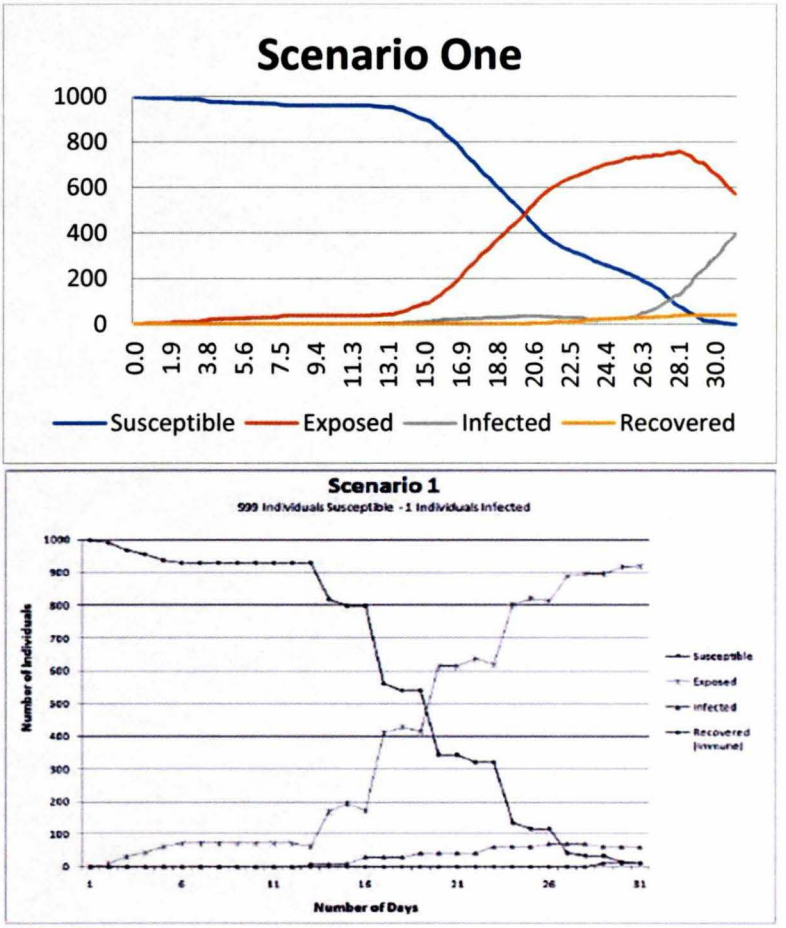

Fig. 13. Results from Scenario 1 with the Contact Probability Model on Top and the Calibration Model Underneath.

5.3.2 Scenario Two

Scenario two starts out with ten infectious agents and a contact probability of 0.01, therefore, the rate of exposure increases when compared to scenario one. Figure 14 shows the outputs for scenario two. The two outputs for this scenario are also very

similar. The susceptible agent curve is almost identical between the two outputs, but the exposed agent curve shows some differences. Even with the differences, the contact probability output exhibits a peak in the exposed agent curve approximately halfway through the simulation as well as a dip in the infected agent curve just prior to the end, both of which are characteristics of the calibration model.

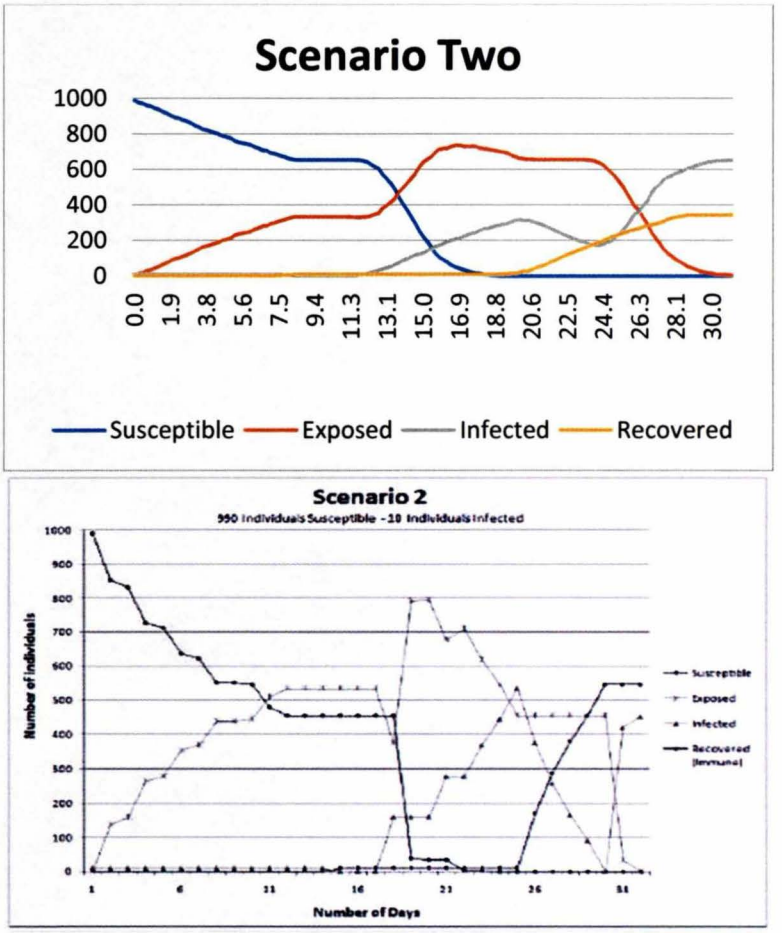

Fig. 14. Results from Scenario 2 with the Contact Probability Model on Top and the Calibration Model Underneath.

5.3.3 Scenario Three

Figure 15 shows the outputs for scenario three, which has fifty initially infected agents and a contact probability of 0.01. The outputs from this scenario show some minor differences between the two models. For example, the exposed agent curve starts to decrease in the contact probability output much too soon. It appears that the contact probability output is sped up when compared to the calibration model output. The infected agent curve also does not match up very well between the two models. Even with these differences, the shapes of the curves are very similar between the two models.

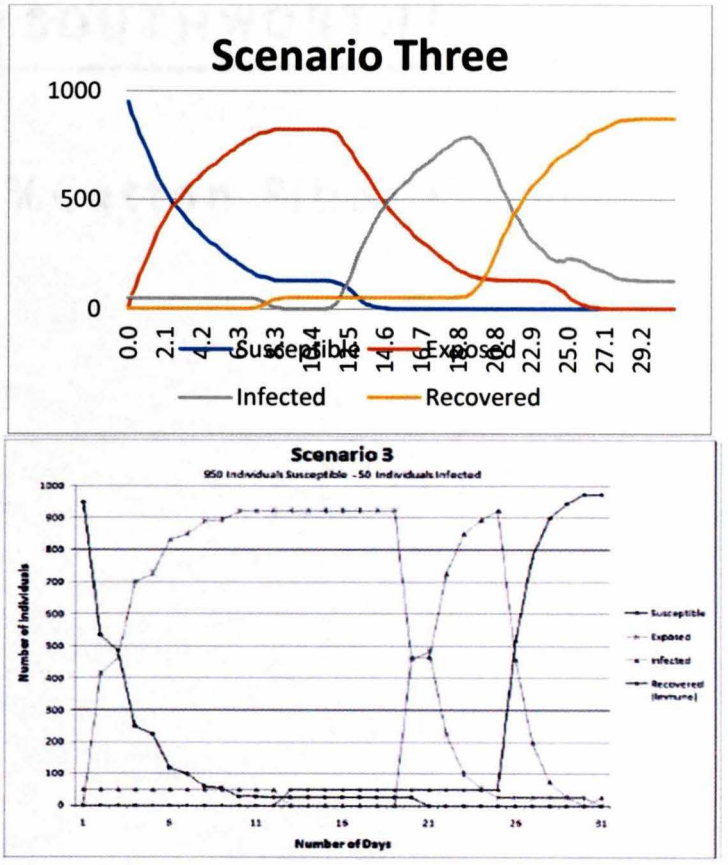

Fig. 15. Results from Scenario Three with the Contact Probability Model on Top and the Calibration Model Underneath.

5.3.4 Scenario Four

The final scenario in this study starts out with two hundred initially infected agents and has a contact probability of 0.01. Figure 16 shows the output of the two models for this scenario. There are some noticeable differences between the two plots. The exposed agent curve starts to decrease much too soon and the infected agent curve starts to rise too soon as well. The susceptible agent curve and the recovered agent curve are very similar between the two models.

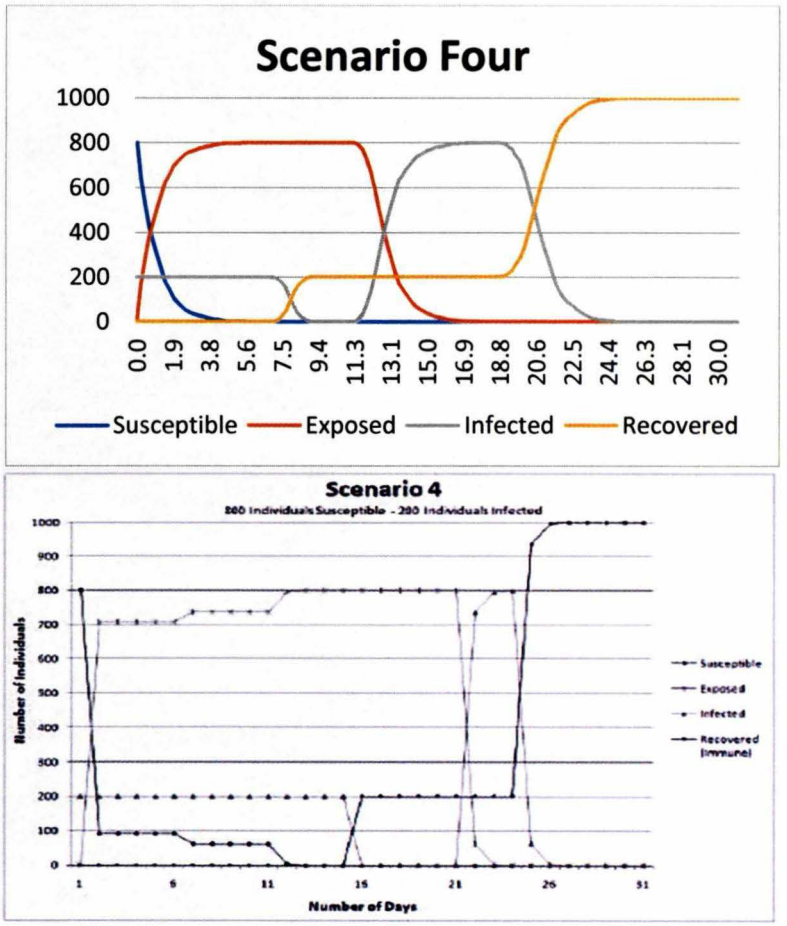

Fig. 16. Results from Scenario 4 with the Contact Probability Model on Top and the Calibration Model Underneath.

5.4 Multiple Group Model without Movement

For the second stage of this study multiple groups are used. The groups can be thought of conceptually as a grid with the agents distributed evenly between the groups. Each group in the grid can be considered a leaf group that has a parent group which it shares with three other groups. The assigning of parent groups is continued with each parent recursively until the parent group has no neighbors. This final group without neighbors is known as the root group. This tree structure makes it so that the environment can be thought of as a single root group, which has access to all of its children.

Since there are more groups, how agents are exposed to the disease is modified. Rather than agents only being able to expose susceptible agents in the same group, the infected agents are now able to expose susceptible agents in neighboring groups. There is still no movement of the agents; however, this is a logical second step towards a more realistic model when there is more than a single group. A group is considered a neighbor of another group if it is in the same row, same column, or diagonal in the grid space.

For this study, the local output of the groups is also desirable so an after-action viewer is implemented using OpenGL. This viewer takes the output from the contact probability model and uses it to produce an animation of the disease state changes that are taking place in each group. Figure 17 depicts the visualization of the after-action viewer for a grid size of 8 X 8 with the progression of disease states in each group shown on the left and agent location shown on the right. The yellow color stands for susceptible, the blue color stands for exposed, the red color stands for infected, and the green color stands for recovered. This viewer works for any grid size; however, as the size increases the output is harder to visualize.

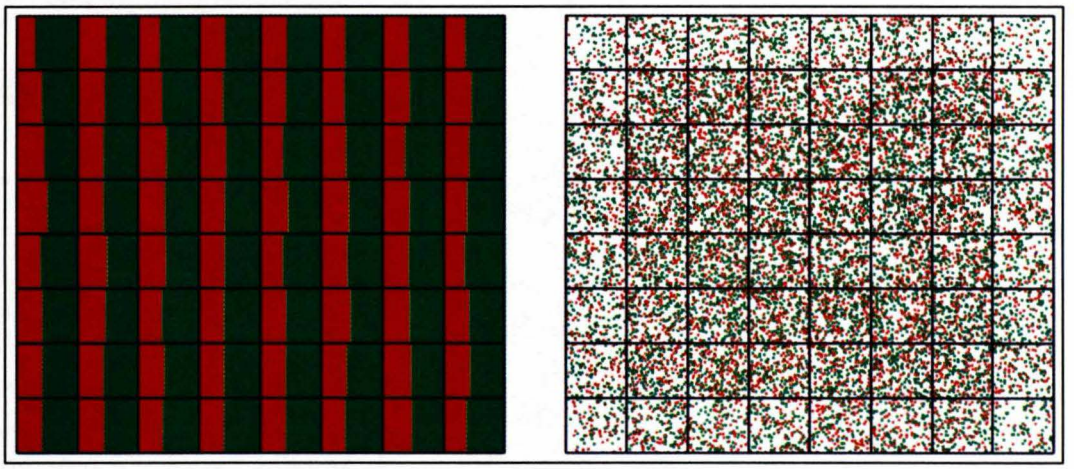

Fig. 17. After-action viewer snapshot during the 8 X 8 grid run.

After nearly one hundred trials of this study, the contact probability is calibrated to 0.04. Not only is the output of each scenario compared, but the output between the different grid sizes is also compared. In this fashion, a contact probability that maintains similar output between the different grid sizes is found. The grid sizes are increased by multiples of two. For example, the first grid size is 2 X 2, the second grid size is 4 X 4, and the third grid size is 8 X 8. Each grid size is used to run the four different scenarios and the grid size is increased to 64 X 64. The number of agents is scaled as well using multiples of four for each grid size. For example, the 2 X 2 grid had 1,000 agents and the 4 X 4 grid had 4,000 agents. This scaling enabled the final grid size to have just over 1,000,000 agents. The number of initially infected agents is scaled proportionally for each grid size. In this way, six different trials are conducted with each trial having the four core scenarios.

5.4.1 Grid Size 2 X 2

Each of the four scenarios is run using a grid size of 2 X 2 and a contact probability of 0.04. There are a total of 1,000 agents for this trial. This grid size has very similar results to that of the first stage in the study which uses a single group. The output for each scenario is shown in Figure 18. These plots are very similar to those of the previous stage, therefore, they are a good fit to the calibration model.

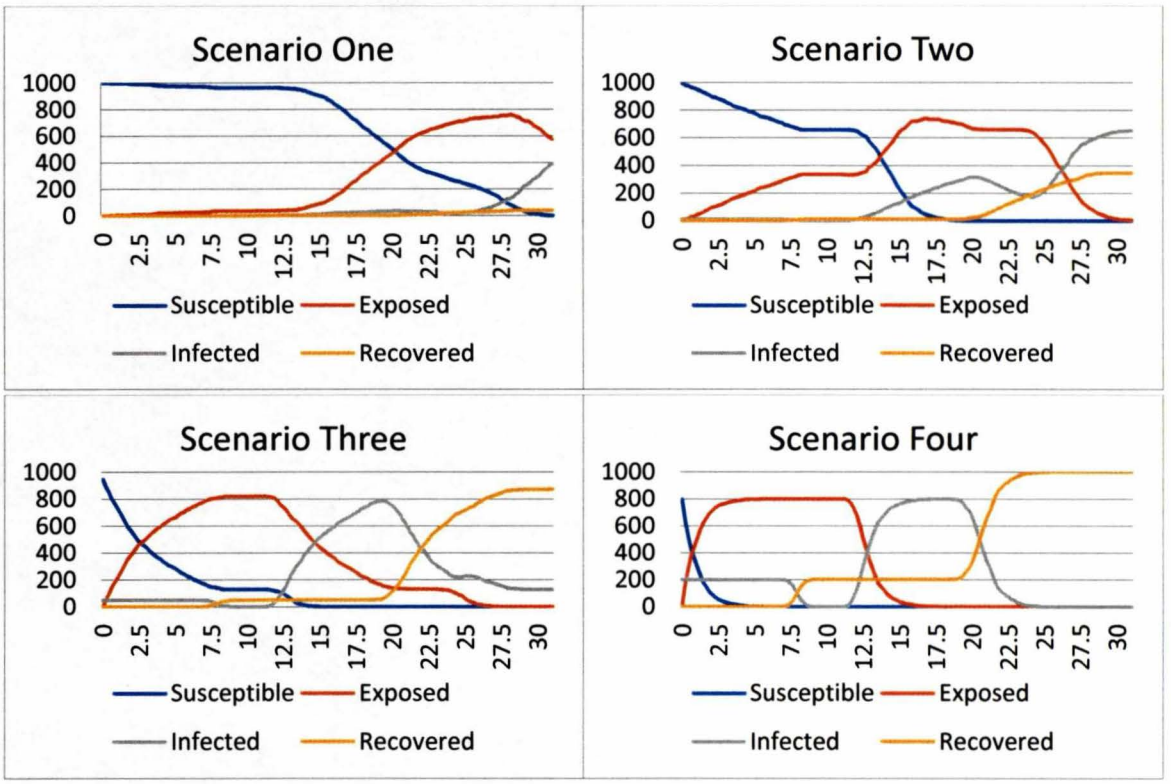

Fig. 18. Output for all Four Scenarios using the 2 X 2 Grid without Movement.

5.4.2 Grid Size 4 X 4

All four scenarios are run using a grid size of 4 X 4 and a contact probability of 0.04. The total number of agents for this grid size is 4,000. Figure 19 depicts the results

obtained from the four scenarios. There are some noticeable differences in the outputs of the first two scenarios. For example, the peak of the exposed agent curve happens around 27.5 days in the previous grid size and 25 days for this grid size. The results from the second scenario are becoming closer to the results of the calibration model. The exposed agent curve of scenario two stays constant at fifty percent from day 7.5 to day 12.5 and then rises to a peak of seventy-five percent at day 15 followed by a decrease back to fifty percent from day 20 to day 25. The last two scenarios show little change from the previous.

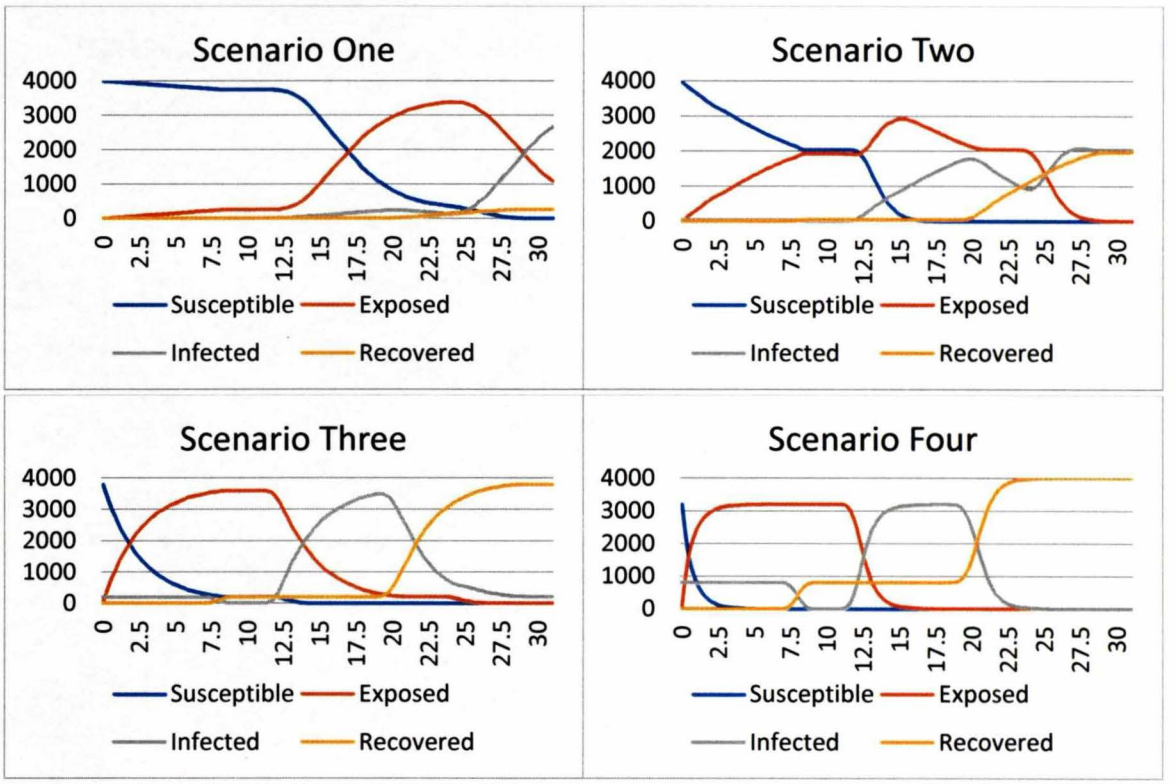

Fig. 19. Output for all Four Scenarios using the 4 X 4 Grid without Movement.

5.4.3 Grid Size 8 X 8

Using a grid size of 8 X 8 and a contact probability of 0.04, results are obtained for each of the four scenarios. There are 16,000 agents interacting for this grid size. The plots for each of the four scenarios are shown in Figure 20. The results from the first scenario show little change from the previous. In scenario two, the susceptible agent curve stays constant right below fifty percent from day 7.5 to day 12.5 and the exposed agent curve stays constant right above fifty percent over the same range. This change from the previous brings the results from scenario two closer to the calibration model's results. There is little change in the outputs of scenario three or scenario four.

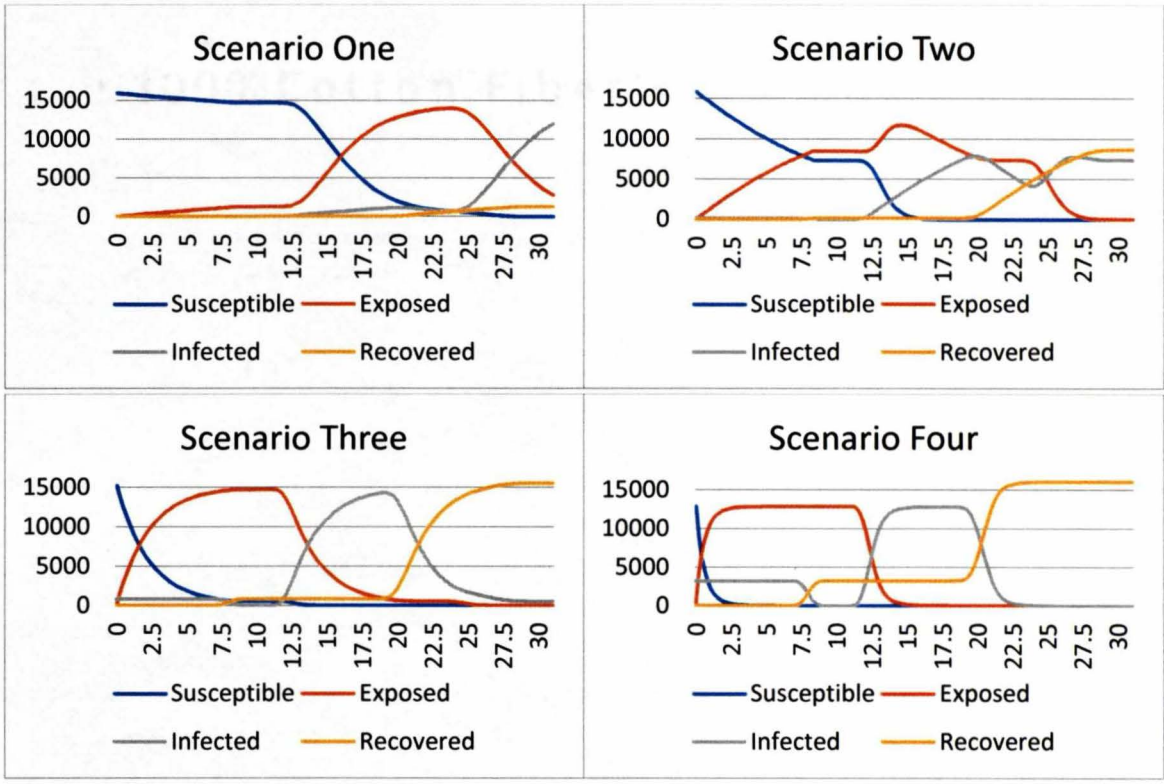

Fig. 20. Output for all Four Scenarios using the 8 X 8 Grid without Movement.

5.4.4 Grid Size 16 X 16

Each of the four scenarios are run using a grid size of 16 X 16 and a contact probability of 0.04. The total number of agents is 64,000, and the outputs for each of the scenarios are shown in Figure 21. The most noticeable difference for this grid size appears in the output for scenario two. The peak of the infected agent curve takes place around day 20 and rises just above the exposed agent curve. This same characteristic appears in the calibration model's output for scenario two. There is no noticeable change in the outputs of the other three scenarios.

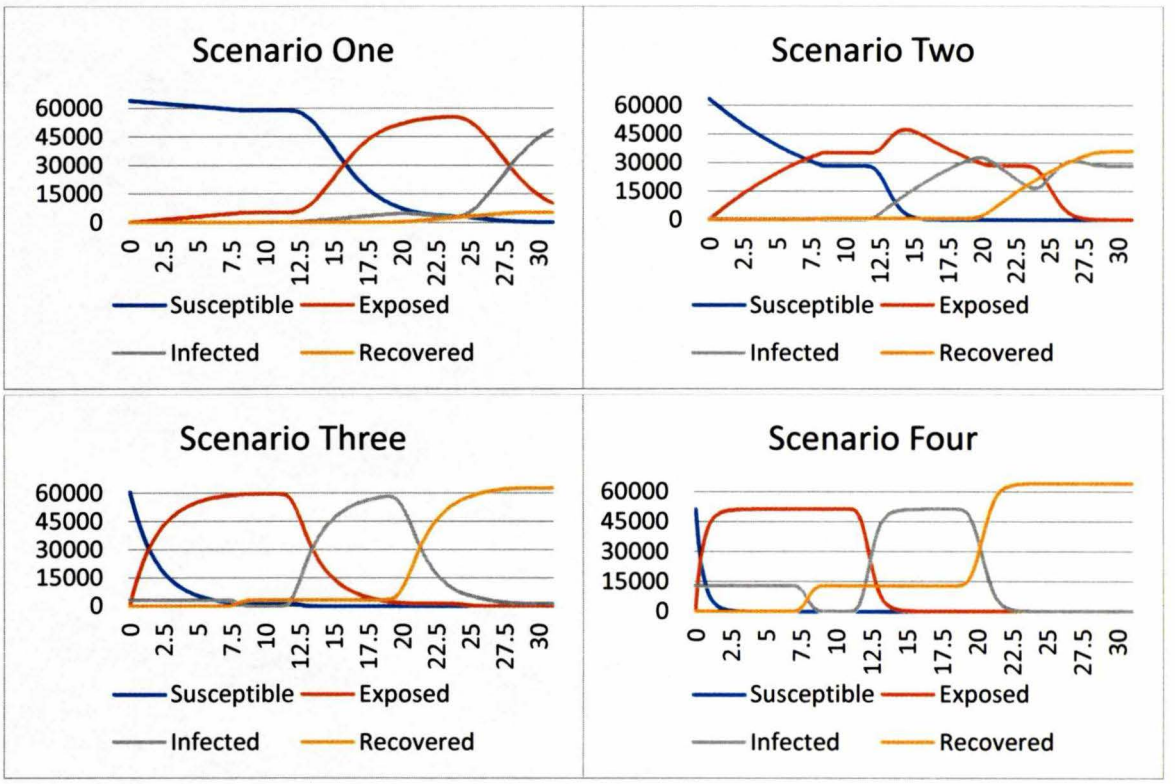

Fig. 21. Output for all Four Scenarios using the 16 X 16 Grid without Movement.

5.4.5 Grid Size 32 X 32

The grid size is increased to 32 X 32 for this trial and all four scenarios are run using a contact probability of 0.04. There are 25,6000 agents populating all of the groups. The outputs for each of the scenarios are shown in Figure 22. It appears that the second scenario provides the perfect number of initially infected agents to appreciate noticeable changes in the model's output. There are two key changes in scenario two's output. First, the gap between the constant period of susceptible and exposed agent curves that takes place between days 7.5 and 12.5 has grown larger more closely matching that of the calibration model. Second, the peak of the infected agent curve at day 20 has risen higher, which also is a better fit to the calibration model. The outputs from the other three scenarios show little change from previous.

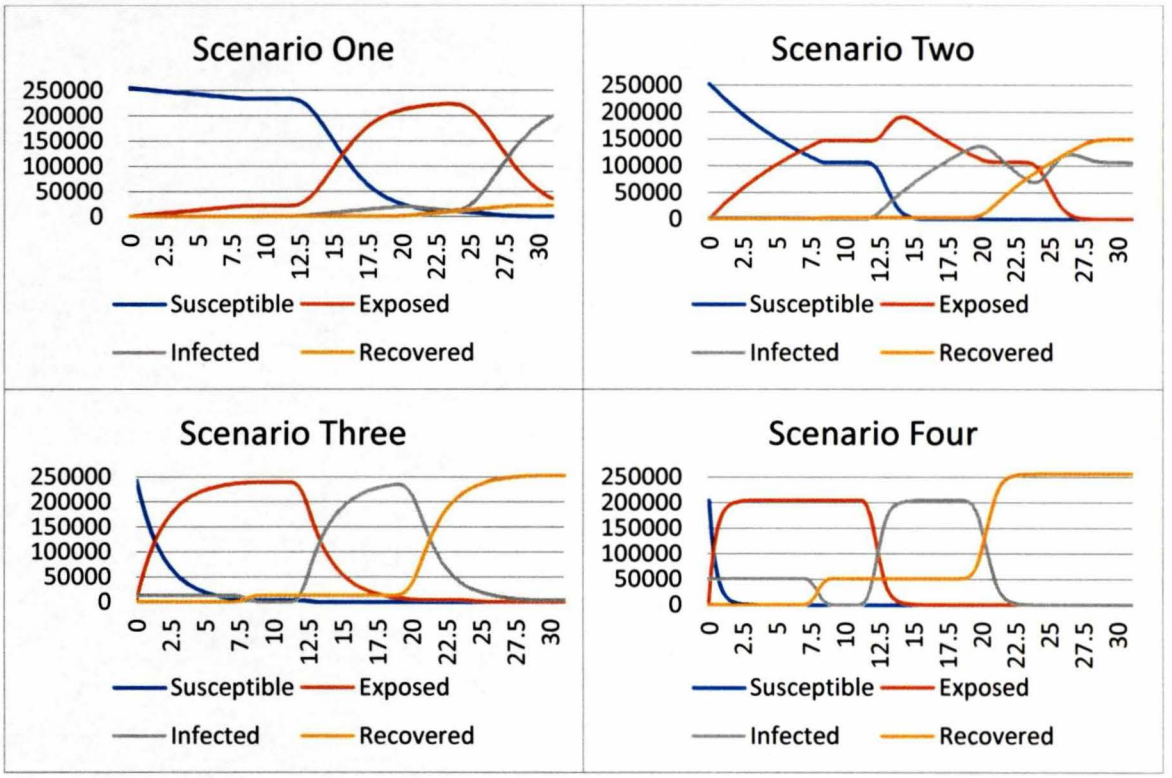

Fig. 22. Output for all Four Scenarios using the 32 X 32 Grid without Movement.

5.4.6 Grid Size 64 X 64

For this final trial in the multi group stage a grid size of 64 X 64 is used with a contact probability of 0.04. There are a total of 1,024,000 agents in this trial. Only the first scenario is run using this grid size mainly for proof that the simulation architecture is capable of handling a large number of agents. The output for scenario one is shown in Figure 23. As can be seen in the output, there is very little change from the previous trials. This demonstrates that a given contact probability is scalable not only to different group sizes, but also to different agent population sizes.

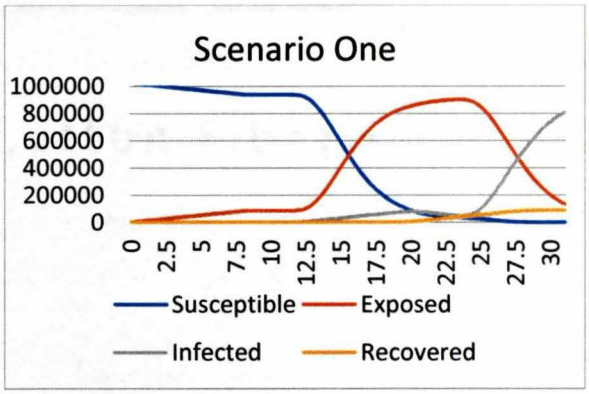

Fig. 23. Output for the First Scenario using a 64 X 64 Grid and Running 1,024,000 Agents.

5.5 Multiple Group Model with Movement

The final stage in this study adds movement to the agents. In order to do this, movement of agents first needs to be defined. When dealing with contact probabilities, it does not matter where the infected agent is in relation to susceptible agents in a given group. The contact probability is used to predict what susceptible agents the infected agent will come in close contact with. For this reason, it does not make sense to add movement of an agent within a group to this study. Rather, it is better to focus on the

movement of agents between groups. In this fashion, when an infected agent moves to a new group there exists a new population of susceptible agents that can be exposed to the disease. It does not matter where these agents are in the group. All that matters is that these agents are in the same group. The contact probability takes care of the rest.

The next question is, what groups can an agent move to? Upon examination of the grid space, it is logical that an agent can move to a group that is adjacent in either the same column or same row. In order to move to a group that is diagonal the agent would have to first go through one of these adjacent groups, so the diagonal groups are removed from the equation. The assumption is made that an agent has an equal probability of moving to any of the adjacent neighboring groups. Of course, there could exist situations where one group would have a higher probability of travel, but these situations are not taken into account for this study.

The only other thing that must be taken into account here is the time of exposure, or the distribution that is a good fit for the exposure time. When an infected agent travels to a new group the agent is ready to expose susceptible agents to the disease. There are two things that must be true in order for the exposure to take place. The first requirement is that a generated random number is less than the contact probability of the group. Second, the time of exposure must take place when both the infectious agent and the susceptible agent are both in the group at the same time. For example, what if the exposure time for a susceptible agent is two hours in the future, but the susceptible agent will be leaving the group in one hour? This would make it so that the exposure would not take place.

In order to account for this second requirement, a new method of determining the exposure period's distribution needs to be used. This distribution is determined by the time the infected agent is going to be in the new group. There was also an assumption made that the probability of a susceptible agent becoming exposed to the disease is uniformly distributed. Putting all this together, when an infected agent travels to a new group, the agent first must calculate randomly when it will move to the next group. The agent then uses the current simulation time for the minimum of a uniform distribution and the current simulation time plus the calculated departure time as the maximum of a uniform distribution. If the generated exposure time is earlier than the departure time of the susceptible agent, then the susceptible agent will be exposed to the disease at said time.

Before running the trials, the contact probability is calibrated the same way as has been previously done. After performing the first round of calibration, it is found that the contact probability does not need to be changed from the previous value of 0.04. In order to reduce redundancy, there are only three trials for this stage of the study. The first trial starts with the 4 X 4 grid size, and the remaining two trials use the 8 X 8 grid size and the 16 X 16 grid size. All four scenarios are run using these grid sizes. Looking at the previous stage shows that the majority of noticeable changes take place using the three grid sizes previously mentioned.

5.5.1 Grid Size 4 X 4 with Movement

For the first trial with movement, a 4 X 4 grid size is used with a contact probability of 0.04. Figure 24 depicts the results from each of the four scenarios. The

total number of agents used in this trial is 4,000. For the first scenario's output, both the susceptible agent curve and the exposed agent curve decrease at a faster rate when compared to the same grid size's output with no movement. The output from scenario two shows that there is already some overlap between the constant sections of the susceptible agent curve and the exposed agent curve that take place between day 7.5 and day 12.5. This is a characteristic that appears in the calibration model's output. There is very little change in the outputs of scenario three and scenario four when compared to the same scenarios in the previous stage of this study.

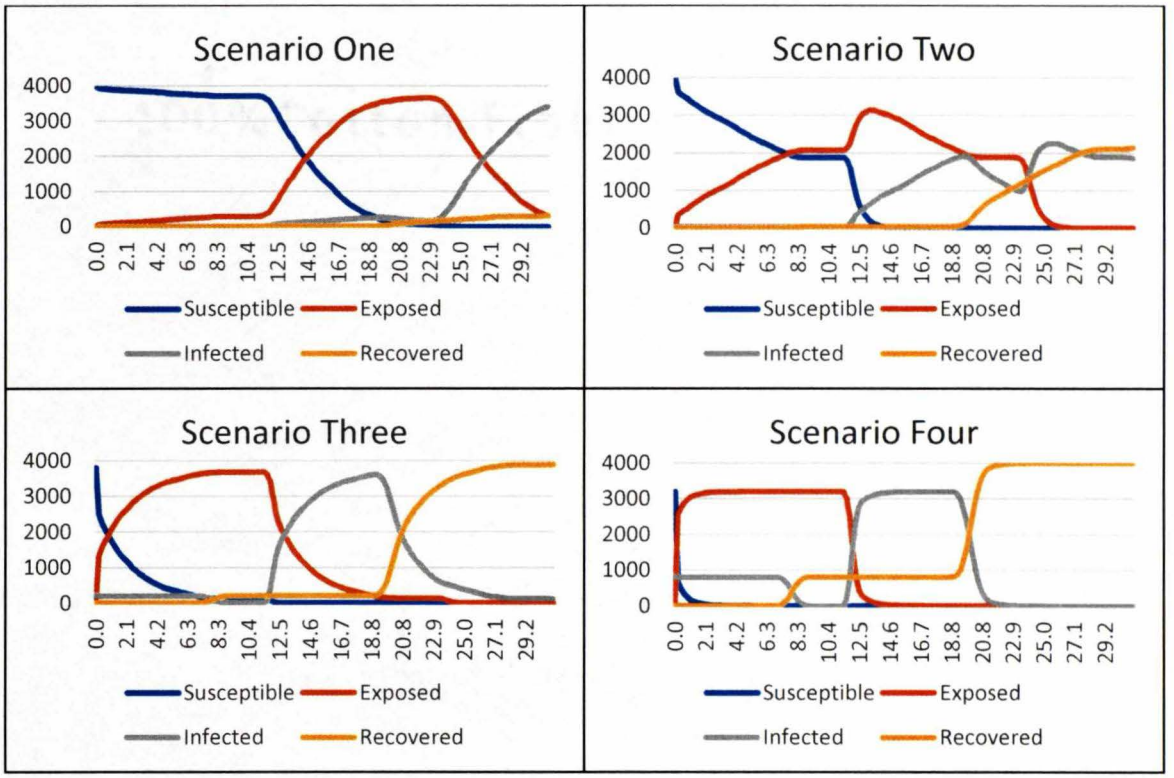

Fig. 24. Output for all Four Scenarios using a 4 X 4 Grid with Movement.

5.5.2 Grid Size 8 X 8 with Movement

Each of the four scenarios is run using a grid size of 8 X 8 and a contact probability of 0.04. There are 16,000 total agents in this trial. The output for the scenarios is shown in Figure 25. In the first scenario's output, the point at which the susceptible agent curve and the exposed agent curve start to decrease is much more defined when compared to the previous grid size and the previous stage of this study. This same characteristic appears in the output for scenario three. There is no noticeable change in the outputs of scenario two or scenario four for this trial.

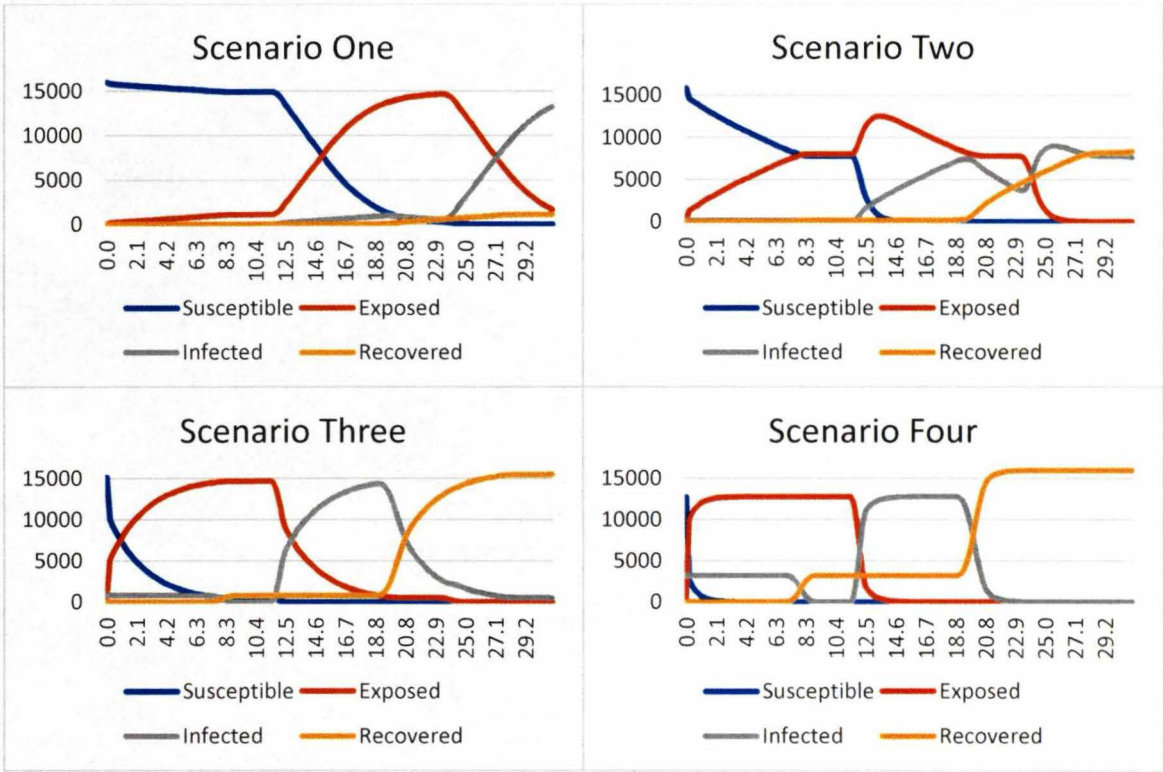

Fig. 25. Output for all Four Scenarios using an 8 X 8 Grid with Movement.

5.5.3 Grid Size 16 X 16 with Movement

The final grid size used for this trial is 16 X 16 with a contact probability of 0.04. The total number of agents used in this trial is 64,000. Figure 26 shows the outputs of each scenario. There is very little change in the plots of scenario one, scenario three, and scenario four. The only noticeable change that takes place in scenario two is the lowering of the first peak of the infected agent curve. This peak is no longer overlapping the exposed agent curve. This change acutally makes the output move away from the calibration model's output.

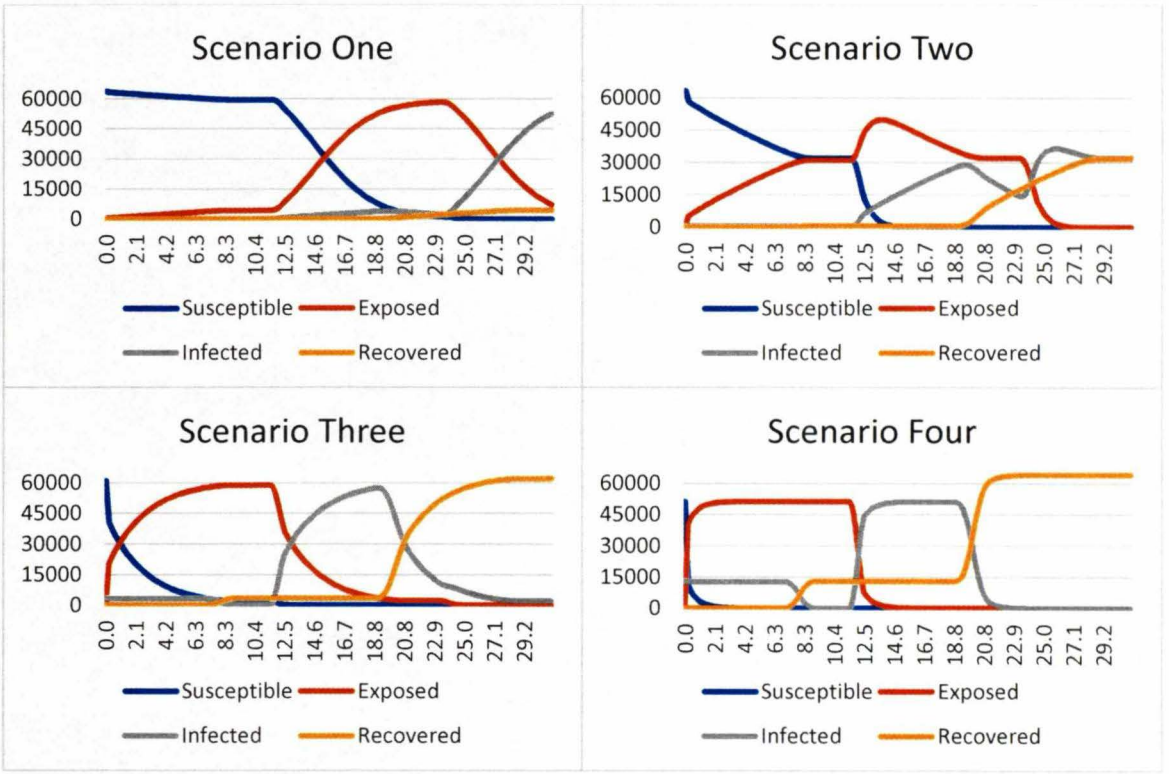

Fig. 26. Output for all Four Scenarios using a 16 X16 Grid with Movement.

5.6 Computation Timings for the Model with Movement

Since the model with movement is more complex than the others, the computation time of the simulation is recorded for the first four grid sizes. The timing results are shown below in Table 3. For the 2 X 2 grid size with 1,000 agents, the total computation time is 1.33 seconds. When the grid size is increased to 4 X 4 with 4,000 agents, the total computation time increases to 7.31 seconds. The computation time for the 8 X 8 grid size with 16,000 agents is 52.21 seconds. For the final grid size of 16 X 16 with 64,000 agents, the computation time is 8.15 minutes. This demonstrates that for the most complex model used in this study, the total number of agents can be much larger than 10,000, which is the limit in Repast Simphony. These grid sizes were chosen because they had the best computational timings.

The scalability of the model is provided by the number of levels in the tree. For example, the 2 X 2 grid size only has two levels in the tree. When the number of levels is increased to three the grid size becomes 4 X 4. This trend continues until the 64 X 64 grid size is reached, which uses seven levels in the three. The number of levels used depends on the number of groups being studies as well as the number of agents being simulated.

TABLE 3

COMPUTATION TIMINGS FOR MODEL WITH MOVEMENT

	1000 Agents	4000 Agents	16000 Agents	64000 Agents
Timing	1.33 seconds	7.31 seconds	52.21 seconds	8.15 minutes

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

This study has demonstrated that the use of contact probability in an agent-based infectious disease model is a plausible approach, demonstrating appropriate behavior with improved computational performance. Each stage of the study was easily calibrated so that the output matched that of the calibration model. There is very little change in the output of the model without movement and the model with movement.

The performance of the model was exceptional. The grid design of the groups made scaling very easy. As additional levels were added to the tree the number of groups was increased and the number of agents was increased. The timings shown in Section 5.6 demonstrate that this model can handle 64,000 agents in a reasonable amount of time.

There are several areas in which this study could be expanded upon. The first, and most interesting, would be to divide the disease agents into different groups such as children, adolescents, adults, and elderly. Each subgroup of disease agents could then have their own probability of contracting a disease. This probability could then be used in conjunction with the contact probability in each group to determine if an exposure takes place. For example, the elderly subgroup would most likely have an increased chance of becoming exposed to a disease due to a weaker immune system as opposed to a healthy adult with a stronger immune system.

In addition, an additional stage could be added to the study. This stage could be the mixing of moving between groups to infect agents and infecting agents in neighboring groups. For example, when an infected agent moves to a neighboring group they would be able to expose susceptible agents in that same group as well as susceptible agents in neighboring groups. Of course, there would be a different contact probability

for exposing agents in the same group vs. exposing agents in a neighboring group. This simulation could give an interesting blend between the final two stages used in this study.

In the current model, if a contact is predicted between an infected agent and a susceptible agent, then the susceptible agent is exposed to the disease. This is not what really happens, so it would be desirable to add some more realistic features to the model and see what the results would be.

Another area of expansion would be to vary the contact probabilities between the groups. There are some groups that would have a higher probability of contact then others. The model is set up to handle this; however, there is no available data on contact frequencies in different settings, therefore, this area would most likely need a subject matter expert's input on social interaction in order to come up with an estimation of the different contact probabilities.

This study has provided a robust framework that is very easy to build upon. The results have shown that using contact probability is a valid method for modeling infectious disease spread. Hopefully, this study is expanded upon at a future date once there is some available data on the frequency of contact between individuals in different settings.

REFERENCES

[1] Y. Yang, P. M. Atkinson, and D. Ettema, "Analysis of CDC social control measures using an agent-based simulation of an influenza epidemic in a city," *BMC Infectious Diseases, 11,* p. 199, 2011.

[2] N. Gilbert, *Agent-based models* vol. 153: Sage, 2008.

[3] L. Perez and S. Dragicevic, "An agent-based approach for modeling dynamics of contagious disease spread," *International Journal of Health Geographics, 8,* p. 50, 2009.

[4] H. Rahmandad and J. Sterman, "Heterogeneity and network structure in the dynamics of diffusion: comparing agent-based and differential equation models," *Management Science,* vol. 54, pp. 998-1014, 2008.

[5] M. Ajelli, B. Gonçalves, D. Balcan, V. Colizza, H. Hu, J. J. Ramasco, *et al.*, "Comparing large-scale computational approaches to epidemic modeling: Agent-based versus structured metapopulation models," *BMC Infectious Diseases, 10,* p. 190, 2010.

[6] F. Klüg and A. L. C. Bazzan, "Agent-based modeling and simulation," *Artificial Intelligence Magazine,* pp. 29-40, 2012.

[7] H. Jiang, W. Karwowski, and T. Z. Ahram, "Applications of agent-based simulation for human socio-cultural behavior modeling," *Work, 41,* pp. 2274-2278, 2012.

[8] A. L. Bauer, C. A. A. Beauchemin, and A. S. Perelson, "Agent-based modeling of host–pathogen systems: The successes and challenges," *Inf Sci (Ny), 179(10),* pp. 1379-1389, 2009.

[9] C. I. Siettos and L. Russo, "Mathematical modeling of infectious disease dynamics," *Virulence, 4(4),* pp. 295-306, 2013.

[10] J. Stehlé, N. Voirin, A. Barrat, C. Cattuto, V. Colizza, L. Isella, *et al.*, "Simulation of an SEIR infectious disease model on the dynamic contact network of conference attendees," *BMC Medicine, 9,* p. 87, 2011.

[11] E. Vynnycky and R. White, *An introduction to infectious disease modelling*. New York: Oxford University Press, 2010.

[12] K. M. Khalil, M. Abdel-Aziz, T. T. Nazmy, and A. M. Salem, "An agent-based modeling for pandemic influenza in egypt," Faculty of Computer and Information Science Ain shams University Cairo, Cairo2010.

[13] S. Funk, M. Salathé, and V. A. A. Jansen, "Modelling the influence of human behaviour on the spread of infectious diseases: a review," *J. R. Soc. Interface, 7(50),* pp. 1247-1256, 2010.

[14] L. Bian, "Spatial approaches to modeling dispersion of communicable diseases – a review," *Transactions in GIS, 17(1),* pp. 1-17, 2013.

[15] S. Meloni, N. Perra, A. Arenas, S. Gómez, Y. Moreno, and A. Vespignani, "Modeling human mobility responses to the large-scale spreading of infectious diseases," *Scientific reports, 1,* 2011.

[16] S. C. Patel and G. Bhatt, "Simulating the transmission of epidemics through complex systems: A classical model," *Global Conference on Business and Finance Proceedings, 9(1),* pp. 125-131, 2014.

[17] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of reusable object-oriented software*: Addison-Wesley, 1995.

[18] A. Machens, F. Gesualdo, C. Rizzo, A. Tozzi, A. Barrat, and C. Cattuto, "An infectious disease model on empirical networks of human contact," *Machens et al. BMC Infectious Diseases,* vol. 13, p. 16, 2013.

[19] T. Hornbeck, D. Naylor, A. M. Segre, G. Thomas, T. Herman, and P. M. Polgreen, "Using sensor networks to study the effect of peripatetic healthcare workers on the spread of hospital-associated infections," *J Infect Dis,* vol. 206, pp. 1549-57, Nov 15 2012.

[20] S. Eubank, H. Guclu, V. Kumar, M. Marathe, A. Srinivasan, Z. Toroczkai, *et al.*, "Modeling disease outbreaks in realistic urban social networks.," *Nature,* vol. 429, pp. 180-184, 2004.

[21] C. Barrett, K. R. Bisset, S. Eubank, X. Feng, and M. Marathe, "EpiSimdemics: An efficient algorithm for simulating the spread of infectious disease over large realistic social networks.," in *ACM/IEEE conference on Supercomputing*, 2008, p. 37.

[22] J. Wallinga, P. Teunis, and M. Kretzschmar, "Using data on social contacts to estimate age-specific transmission parameters for respiratory-spread infectious agents.," *American Journal of Epidemiology,* vol. 164, pp. 936-944, 2006.

[23] J. B. Dunham, "An agent-based spatially explicit epidemiological model in MASON.," *Journal of Artificial Societies and Social Simulation,* vol. 9, 2005.

[24] D. Balcan, V. Colizza, B. Gonçalves, H. Hu, J. J. Ramasco, and A. Vespignani, "Multiscale mobility networks and the spatial spreading of infectious diseases.," *Preceedings of the National Academy of Sciences,* vol. 106, pp. 21484-21489, 2009.

[25] J. M. Epstein, "Modelling to contain pandemics.," *Nature,* vol. 460, p. 687, 2009.

[26] W. J. Edmunds, G. Kafatos, J. Wallinga, and J. R. Mossong, "Mixing patterns and the spread of close-contact infectious diseases.," *Emerging Themes in Epidemiology,* vol. 3, p. 10, 2006.

[27] N. C. Grassly and C. Fraser, "Mathematical models of infectious disease transmission.," *Nature Reviews Microbiology,* vol. 6, pp. 477-487, 2008.

[28] M. Salathé, M. Kazandjieva, J. W. Lee, P. Levis, M. W. Feldman, and J. H. Jones, "A high-resolution human contact network for infectious disease transmission.," *Proceedings of the National Academy of Sciences,* vol. 107, pp. 22020-22025, 2010.

[29] V. Grimm, U. Berger, F. Bastiansen, S. Eliassen, V. Ginot, J. Giske, *et al.*, "A standard protocol for describing individual-based and agent-based models.," *Ecological Modelling,* vol. 198, pp. 115-126, 2006.

[30] N. R. Jennings, "On agent-based software engineering.," *Artificial Intelligence* vol. 117, pp. 277-296, 2000.

VITA

Department of Study

Modeling, Simulation & Visualization Engineering

1300 Engineering & Computational Sciences Building

Norfolk, VA 23529

Education

Bachelor of Science in Modeling and Simulation Engineering

Awarded: May 2013

Old Dominion University, Norfolk, VA

GPA: 3.99

Related courses: Discrete Event Simulation, Continuous Simulation, Systems Modeling, Analysis for Modeling and Simulation, Simulation Software Design, Visualization, Distributed Simulation, Biochemistry, Medical Image Coding and Compression

Awards: Outstanding College Scholar for Frank Batten College of Engineering and Technology

Related Experience

Old Dominion University, Norfolk, VA

Research in Organic Chemistry (August 2010-August 2012)

- Researched store-operator calcium inhibition of 2-Aminoethoxydiphenyl borate
- Became familiar with Gaussian 2009 and NW Chem software

Research in Adoptive Immunotherapy (November 2011-April 2012)

- Researched patient thresholds during Interleukin-2 adoptive immunotherapy treatment for metastatic melanoma
- Developed a continuous simulation using Ptolemy II from three ordinary differential equations in order to simulate the effects different levels of treatment entailed
- Recognized as "Best Presenter" in the Medical Modeling and Simulation track at the 2012 Student Capstone Conference