May 2015

An Approach To Artificial Society Generation For Video Games

Bryan Sarlo
The University of Western Ontario

Supervisor
Dr. Michael J. Katchabaw
The University of Western Ontario

Graduate Program in Computer Science

A thesis submitted in partial fulfillment of the requirements for the degree in Master of Science

AN APPROACH TO ARTIFICIAL SOCIETY GENERATION FOR VIDEO GAMES

(Thesis format: Monograph)

by

Bryan Sarlo

Graduate Program in Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Masters of Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Abstract

Since their inception in the 1940s, video games have always had a need for non-player characters (NPCs) driven by some form of artificial intelligence (AI). More recently, researchers and developers have attempted to create believable, or human-like, agents by modeling them after humans by borrowing concepts from the social sciences. This thesis explores an approach to generating a society of such believable agents with human-like attributes and social connections. This approach allows agents to form various kinds of relationships with other agents in the society, and even provides an introductory form of shared or influenced attributes based on their spouse or parents. Our proposed method is a simplified system for generating a society, but shows great potential for future work. As a modularized and parameterized framework, there are many opportunities for adding new layers to the system to improve the realism of the generated society.

Keywords

Video games, artificial societies, simulation, generation, believable agents, social network, population, artificial intelligence, non-player characters, society models

Acknowledgments

I want to thank my supervisor, Dr. Michael Katchabaw, for his creativity in this thesis idea and his immense help and guidance throughout the design, implementation, and the writing of this work.

Table of Contents

List of Tables

List of Figures

Chapter 1

1 Introduction

Video games are increasing in realism in many ways from photorealistic worlds and characters to intelligent behavioural game bots. The level of realistic artificial intelligence (AI) has also come a long way over the past years and new research is continually adding new layers of human-like behaviour to video game agents [1, 2]. Researchers and developers alike have been creating game entities with human attributes and behaviours [3]. In simple cases, agents may contain a one-dimensional personality or identity, as in *Grand Theft Auto 3* in which the non-playing characters (NPCs) are classified with a certain label which will directly govern how they react to specific stimuli. More complex systems provide NPCs with a large set of independent traits that then influence their actions and behaviours [4]. Game agents have also been modeled with emotions, memories, and behaviours that are affected by their personality and emotions as in *The Sims 4*. Several researchers have also explored the realm of believable behaviour systems for creating autonomous agents [5, 6]. Ideally, as NPCs are given more complex traits, emotions, and other human attributes, we should be able to observe common human issues in those agents. For example, complex agents may deal with conflicting thoughts in which their various traits each point to differing plans of action, resulting in a need to weigh out the options from a moral standpoint to decide how to resolve the conflict.

Currently, many games that contain NPCs will just script in those necessary agents, who are typically limited in intelligence and human-like behaviour. Even in games that boast of great AI, the characters are given the ability to play the role necessary in that game [7], but no other skills or personality. For example, AI in a first-person shooter (FPS) would be given the ability to shoot at their enemies, to hide and duck when shots are fired at them, and maybe even to learn which areas yield a better success for them to shoot others without getting shot [8]. Nonetheless, these agents are very specialized in their abilities and typically do not express human-like emotions or traits, and are even less likely to have a network of family members, friends, and other peers.

One area of video game AI research that is largely untapped is that of artificial society generation for a game population, and particularly that on a micro-level with many individualistic human-like agents comprising the population. The term “society” is used frequently throughout this thesis, so we will define the term now. A society has two components: a population and a network. This stems from the actual dictionary definition, “the sum of human conditions and activity regarded as a whole functioning interdependently” [9]. It must be emphasized that a population of agents is not a society; the agents must be able to connect with others to comprise a social network.

Artificial society generation has very little historical use in video games, but has traditionally been used for network-based research for connectivity within geographical locations and analyzing spread rates and patterns throughout networks [10]. Furthermore, many of the artificial societies implemented to date have been macro-level simulations, dealing with the population as a whole, and treating its individual nodes as empty shells that only respond to stimuli within the simulation with one of a small number of possible responses.

Consequently, there is a need to explore methods for generating a large artificial society of socially-connected personified agents. Research in this area benefits both academia and the industry, opening up whole new realms of ideas for video games that make use of such societies and networked communities. This even creates a plethora of new sub-genres and formats for games in which the character could interact with NPCs in the society who contain a variety of human characteristics ranging from age, race, and religious beliefs, all the way to a list of all cities they ever lived in, and all companies they ever worked for. There are many scenarios that can be incorporated into games involving social networks. Agents who are emotionally connected to other individuals will be able to react with believable behaviours when a loved one or friend becomes ill or encounters strife. The agents can interact with the player and reveal information about other agents in the society who fit the criteria that the player is looking for, such as a religious leader or an athlete on the local hockey team. Endless options arise out of the incorporation of a networked society of micro-level intelligent agents within video games. There has been a trending decline in individual video games’ shelf life [11], so new advances, such as the currently discussed dynamic artificial societies, may be factors in helping to revitalize video games of certain genres and increase games' longevity that static, scripted gameplay is struggling to yield.

Generating a non-scripted artificial society is a balancing act with the population and its social network. It becomes especially difficult with bi-directional interdependencies between the population and the network; that is, when particular aspects and attributes of the agents depend on their relationships and when their social network forms as a result of their personality and attributes.

This thesis research combines the vision for non-scripted believable game characters and the principles of artificial societies to explore methodologies for generating populations of human-like entities. For the purpose of this thesis, we propose a computational model of a society for use in a video game, virtual world, or simulation. We take a flexible and general approach that can be readily adapted or modified to suit the needs of the particular game or simulation. We do not claim that the proposed framework will generate a perfect, comprehensive society or that the features and parameters we use are optimal and scientifically correct. Several psychosocial concepts are borrowed and integrated into various aspects of the model, but we also make many assumptions and simplifications where necessary. We are not experts on society; we are proposing a framework that uses numerous modular functions and parameters so that others could make the necessary adjustments to generate a society that suits their specific needs.

1.1 Thesis Questions

We begin with three questions upon which this research is based:

1. Can we generate an artificial society?
2. How well can our society "look" like a realistic society?
3. Can the society be generated in real-time or production-time? What are the resource usage requirements to generate a society?

These three questions are an important foundation to guide our research in a thorough manner. We begin with the opening broad question of the plausibility of this work. The latter two questions refine the research to evaluate the reliability of such methodologies.

1.2 Thesis Structure

This first chapter was an introduction to the subject matter related to the thesis. The remainder of the thesis is divided into five chapters of content. Chapter 2 provides a review of the literature in this field. Chapter 3 provides a detailed model design, and then our prototype system is described in Chapter 4. The experimental results are given in Chapter 5, followed concluding remarks and future work in Chapter 6.

Chapter 2

2 Related Works

There is very little literature on the generation of artificial societies for video game characters. Traditionally virtual societies have been used for the purpose of analyzing the flow of disease, knowledge, or other elements over a networked macro-level population. Video games have historically scripted in the characters required for the game, and any extra characters who are not necessary for gameplay would be added as empty shells (lacking personality and behavior) serving more of an aesthetical purpose rather than being an intelligent and realistic entity. As far as we are aware, the marriage of these concepts has not been done before. Related works will be discussed for each of the contributing fields of study: Artificial Societies, Population Generation, Believable Agents, and Non-Scripted Agent Intelligence.

2.1 Artificial Societies

Epstein and Axtell describe a method for generating an artificial society consisting of three parts: a population, an environment, and a set of rules. Their society is agent-based and micro-level, but the agents are only identified by a small number of attributes, so they lack personality and other believable properties. The agents are not networked together from the generation phase, but the set of rules govern the agents' behaviours and interactions with one another in a cellular automata approach, allowing them to bond and form relationships at runtime [12].

A mathematical algorithm has been proposed which examines large networked populations to identify the densely connected sub-networks, or communities [13]. This community detection algorithm has been incorporated into an artificial society generation method known as the friend attachment (FA) model. In this model, the social network in is grown over a series of timesteps. At each timestep, the entire network is divided into distinct communities using the algorithm in [13], and then as each new agent is created, they are added to a randomly selected community, or linked to several random individuals throughout the general population. As time goes on the networks are re-evaluated and the communities of interconnected agents become clearer. This work focuses on social connectedness and building networks [14], but does not treat the society on a micro-scale with individual personified agents.

2.2 Population Generation

A model for generating a population of vehicular drivers was proposed in [15]. The method uses unsupervised learning from sample data to determine agent classifications and parameter values, which it then feeds into the population generator algorithm. This creates the population of agents according to the determined demographics, and the agents' classes govern how they behave in traffic [15].

A team of geographers in the UK present a simulation to estimate the evolution of Britain into the coming years. The simulation uses historical census data to calculate the projected populations and demographics for the following decade [16].

An agent-based population simulation was developed for tracking the spread of H1N1 influenza over a socially networked school setting. The agents, who are split into student and teacher roles, are assigned demographic attributes and are connected with others in a social network. A time schedule governs the positions of each of the students and teachers, and their interactivity with the others. The agents are simple one-dimensional nodes, attributed only with a health status: susceptible, infectious, or recovered. As the simulation runs, the agents may spread the disease to those who they are in close contact with; but there is also a period of recovery for those who have been infected [17].

Three synthetic population methodologies were explored for creating a number of individuals who match a given sample demographic. These techniques are mathematical and scientific optimization algorithms, and are used to determine the best parameters and weights such that there is a minimal error between the artificial population demographics and the sample data [18].

2.3 Believable Agents

Researchers and simulation developers often strive to create outstanding AI for creating believable agents, but the term "believable" can be ambiguous.

One vital element that makes an agent believable is its autonomy and ability to make rational decisions. A generalization, which sounds like a derivative of the Turing test, is that agents are believable if their behaviour causes the audience (or players) to "suspend their disbelief" that they are virtual [19]. Furthermore, agents must have individual personality traits that define their persona and have an effect on the decisions they make. Loyall wrote a doctorate thesis on building believable agents incorporating several facets of realism from personality, emotion, goal-seeking and achieving, and natural language dialogue. Loyall compiles a list of seven requirements for creating believable agents: Personality, Emotion, Self-Motivation, Change, Social Relationships, Consistency of Expression, and Illusion of Life [19]. The manifestation of all or most of these seven attributes is necessary for creating the element of believability in virtual agents. This is a great guideline to follow when attempting to create human-like believable agents.

Computer scientists have had to look into the social sciences so that the virtual agents could be modeled after actual human behaviours. Psychological foundations have been used extensively in populating an agent with a personality and emotions [3, 20], and sociological concepts have also been availed in building believable agents with relational needs [3, 21]. These psychosocial elements are a vital foundation in creating believable agents.

The authors of [3] describe complex models of personality traits and emotions ranked hierarchically and weighted against other traits. For the sociological component, the authors examine role-based relationships to model the various forms of agent-agent connections, and the agents' behaviours and actions are influenced by the specific role relationships they have with one another.

2.4 Non-Scripted Agent Intelligence

Several researchers and developers are also exploring ways to create autonomous agents, who are completely driven by AI, and not by pre-scripted trigger actions [10, 22]. Autonomous agents should be able to rationally create goals for themselves and an action plan from which the goals would reasonably be attainable [8]. A common technique for such autonomy is the belief-desire-intention (BDI) architecture, in which agents are given a set of preconceived beliefs from which they form objectives, or desires. The agents then create a rational action plan to attain those desires [20, 5]. There has also been a lot of work towards learning techniques to allow virtual agents to grow their knowledge-base and strategizing methods by learning from their past mistakes and achievements, and from social interactions with other agents [23, 24, 25].

2.5 Research Gap

As explained in the previous sections of the related works, there has been significant research into many areas of AI over the years. The overlap of these several areas has not been investigated to date. Thus, a large research gap is left in the merging of these AI concepts to generate a population of personified and socially networked agents, and particularly to do so without pre-scripting them in.

Chapter 3

3 Modeling and Generation of Artificial Societies

We present an abstract framework design in this chapter. Despite being a high-level design for a society generation framework, this chapter is very detailed and thoroughly explains every aspect of the proposed approach. This framework has been designed in a flexible and modular fashion so that individual features can be easily modified. We have selected specific models and parameters to perform the tasks at hand, but in essentially every aspect, these can be replaced at the modular level, and smaller modifications can be made simply by tweaking parameters in the external configuration files.

We first define the atomic units of the population, the agents, and then the different formations of agents are listed in Section 3.2. In Section 3.3 we explain the process for assigning agents into their workplaces. The romantic relationships between agents are explained in Section 3.4, including the creation of such relationships, calculating their compatibility and strength, and the option of having children. Section 3.5 describes and illustrates the agent creation process. We then cover the various forms of simulation models for society generation in Section 3.6. Lastly, the social network and the sub-processes associated with forming friendships are presented in Section 3.7. Because this chapter is lengthy and contains many detailed processes, we end with a summary of the key portions in Section 3.8.

3.1 Agents

Being a micro-level artificial society, we are creating an agent-based population with each agent representing a human. The agents contain several human-like attributes including personality traits, records of their past experiences, current status and situation, and a network of friendships. An overview of the agents' attribute categories is illustrated in Figure 3.1. Each of these categories will be explained in more detail and illustrated with additional figures in the following subsections.

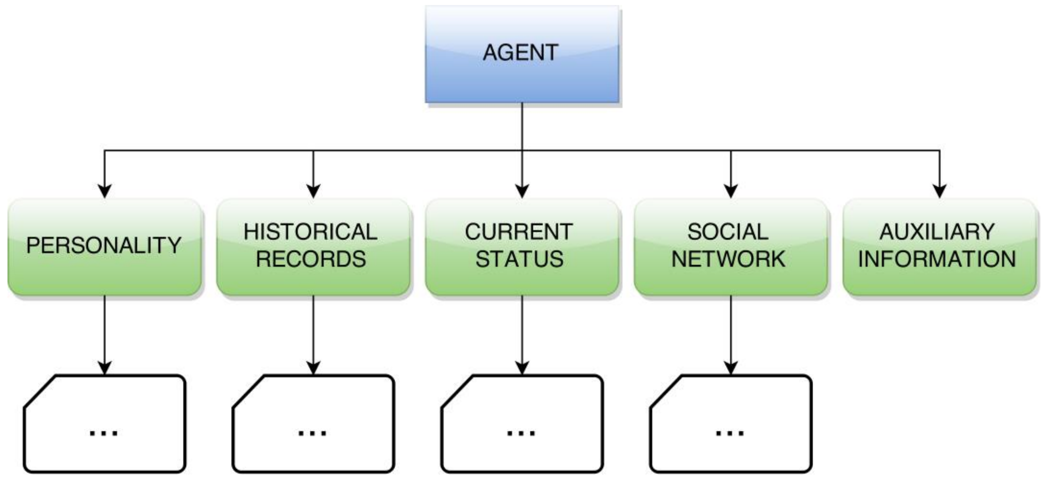

Figure 3.1: Overview of an agent's attribute categories.

3.1.1 Personality

Several models have been proposed that encompass a wide range of personality trait combinations. When one model does not sufficiently cover all the personality profiles required for the specific research needs, models can be supplemented with additional traits or grouped together with other models [3].

For our purposes in this thesis, the popular Myers-Briggs Type Indicator (MBTI) model is the basis of the agents' personality profile. In addition to the four trait categories from MBTI, we have added Intelligence and Athleticism as two supplemental traits, which are all shown in Figure 3.2. Note that the framework is not bound to the MBTI model, but *a* model is required to support personality so we selected the MBTI for this work.

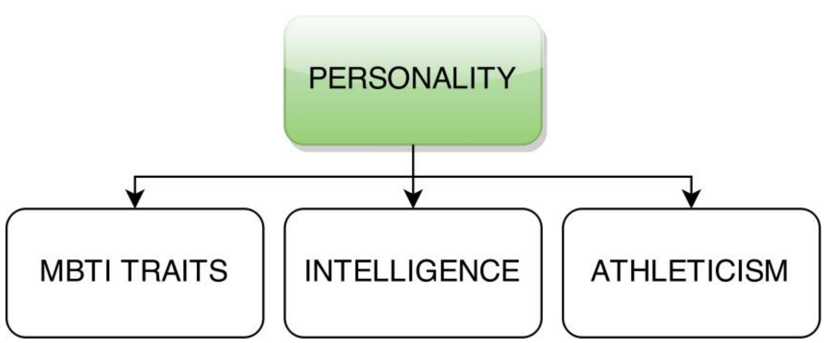

Figure 3.2: Agent's attributes under the "Personality" category.

Each of the four basis traits are measured with ternary values on [0.0, 1.0]. The values of 0 and 1 for a given trait are respective indicators for off and on. For example, for the Introversion/Extraversion trait, a value of 0 represents complete Introversion while 1 is complete Extraversion. An agent may also display both sides of a trait dimension, and thus are assigned 0.5. This means that the agent expresses both traits on that axis equally. The continuous scale on [0.0,1.0] was unnecessary for this simulation, as we do not require a percentage of how much an individual exemplifies a given trait, but rather whether or not they possess that trait.

The additional two traits, however, are measured on continuous scales on [0.0,1.0]. Both intelligence and athleticism are attributes that come in a range of degrees, and thus it is best represented in this way rather than with a discrete on/off setting.

3.1.2 Historical Records

Agents are assigned several historical archives to keep track of certain forms of activity or placement from their entire life. The primary purpose for keeping all this historical information is for the group assignment process, in which the agents are put into groups for the activities they were involved in; including school, work, among various other involvements.

There are 3 types of archives in the current implementation, as illustrated in Figure 3.3. These historical records are: (i) **hometown archive** keeping track of all cities the person has resided in; (ii) **school archive** recording which schools (post-secondary institutions included) they attended and the institution type; and (iii) **work archive** including every workplace that employed the person as well as their job position there.

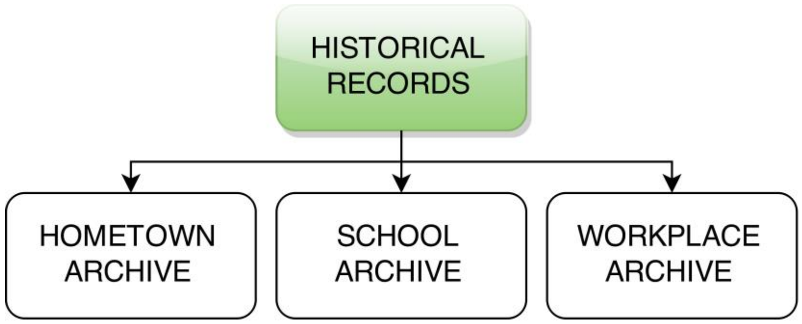

Figure 3.3: Agent's attributes under the "Historical Records" category.

In addition to each of these archives described above, there is a secondary archive of each type which only includes those activities that take place in the local society. For example, consider an agent who lived locally in London in childhood, then moved to Ottawa, Toronto, and then back to London. Then the primary hometown history would contain entries for all four cities: London, Ottawa, Toronto, and London. The local hometown history, however, would contain the two entries: London, and London. Note that since they are not chronologically consecutive they are listed separately.

These archives can also play an important role in future work in a real-time game involving these agents, as they record elements of the individuals' past experiences. This idea is discussed in more detail in the Future Work chapter.

3.1.2.1 Hometown Archive

Each person contains an archive of all hometowns from birth until present. Each entry in the archive retains the city name and the period in which they lived there. In a more advanced simulation, the agents could be given home addresses where they reside within the cities, and that information could also be included in the hometown archive.

When a couple is created in the initial population, they are assumed to have lived together from the time of their wedding to the present, so a single list of hometowns is generated for that period and used in both agents' archive. Before their marriage, however, they are assigned histories individually and independently.

Similarly, we assume that a child born from a married couple is living with the parents from child birth until the end of secondary school – at which point they are considered adults and able to move out on their own. During that entire childhood phase, the child's hometowns are identical to that of its parents for the entire period. After the child finishes secondary school, the dependency is removed and they can live anywhere, regardless of the parents' hometown.

Several assumptions or simplifications are made regarding the shared hometowns between married agents and couple's children, but this was a design decision for this current work. The modularity of this framework allows for other approaches to be easily added and tuned.

3.1.2.2 School Archive

The agents also keep a history of which schools they attended, including both public school (elementary and secondary) and post-secondary institutions. The name and school type are recorded in each entry in the archive. The primary purpose for recording the schools is for the friend network process, so that schoolmates are grouped together as possible friends. For more information, see the **Groups** discussion in Section 3.7.5.

3.1.2.3 Workplace Archive

The workplaces are also archived for each agent over the periods that they work at each place. Since there may be several career positions employed by one workplace, both the workplace and the person's career are recorded in each entry. For example, the Clinic employs Doctors, Physicians, and Nurses. Thus, the career must be recorded in addition to the workplace.

3.1.3 Current Status

In addition to the personality profile previously discussed, each agent has a plethora of attributes comprising their individual identity. These include: age, sex, race, religion, career path, income, and a list of interests. There are numerous attributes that fall into this category, as well as some that could be cross-listed with other categories. A summary of some of the primary current status attributes are displayed in Figure 3.4.

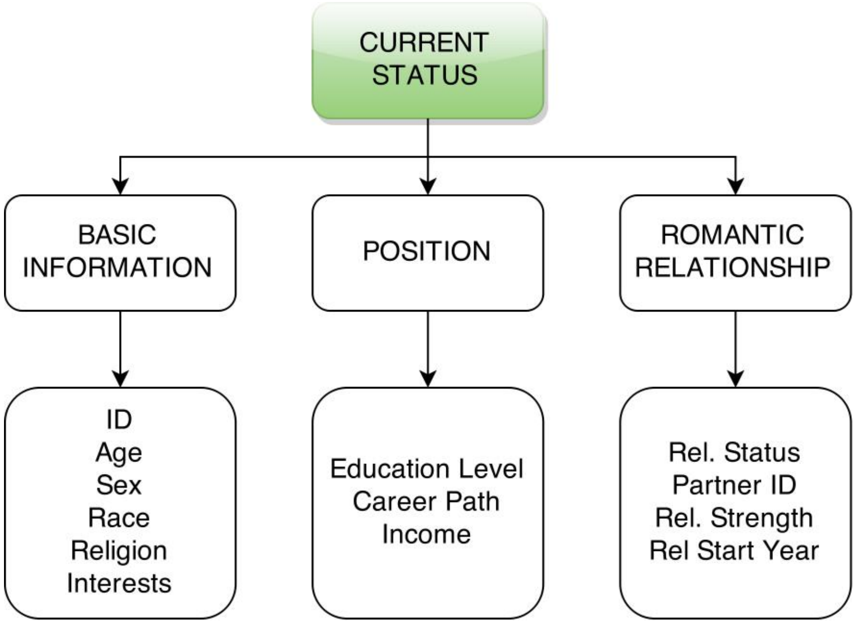

Figure 3.4: Agent's attributes under the "Current Status" category.

3.1.3.1 Age

During the generation of the initial population of adult dyadic agents, one agent is assigned a random age from a normal distribution, ranging from 20 to 90 years old. The agent's partner in the dyad is then given a close age within a fixed number of years on either side.

When a child is created from two parents, its age is computed from the parents' ages and marital longitude.

3.1.3.2 Sex

Each agent is given a random sex. The current simulation only uses heterosexual dyads, so each couple consists of one male and one female, but the framework is flexible enough that some small modifications could also incorporate homosexual dyads in the simulation.

3.1.3.3 Race

In a dyad, one person is assigned a random race from weighted probabilities. The list of possible races and their corresponding weights are parameters from a configuration file.

For these trials, we used the following list of races: *Caucasian, African American, Aboriginal*, *Indian*, *East Asian*, and *Hispanic*. These races are simply text labels and can be easily modified.

From a look-up table, the second person is assigned a race with probabilities based upon the first person's race. Typically, the highest likelihood is that the two will be of the same race, and then each of the other possible races is less likely. This promotes racial homogamy in relationships.

When an interracial couple has a child, one random parent is chosen and the child is given the same race as that parent. This way, we simplify the problem of children of mixed race, which especially gets progressively more difficult to handle over many generations of potential interracial families.

3.1.3.4 Religion

The religious beliefs are handled similarly to the races. One person is assigned a random religion from weighted probabilities. Their partner is then assigned a religion from a set of probabilities from the first person's religion. In most cases, the highest likelihood is that the two will be of the same religion. This promotes religious homogamy in relationships.

We have to make simplifications when dealing with more complicated scenarios of religious heterogamy in families, but the framework is general and modular so other approaches can be easily inserted to replace these selected methods. When an interreligious couple has a child, one random parent is chosen and the child is given the same religion as that parent. This simplifies the problem of children being pulled in two different directions of religion, as well as the issue of children seeking their own faith different from one or both of their parents.

The religions and their weights are stored in a configuration file so they can be easily changed. For these simulations, we use the following list of religions: *Catholicism, Protestantism, Judaism, Islam*, *Buddhism, Hinduism*, and *None*. The "none" option represents a non-religious or agnostic belief. Like the race, these religions are simply text labels and can be easily modified.

3.1.3.5 Career Path and Income

When an agent is first created and given a random personality profile, their career path is then chosen immediately. Based upon their traits and possible additional attributes (race, religion, sex, etc.), an appropriate career is selected for the person. The career is also chosen based on the number of job openings for that career at the time the person is created – not at the time they begin working, since their career path must be assigned initially. The number of job openings is computed from the total current population multiplied by that particular job's given weight for number of openings per capita. At the same time the career path is made, the educational requirements are recorded and stored in the agent object. The education for some careers is just secondary school completion, while many others require college or university degrees, and the number of post-secondary years is recorded too.

Although the career path and required education are recorded with the person from creation, their age dictates whether or not they are actually working that career at the current time. They must be old enough to have gone through all the required education and then are able to start working.

3.1.3.6 Interests

The agents each have a set of interests and/or values. These interests can represent any number of things they enjoy (food, music, going out with friends) or esteem highly in their life (work, family, religious beliefs); but the ordering must be identical across the entire population, as these interests will be compared to others to calculate interest similarity [26].

For this simulation, we give every agent an interest vector of seven elements. The first three represent race, religion, and love for sports, and the remaining four are arbitrary unlabeled interest fields. Each race and religion are given mean levels of ardency, so the agent's interest values for these two fields are taken as a random number within ±10 of their respective race and religion's means levels. Their love for athletics is taken directly from their own level of athleticism for simplicity. The remaining arbitrary interests are chosen from a normal distribution. All interest values are scaled to the interval [0,100].

In addition to the interest values, each agent contains a vector of interest weights for the measure of the importance of each of those interests. Typically the weights would cover a spectrum of percentages, with some interests heavily weighted for major importance and others with a very small importance weight. In this simulation, we simplify the weights by treating all interests equally, so each is computed as $w = 1 / numWeights$.

The interest weights and values are used in calculating the interest similarity between two persons. We calculate the interest similarity only for romantic relationships, but this could be applied to all forms of relationship and acquaintanceship as well. In fact, Sun and Wang's proposed method of society growth is based upon a pool of individuals who form friendships with others with whom they yield an interest similarity of significant magnitude [26]. In our simulation, we do not use the interest simulation for forming relationships, but to measure how similar a romantic couple is; which is also a factor in calculating their overall relationship strength.

3.1.4 Social Network

Each person is assigned several friendships and relationships with other agents in the population. They each contain lists of family members, broken down into distinct lists for parents, children, and siblings.

Additionally, they contain the primary list of all friends (including those family members mentioned above) of all kinds. These are the chief lists from which the entire social network for the whole society is created. These different social network attributes are summarized in the tree in Figure 3.5.

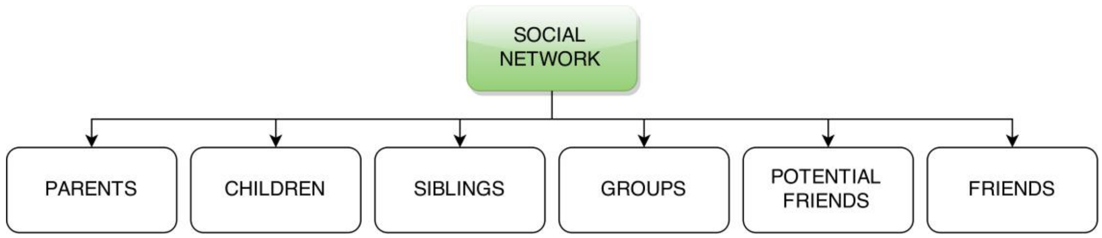

Figure 3.5: Agent's attributes under the "Social Network" category.

The connections created between individuals are given a type indicator to represent whether it is a marriage or courtship, kinship, or friendship. An additional descriptor is attached to each to provide additional details about the friendship origins (i.e. Teacher-Student, Co-worker, etc.).

3.1.5 Auxiliary Information

The agents contain numerous additional variables that are auxiliary, intermediate, or miscellaneous. Many of them are trivial but beneficial in the assignment of other attributes. For example, agents contain an auxiliary Boolean flag, isInSchool, so that their school and post-school involvements can be selected easily.

Another example is the agents' expected year of death. At the time of their creation, each agent is assigned an expiry date which will govern how long they live. An agent's expected year of death is taken from a normal distribution with $\mu = 80$ and $\sigma^2 = 4$ to give an even spread of death ages between 65 and 95.

3.2 Agent Formations

As described above, the agents are individualistic so each one contains its own unique characteristics, attributes, life history, and social circle. However, there are a few forms of cluster units, based upon human sociology, that are used in the simulation.

3.2.1 Monad

An individual single agent is known as a **monad**. Although every agent is itself a monad, we consider only those that are singular (not in a relationship) as monadic.

3.2.2 Dyad

A set of two agents in a romantic relationship comprise a **dyad** unit. Although each agent is atomically a monad, they are seen as part of a dyad along with their partner for this simulation. The reason is that in some generation models, couples are created as a unit with some shared characteristics and attributes. Thus, a dyad is different than two singular monads because the agents in the dyad are created as an interdependent unit of two, rather than two independent units.

Sociologically, a dyad includes any pair of two individuals in any form of connection. This would encompass everything from romantic relationships to family or friend relations, and even less relational role connections (i.e. doctor-patient) [27]. This simulation includes all those forms of relationships; however we use the term dyad strictly with respect to romantic relationships, due to the inherent need for dependencies when forming such relationships.

3.2.3 Group

In addition to their dyadic relationships, agents are placed into several groups for school, work, religious, and extra-curricular activities. These groups are a little different from human's large group circles, in that the agents aren't connected to all members of the groups they are in, but only a relatively small subset of them. The groups in the simulation are in fact the basis for making friends, as the agents make friends only from others who share in their groups. See Section 3.7.5: **Groups** for more details.

3.3 Workplace Assignment

As described in Section 3.1.3.5: **Career Path and Income** above, each agent is assigned a career path at the time of their creation, and they go through all the mandatory education required for their career path.

When an agent is finished all their education, they seek employment in their career field. The simulation is focused on the one local society but also contains people moving in and out from the main society, so we must include all people in the employment-search phase. Agents will always first look for work in the city they currently reside in, and if they do not find a job or receive the job they hope for, then they search externally. This applies to people within the main society who begin by looking locally and then search externally, and for people living in other cities who first look in their own hometown and then look to the focus city after.

When searching for employment in the simulation's focus society, a list is produced of possible workplaces that employ the given career path. For example, the career path "Computer Technical Support" is associated with two workplaces: "Geek Squad" and "Londonsoft Software Inc.". From the list of possible workplaces, one is chosen uniformly randomly as the job that the person is applying for. For simplicity, just the one job is examined as a possible job. The person then has a certain probability (60% in the current simulation) of receiving the job. If they are local people beginning their search here, then they could proceed to search externally if they do not receive the local job.

Due to the possibility of agents finding work outside of their hometown, those agents and their families must move to the city where they are newly employed. Thus, there will be a continual flow of people into and out of the local society, but also a fairly consistent base population.

3.4 Romantic Relationships

Relationships are a vital part to a society's maintenance and especially to its growth. Without people mating and reproducing, the society would die out very quickly. However, people typically seek out relationships for their own desires and not for the interest of preserving their society's population. Either way, romantic relationships are important, common, and a major component in a real society, and thus a necessary feature in an artificial society aimed to be realistic. The agents in the virtual society are looking for a partner to date and eventually to marry and begin a family with. For simplicity, we only allow heterosexual romantic relationships in this society, but with a few modifications, the framework could also include homosexual relationships.

3.4.1 Relationship Creation

Relationships are created in one of two ways, depending on the simulation model being used (see Section 3.6.3, **Population Generation Models**, for more details). In the case of a "matching" model, i.e. Monadic Matching Model, relationships are formed from a pool of single individuals. The other technique for creating relationships, as in the Dyadic Model and Genesis Model, is to initially form a dyad (couple) as a unit, and fill in some common attributes based or influenced by one another.

3.4.2 Relationship Strength

Every relationship needs a strength measurement to indicate and govern the couple's actions and the structure of their family. The couple's decision to have children will depend partially upon the strength of their relationship. A couple will be more likely to have children if they have a strong and secure relationship rather than if it is a little shaky, but on the other hand, some couples will want to have children in the hopes that their weak relationship grows stronger with the new additions to the family. A more detailed discussion on relationships and children is in Section 3.4.5 below). The relationship strength is calculated twice in the simulation: once initially to dictate whether or not the couple has children and if they remain together or separate; and then it calculated after the children generation stage to determine how strong the relationship is after having children.

There are several factors that impact the strength of a relationship including racial and religious homogamy, age similarity, children, economic homogamy, interests, and the type and longitude of their relationship. In the real world there are many additional interpersonal, societal, and intergenerational factors that impact a relationship's strength. These include previous sexual relations, pre-marital cohabitation and parenthood, educational and economic homogamy, as well as the strength of each of their parents' marriages. For simplicity, we omit the generational and factors relating to past relationships, and calculate the strength based on the how homogamous the individuals are in race, religion, education, income; as well as their relationship type, longitude, and number of children. In addition, there is a random component to the relationship strength since we cannot accurately measure the strength of a relationship with logical formulae. There are always unpredictable and unquantifiable factors that can increase or decrease the health of a relationship.

3.4.3 Interest Similarity

One of the main contributing factors in measuring relationship strength is how compatible the couple is in terms of their interests and values. A measure of interest similarity for between two individuals has been presented in [Sun08]. The interest similarity for persons *i* and *j* for a set of *n* interests is defined as:

$$I.S.(i,j) = 1 - \sqrt{\frac{\sum_{h=1}^{n} w_{ih}^2 (I_{ih} - I_{jh})^2}{\sum_{h=1}^{n} w_{ih}^2}}$$

where w_{ph} is person p's weight for interest h and I_{ph} is person p's value for interest h.

As described in the **Interests** discussion in Section 3.1.3.6 above, we use vectors of seven interest weights and values in this simulation. The first three interest fields are for race, religion, and love for athletics, and the remaining four are arbitrary fields.

3.4.4 Assortative Mating

People tend to become romantically involved with those with whom they are compatible. Assortative mating describes people coupling with others with whom they share specific attributes or are similarly positioned. This includes homogamy in age, race, and religious beliefs, as well as economic equality. For example, a person who is making a salary of $120,000 is not likely to be with a spouse making only $40,000. There is a large gap in their income levels, and thus they are not very likely to attract one another, at least in that aspect [28, 29].

These homogamous factors are incorporated into the simulation in the creation of spousal dyads, and in calculating their relationship strength as time goes on. When a married couple is generated, one person is given a set of random attributes first. Then those attributes influence the attributes of that person's spouse. The race, religion, and education of the second person have a higher probability of matching that of their spouse.

The age of the second person is determined from their spouse's age within a range of a specific number of years. In this simulation, the education level and income equality are not included in the marriage creations, but are factors in the calculation of their relationship strength. The education levels are checked for an exact match to determine their homogamy. College education is different than University education, so those two are **not** considered a match in this calculation. The income factor is calculated with discrete bins since income is such a flexible attribute; so if the couple's incomes are within a certain threshold of one another, they are considered in the same income level. The current configuration uses a threshold of $30,000.

3.4.5 Children

Couples may have children. For simplicity, we are assuming only married couples have children in this simulation. A number of factors will determine whether or not a couple has children as well as how many they have. Factors include the relationship strength, relationship type (marriage, courtship, etc.), and a random variable to allow for anomalies. If a couple has been found to have children, then the number of children is chosen from a distribution. Our current simulation configuration uses a normal distribution with mean 2.0 and variance 1.4; with additional bounding limits of [1.0, 5.0]. The reason that the lower limit is not 0 is because there was already a check to determine whether or not the couple is having children, so they cannot have zero if they are already determined to have children.

Once the children are created, it is assumed that a family is inter-connected in relationships, so each person is connected to each other person in the family in parent-child, child-parent, and child-child connections, and of course parent-parent are already connected from their relationship (more on this in Section 3.7).

The timing of a couple having children is taken from a normal distribution spanning the decade beginning at the couple's marriage year. This seemed a reasonable distribution as many couples wait several years before having children so they can enjoy their time together without having to deal with the responsibility of children, and so that they can build up their finances first. Furthermore, the couples typically don't wait too long, so the normal distribution seemed a suitable fit for this implementation, but of course is a parameter than can easily be changed.

In some population generation models, having children is essential to keeping the model alive - just as it is in the real world. Some models require several generations to achieve a society that is well populated and reasonably stable demographically.

3.5 Agent Creation

Agents can be categorized in a number of ways due to their relationship status and role. These agent categories include single adult; in courtship; married; and child. Each agent is created with the same general process, but with variability dependent upon the type of agent. All agents begin as an empty Person instance without any attributes attached yet. Then, depending on the kind of agent being created, a series of attributes may be assigned directly to the agent immediately.

After any attributes are directly added to a new agent from this first phase, every agent (of any kind) undergoes a "*complete*" process in which all gaps of missing attributes are filled in. Regardless of what information the agents may have been given initially, this final process fills any and all attributes that are empty at this point, so all agents coming out of this phase will have complete personae, no matter the form in which they were created.

3.5.1 Phase 1 – Initialization

This first phase is where some agents will be given attributes as a result of dependencies between this agent and one or more other agents. Each of categories of agents has a different set of attributes ascribed in this phase because of the different dependencies.

3.5.1.1 Single Adult Agent

A single adult agent has no dependencies on any other agents, so no attributes are given initially to this agent. This is the simplest form of agent creation, as nothing occurs during this first phase. All traits, attributes, and other elements will be assigned solely during the complete (second) phase.

3.5.1.2 Agent in Relationship

Agents who are in romantic relationships at the time of the society's creation are made as a dyadic unit of the two agents. There are several dependencies between two individuals in a relationship, so the couple is made at once with such dependent attributes formed once and assigned to the two together.

There is a small variation between a couple who is married and a couple in a courtship, but the main process is the same. This process for creating a couple as a dyad is illustrated in Figure 3.6 below.

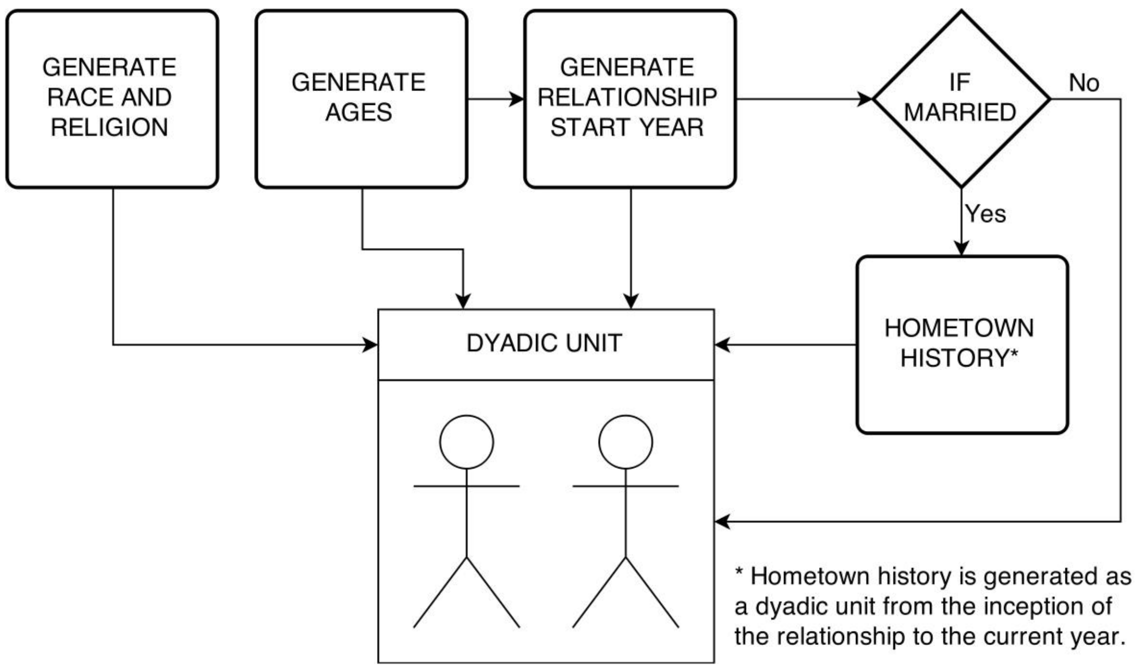

Figure 3.6: Creation process of a married couple.

3.5.1.3 Child Agent

When a couple has a child, there are also several dependencies between the parents and the child. The child will inherit various attributes genetically or through their upbringing, as depicted in Figure 3.7. The child's race and religious beliefs are assumed to be taken from their parents. These two attributes are handled in the same way: if both parents are homogamous in the attribute, then the child will also share that homogamous attribute; but if the parents are heterogeneous on the attribute, then the one of those two attribute options is selected randomly for the child. This assumption helps eliminate issues of mixed race among interracial marriages, as well as mixed religious views. Each person will have one discrete race and one discrete religion.

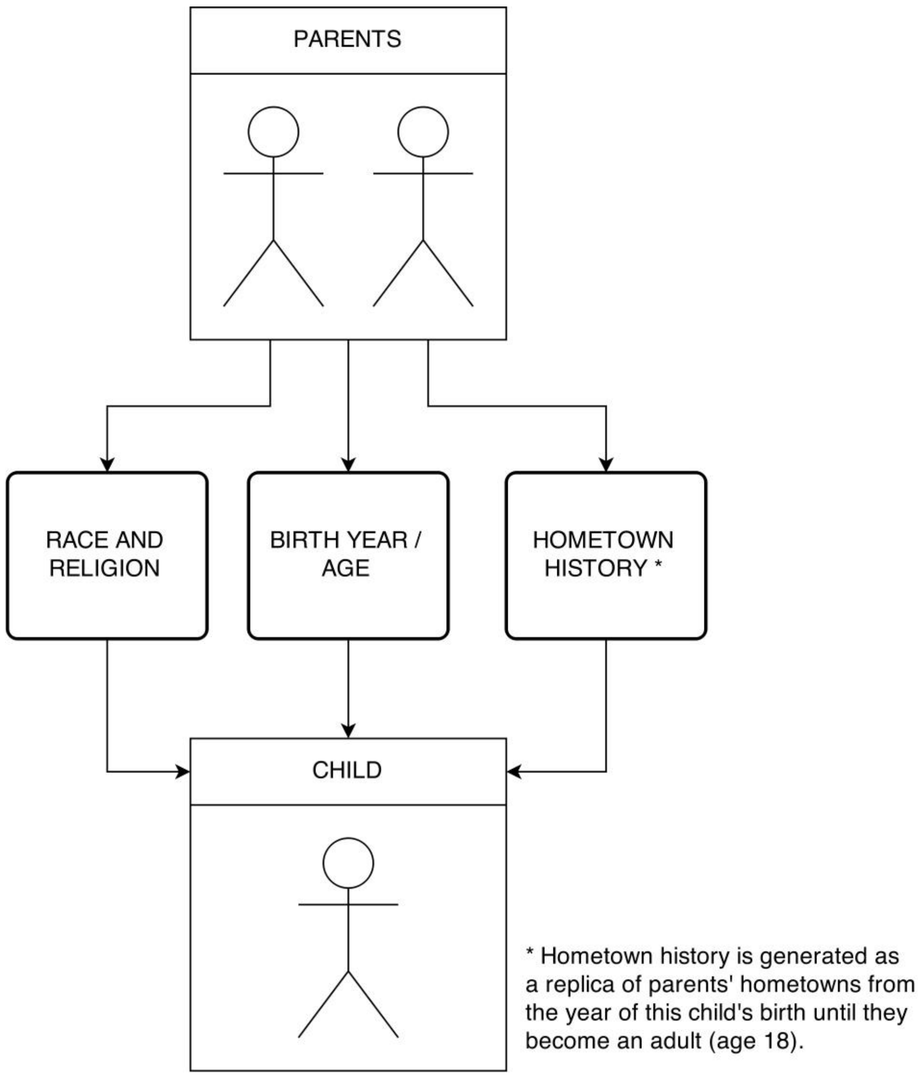

Figure 3.7: Creation process of a child.

3.5.2 Phase 2 – Completion

After the first phase of receiving an initial set of attributes, the person goes through the second and final phase to complete them. The *complete* process is analogous to an object traveling along a conveyor belt in an assembly line. The agent travels metaphorically along a conveyor belt and invokes a sub-process for every attribute. Figures 3.8 and 3.9

illustrate the overall conveyor belt process and the individual attribute sub-processes, respectively. The *Attribute Assigner* function for attribute *attr*$_i$ first checks whether or not the given agent has a value for the attribute *attr*$_i$. If the agent already has a value for that attribute, then nothing has to be done, so the agent continues along the conveyor belt to the next attribute. If that attribute was found to be empty, then the appropriate method is invoked to select (typically randomly) an appropriate attribute value to assign to the agent. This sub-process to assign the necessary attributes to an agent is shown in the Figure below.

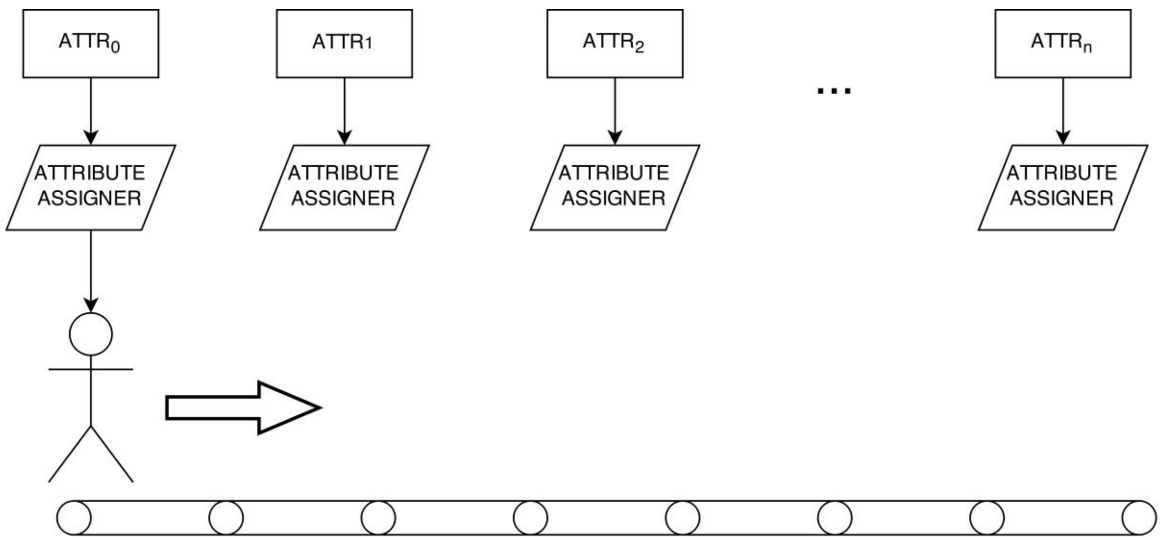

Figure 3.8: Conveyor-belt process for assigning attributes to agent.

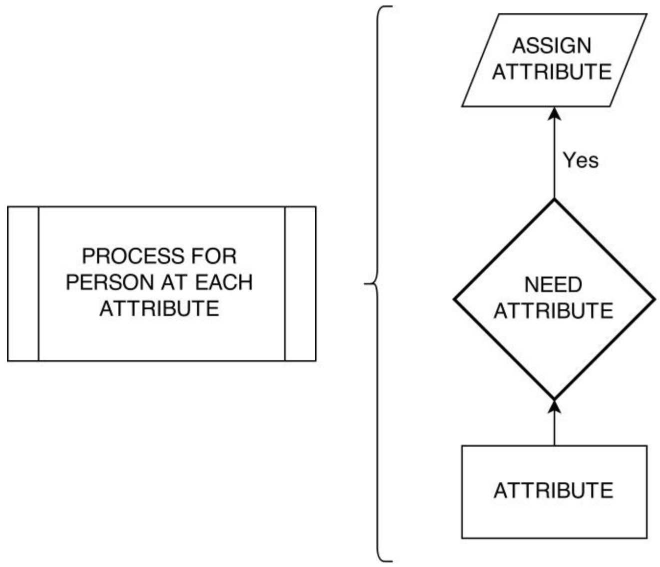

Figure 3.9: Sub-process from conveyor-belt process, for checking if attribute needs to be assigned to agent, and if so, proceeding to assign that attribute.

3.6 Simulation Models

3.6.1 Model Directions

A realistic artificial society must contain both a demographic-profiled population of individuals as well as a realistic network of relationships amongst the agents. The challenge is creating a society that maintains a strong balance of the individual agents' profiles and the global social network.

Creating an artificial society can be done from a **top-down** approach in which a network is created initially and then fitted with agents to match the network's role nodes or from a **bottom-up** approach in which the population of agents is first generated and then a network is created and overlaid on the population, as well as various other derivatives of one or both of these models.

In the top-down approach, the social network graph would be generated initially, on a macro-level, before individual agents are made, or at least before they are given personal attributes. The network would be a graph structure composed of simple, empty nodes and randomly generated connections representing relationships between the nodes. Once the network is created with the desired number of connections and density, then each of the nodes would be filled with a number of attributes to indicate the person that that node represents.

Creating a society in the opposite direction takes a bottom-up approach. For this kind of model, the individual agents are first created and profiled with many human-like traits and attributes. The agents are then linked with other agents to create the network of relationships on top of the population.

Some models may be heavily directed by one of these approaches but with some degree of the opposite approach incorporated as well. Although the two are quite contrary to one another, they are not completely mutually exclusive so some elements of both may be combined in a particular model.

Within each of these higher-level models, various attributes can be created or assigned in various orders based on the desired dependencies. In an example for career path assignment, one technique is to assign each person random personality attributes, and then select an appropriate career path based on the personality profile. A different approach, however, is to begin with generating a random career path for a person, and then reverse-engineering the person's profile from that career to determine their personality. There are many such pairs or groups of attributes in which the dependencies are ambiguous, and can be formed in a number of ways.

Likewise, there are different directional approaches for generating couples together in marriage or other relationships. Empty (attribute-less) individuals can be paired with one another randomly, and then assigned attributes together so that the two both exhibit some like attributes (see **Assortative Mating** in Section 3.4.4 for more details). Alternately, all individuals can be generated and assigned personalities and other attributes initially, and then relationships can be formed from the pool of available agents.

3.6.2 Active Simulation

Some of the generation models require the passing of time so that new relationships can be created, developed further, or break apart, so that the society can evolve in a way that static generation models do not allow. Every one of the models can incorporate an active running simulation to further develop and grow the population, but only some **require** it.

The overall active simulation cycle is depicted in Figure 3.10, and Figure 3.11 focuses in on the individual steps of the evaluation that occurs on each agent in the simulation.

Each timestep in the simulation is one year, as that is a reasonable amount of time to re-evaluate to check for changes and updates in the society on the micro-level. There is no need to simulate smaller timesteps since most forms of updates are recorded or observed in year steps. At each year step in the active simulation, the entire society is evaluated one agent at a time. After all agents have been evaluated for that year, then anyone who was set to die during that year is removed from the society at the end of the year. That concludes the process for each time step (or year) in the active simulation. The year is then incremented and the same evaluation process occurs again, until the simulation has reached the year of completion, as specified in the configuration file.

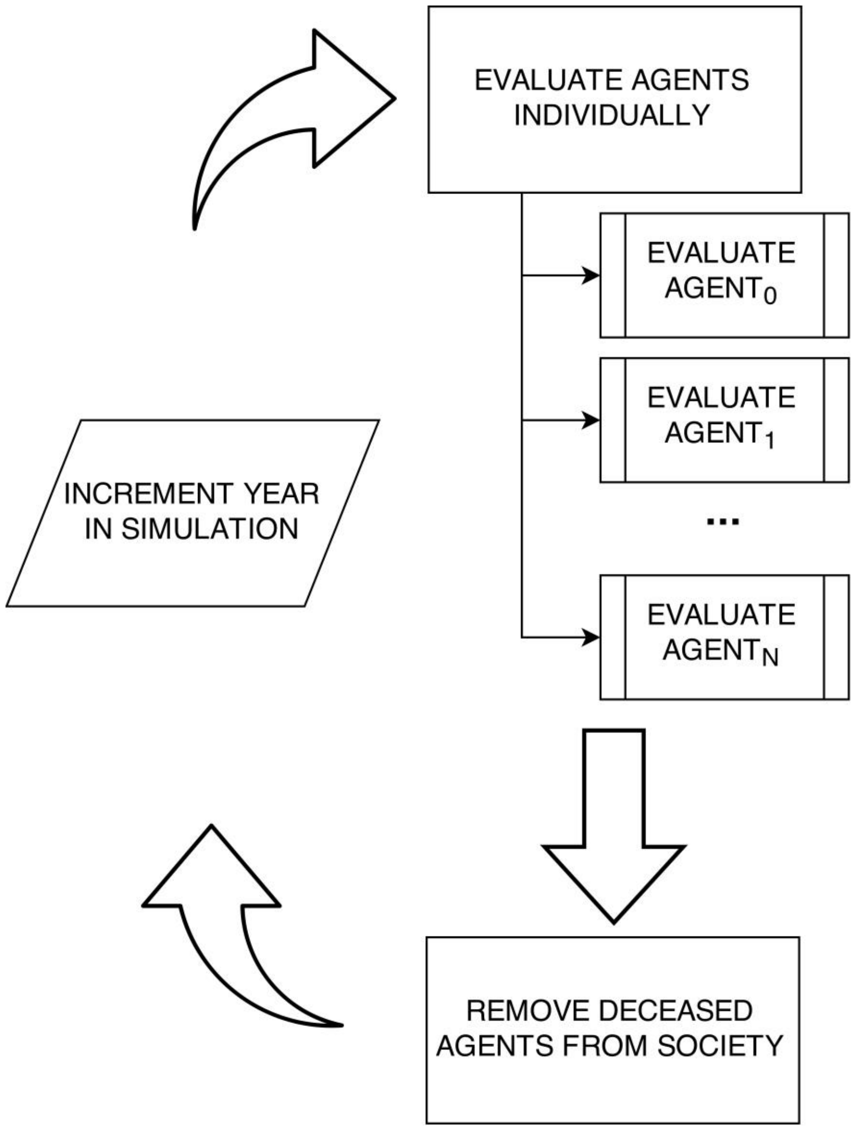

Figure 3.10: Flow process of an active simulation.

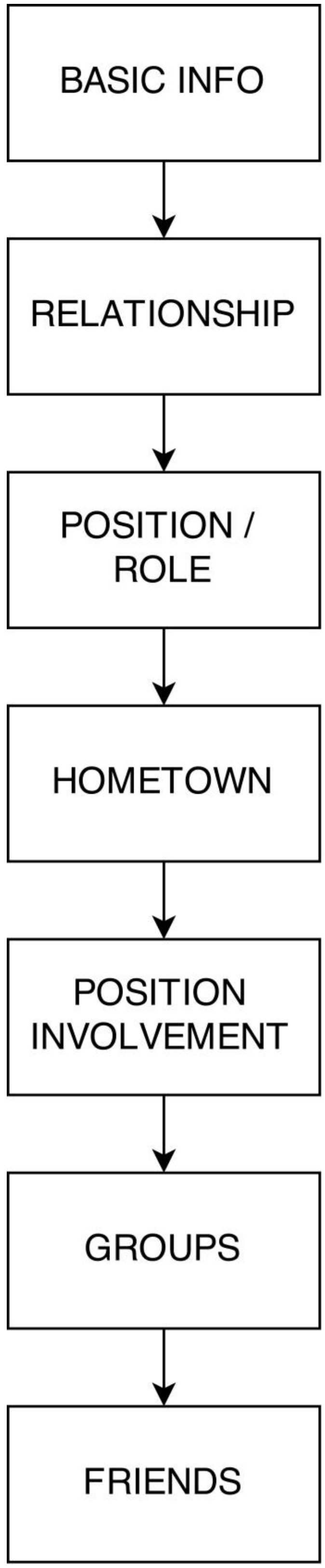

Figure 3.11: Sequence of evaluations on an agent during an active simulation.

3.6.2.1 Basic Information

In the current implementation, the person's age is the only basic information that is updated. Their age is re-computed, and thus incremented with each time step. Future considerations could include other basic attributes being re-evaluated, for example the person's religious beliefs, since those could change a little bit over time, especially during adolescence.

3.6.2.2 Death

Since each agent has been already assigned an expiry date since their conception, they are examined each year to see if they have reached the end of their life. In future work, there could be possibilities of accidents, murder, and health problems that could cause premature deaths of agents. For simplicity in this program, the life expectancy is the only factor.

When an individual has reached their expiry date, they immediately die and are removed from the society without a trace. This means the person is taken out of the global list of the population, and all connections to this person are removed. Any family members (spouse, parents, children, and siblings) and friends will lose those connections. This is done so that only the connections between living persons will remain. In future versions of the simulation, deceased persons could be handled differently so that the relationships and connections remain intact and are just labelled in a way that indicate a non-active connection. The deceased person is also removed from any groups and from all agents' lists of potential friends, so that no new friendships can be formed with deceased persons.

3.6.2.3 Relationship / Children

Singles in the society will have a chance of forming a relationship at each year. A series of factors, including their age and personality, will be used determine whether or not those singles are now in a relationship. The single agents will evaluate each agent from their list of potential friends – a list also used in the friendship selection process (see Section 3.7.6 for more information on this). The searching agent checks for each agent in the list if the two are compatible in a number of aspects including age difference, sex, and closeness in family. Note that in the Genesis model (see description in Section 3.6.3.4), close family members are still allowed to mate together initially. After those aspects are examined, the searching agent will focus on the potential mate with the highest compatibility, and will attempt to pursue that mate. A random number will dictate whether or not the two form a relationship.

In existing romantic relationships among agents, the relationship strength is re-calculated, since many factors may influence a change in their relationship from one year to the next.

For simplicity, there are no separations or divorces among marriages even if the relationship strength is very weak. In future work, however, the relationship strength can be compared to a given threshold, possibly in conjunction with additional factors, to govern the fate of the relationship. Couples who are not married can break up in this simulation. A number of factors are aggregated to determine the fate of the dating couple's relationship. The current factors for this calculation are the relationship strength (which itself is an aggregate of several factors), the number of years the couple has been dating, the total number of people in the society, and a random factor. The number of people in the entire society may seem like an unusual factor in computing a couple's fate; however it is a very important factor, particularly in the Genesis model. The population size impacts a relationship because individuals will gauge their relationship under the lens of how many potential partners are around, and whether or not their current relationship is likely an optimal relationship for them. In a small society with few people, a person has minimal potential mates, whereas in a large population, there are many potential partners. Thus, the population size is inversely proportional to the likelihood of a couple staying together. The Genesis model especially relies on this factor, as the initial population is one family and their lineage must spread out to fill the entire society, so the agents cannot be too choosey during those first few generations when there are not many people in the society.

Although we omit the possibility of separation, the relationship strength is still important in determining whether or not the couple will have children. Typically, couples who have a very healthy and secure marriage are more likely to have children than those who are less sturdy in their marriage. On the other hand, some couples in weak relationships may decide to have children in an effort to save the marriage. Thus, we use the calculated strength of relationship among other factors to determine whether or not the couple decides to have a child in the given year.

3.6.2.4 Current Position

The current position indicates the primary role of the agent at the current time. This is represented as a small finite state machine with the possible positions of CHILD, STUDENT, UNEMPLOYED, WORKING, RETIRED, and DEAD.

The *Child* phase indicates that the person is a toddler that has not even started school yet. From this phase, the toddler can only transfer into the *Student* phase, once they reach the correct age (4 years old in this simulation).

The *Student* phase lasts from when the person begins elementary school until they are finished all their education, including post-secondary, if that person requires such education for their career path. From here, a *Student* graduates and is assumed to be *Unemployed* immediately. Although some people may line up a job before graduating, we simplify this by beginning them as *Unemployed* and from there they can search for work.

Whenever a person is *Unemployed* and is working age, they search for a possible job to gain employment. For a detailed explanation of this process, see Section 3.3, **Workplace Assignment**. If the person finds employment, they transfer to the *Working* state, but otherwise will remain *Unemployed*.

The *Working* state is the one that individuals should spend the majority of their life. They will ideally work until they are 65, at which time they will automatically become *Retired*. However, even during their prime working years, there is a chance they could lose their job, which means they go back to *Unemployed* and must search again for a job. For simplicity, we don't account for people lining up new jobs before being fired or quitting their jobs, so they are always released and then must search for a new career afterwards.

The *Retired* phase is a very simple one. Once a person has retired, there is no going back to work or school, so they will remain there for the rest of their life.

An additional state was included for *Dead* to indicate when the agent has died. This state is not being used in the current framework, as deceased persons are removed completely from the society, however this may be an important state in a future version that does not remove the dead agents completely, but rather just labels them as dead.

3.6.2.5 School / Work Involvement

In addition to the role states to indicate their positions, the agents must also be placed into places of school or work during those phases.

Students will need to be placed in elementary schools and possibly post-secondary institutions. Based on the agent's age and education requirements, a school of the appropriate level of education will be selected for the agent to attend. For simplicity, we assume a student will remain in one elementary school during their childhood (unless they move and have to find a school in the new hometown), and at most one post-secondary institution, just to avoid having people moving constantly between various schools. Thus, when a child is first beginning school, they are assigned a school at that time, and after that no further assignment is required until post-secondary. When they graduate from elementary (secondary) school, then a post-secondary institution will be chosen for them to attend. Some agents require University, some College, and others will be able to work without post-secondary education.

When a person is seeking work, they will search for an appropriate workplace that employs the particular career of that person's field of expertise. For simplicity, only one career is assigned to each person even if the skills may be transferrable between several jobs. For more details on the workplace assignment phase, see the **Workplace Assignment** explanation in Section 3.3.

3.6.2.6 Hometown

Although the simulation is intended to portray one particular city, individuals are allowed to move into and out of the main society. For simplicity, they will only move due to finding work in a different city, as explained in more detail above in Section 3.6.2.5. Agents who live in the local society and find work externally, and those who are living in a different location but get a job in the simulation society, will move to the new location for their work, and bring their family with them. If the person is married, their spouse will move with them, and if they have any young children (under the age of 18 with the current configuration), then they too will move with the parents.

If there is not a requirement to move for work, then agents are expected to stay in the same society they currently live in, so their hometown records are updated.

3.6.2.7 Club Assignment

At each year step in the simulation, the persons re-evaluate which clubs they are part of and whether or not they will join or be involved in new clubs. This is important so that new friendships to potentially develop from clubs that the people join.

3.6.2.8 Friendship

One of the most important components of this simulation is the social aspect, so the agents need to re-evaluate their social network at every time step. The agents will make new friends over time and accumulate the relationships. For simplicity, friendships are never lost, so the only way for a friendship to end is when a person dies. Thus, as a person gets older, they will continue to grow their number of relationships, until they get to the age where many of their friends pass away.

3.6.3 Population Generation Models

Society generations can be modelled in a variety of different models, taking one of several possible directions to create the population and network. Since the society is composed of both parts, the population and the network, some simulations begin with the population of agents and then overlay a network of relationships, while another approach is to begin with a network and then add agents to fill the required roles in the network.

Many society simulations are broad, macro-level models in that the population is essentially one entity rather than a collection of individualized affective entities. That form of approach is sufficient for pattern-based purposes, particularly for epidemiological uses [30]. However, this macro-level population approach is not well-suited to simulations that **do** involve monadic individuals. Simulations, such as those for creating human-like NPCs for video games, revolve around the individuals rather than the whole, and thus must be done from a micro-level approach. For that reason, we chose to examine different approaches to agent-based population generation.

3.6.3.1 Monadic Matching Model

One way to generate a society is to start with a population of monads, or single individuals, who are all assigned many attributes including age, gender, race, culture, career, income, and much more. Once generated, the resulting population of singles is viewed as a pool from which the individuals can be matched with one another to find spouses and friendships, deriving from Group membership (see Section 3.7.5, **Groups**, for more details). Various attributes are examined to determine relationship strength and other relationship-based measures.

3.6.3.2 Dyadic Model

Similar to the Monadic Matching Model, the Dyadic Model will start with a population rather than grow one dynamically. In this case, however, people are generated with their spouse or partner rather than starting single. Thus, this method will create a population of dyads (couples) in which the two people are created with some dependencies in their attributes. The relationship start year and the couples' hometowns are identical to the two, so those are created at the dyadic level. Other attributes, like race, religion, and age, are created at the individual level within the dyad, but one person's attribute values will influence those of their partner in the dyad. There are also many independent traits and attributes which are solely created at the individual level and have no influence or dependency on that of their partner.

Only the romantic relationships, and possibly the couples' children, are formed in this way, while all other friendship connections will be formed from the individuals' group involvement, just as in the Monadic Matching Model.

3.6.3.3 Monadic-Dyadic Model

The Monadic Matching Model and Dyadic Model can be combined in a model that generates both monads (singles) and dyads (couples) initially. This model has no new concepts or fundamental modifications. It is simply a merger of the two contributing models, so the population will be comprised of a pool of single individuals who are looking for a relationship and a series of couples in relationship.

3.6.3.4 Genesis Model

Inspired from the biblical account, only one couple is created and through many generations of reproduction, the society population grows eventually into a large society all stemming from that original created couple. Since the society will only contain people born from the original couple, everybody will be considered family for a few generations until there is enough spread in the society. To accommodate this kinship in the society, the agents are permitted to mate with their family members initially, just as in the Genesis account. A threshold is put in place to dictates how long before the inter-family marriage and reproduction is prohibited. After that generational threshold is met, there will be enough spread in the population that people will no longer need to marry people that are closely related. For simplicity, we only consider siblings as family when checking for inter-family mating. Normally, we would also want to check if the potential partner is the individual's parent or grandparent; and ideally whether they are within the extended family. For resource management purposes, we try to minimize the number of unnecessary conditionals, and for this feature, we were able to rationally omit many of the other kinship connections. People will look for a partner around their own age, so this eliminates the need to check if their potential partner is their parent or grandparent, since the generational gap will be greater than their allowance for the partner's age. The extended family, however, is omitted without rationalization, just to simplify the procedure.

This Genesis Model **requires** an active simulation for its development. While the previously mentioned models can include an active simulation to help their growth, it is only truly required in this model.

3.6.3.5 Ego-Centric Model

So far, we have only discussed bottom-up approaches for generating an artificial society. The Ego-Centric Model follows an entirely different paradigm, as it takes the top-down approach, and grows the society outwardly from one initial person. The primary person, or the *ego*, is created and given a profile of attributes and traits. From this ego, any and all required connections are generated and agents are created ad hoc to fill those positions. The network continues to grow outwardly as the person requires specific relationships. Optionally, the others in the society could form relationships with others that don't directly involve the ego.

3.6.3.6 Summary of Simulation Models

The aforementioned simulation models are summarized below in Table 3.1.

	Monadic Matching	Dyadic	Monadic-Dyadic	Genesis	Ego-Centric
Requires Active Simulation				×	
Top-Down					×
Bottom-Up	×	×	×	×	
Significant Initial Population	×	×	×		

Table 3.1: Summary of simulation models.

3.7 Network

The network is the set of agent-to-agent connections between the individuals in the population, including family member relationships, marital and intimate relationships, and friendships. These different interpersonal relationships are the basis of this entire society, for without them, there is just a population of individual entities. These friendships and relationships are an important part of transforming those individual nodes into realistic, human-like agents. This is the component of artificial societies that is typically omitted and only a small number of pre-scripted relationships normally exist in games that require any form of relations. In this simulation, relationships are a vital portion to keep the society connected realistically.

3.7.1 Network Size

In the interest of creating a realistic society, we must aim to create a social network that reasonably matches an actual network in its size and level of connectivity. To create an overall statistically realistic network, it boils down to ensuring the integrity of the individual agents' personal networks.

Kadushin reports on several researchers' theories on individual's social circle sizes in [27]. The most common estimate from the various studies finds that a typical social circle has a mean of about 150 connections; however a large standard deviation is attached to that, so the size drastically changes from person to person. An individual's social network size is greatly affected by their personality.

By its nature, the level of extraversion plays a huge role in the person's ability to approach others and thus to form friendships [31]. We reflect this personality factor in the calculation so that those who are completely introverted have a smaller likelihood of making friends and those with complete extraversion have a higher likelihood. When agents are calculating the probabilities of becoming friends with each of the others they encounter (see the **Potential Friends** explanation in Section 3.7.6 for more details), the probability is scaled according to a function of the primary person's level of extraversion.

Selfhout et al. examined other traits in addition to Extraversion in relation to friendship selection. They used the Big Five personality model (Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism), rather than the Myer-Briggs model. The Extraversion trait is common to both models, and coincidentally, their research suggests that Extraversion is the only trait that impacts individuals' friendship selection [31].

3.7.2 Kinds of Relationships

People form relationships with others of many kinds and of varying degrees of closeness. Each relationship in this simulation is categorized as one of three relationship types: partner (romantic), kinship, or friendship. A more advanced simulation may also include specific indicators for best friends, co-workers, acquaintances, etc. rather than grouping all friends into one general class. It could also be handled as a scale classifier with primary, secondary, and tertiary groups of friends. Alternately, or in addition to the classes, each friendship can be given a measure of strength to indicate the closeness and the frequency of the pair seeing or communicating with one another. This simulation framework was designed with a place for such a friendship strength measure; however the strength is not calculated in the current simulation for non-romantic relationships. Only the couples in romantic relationships have an actual calculated strength. The three relationship types are explained in more detail below.

3.7.2.1 Romantic Relationships

Marital or other intimate relationships – engaged or just dating – are important not just for putting individuals into a dyad for their own sake, but also for the reproduction of agents. Based on the generation model, these couples may be formed directly by creating a pair of pre-connected agents, or the relationship may be formed later from a pool of single individuals. Either way, a number of people in the society are linked with an intimate partner.

3.7.2.2 Kinship Relationships

Kinships are family-based relations including parent-child and sibling-sibling connections. It is assumed that each family is completely interconnected so everyone within the family is connected to everyone else within the family through one of these kinship relations (except the parents who are connected by the intimate relationship).

People cannot form romantic relationships with their kin, except during certain circumstances. The primary reason for even allowing inter-family mating is so that the Genesis model can exist, since it begins with one couple and their offspring must expand to fill the society. A threshold parameter will govern how many years into the simulation the mating within the close family members must stop. For the other model types, this threshold is kept at the initial year of the simulation so that there will never be romantic relationships formed between family members.

3.7.2.3 Friendships

As in the real world, people require more than just their own family and need social lives with various friends. These friends may be current or former schoolmates, co-workers, neighbours, peers from a temple or place of worship, or others involved in various clubs or extra-curricular activities. A person's friendships are assumed to come from one or more of these circles or activities (see Section 3.7.5, **Groups**, for more details.) Each individual will have some number of friends, which will be based on the person's level of extroversion and the number of activities the person is involved in. For example, a person who is very extroverted and who attends several clubs and extra-curricular activities will likely have many more friends than an introverted person who does not go out often.

3.7.3 Clubs

A "club" is a generic term for any form of activity, hangout place, or extracurricular event in which people meet on some kind of regular basis; and does not include any of the pre-existent types of circles (family, school, workplace, temple, etc.). Clubs may be extensions of one or more of these other circles, but cannot be the same sets. For example, there is a club called "London Musicians Jam Fest" in which one group of people who may join are those who work in "Mussein's Music Store" - one of the workplaces in the society - but the music jam club is different than the music store workplace. Other examples of clubs include chess club, drama club, gyms, bars, an elderly card game group, and stargazers group, among many others. People are put into clubs based on their personality traits and other attributes, and the clubs' corresponding trait set. For more information on the club selection process, see Section 3.7.4, **Trait Matching**.

3.7.4 Trait Matching

Trait matching is a process for calculating the probability that a person is put into one or more groups based on how well the person's attributes match the set of trait criteria for each of the groups. This is an important procedure in placement assignment. Each possible component has a list of traits or other personal attributes that are well-suited for that component. These include the MBTI personality traits, age range, sex, race, religion, relationship status, and additional attributes: athleticism and intelligence. A person who possesses the traits listed with a specific component has increased likelihood to be assigned to that component. In some cases, a trait is labelled as *required* to indicate that the person **must** possess that attribute to be assigned to that component. That is, if a person does **not** possess that required trait, then they are automatically disqualified from assignment to that component.

As an example, consider the career option, Priest. From the MBTI traits, a priest is likely to be Feeling rather than Thinking, so the Feeling attribute is included in the list of indicative attributes. Aside from that, the Introversion/Extroversion, Intuition/Sensing, and Perceiving/Judging traits are inconclusive, so all those options are included in the indicative traits. In addition to the 7 MBTI personality traits, the list also contains the required trait of religion being Catholic. To clarify, any person who is not labelled as Catholic in their religion cannot qualify for this position, while those who are labelled Catholic will qualify but then a randomizer will be used, so only some of those will become priests.

Another example is a police officer. Again, there are a number of MBTI traits that are attached to the police officer role, but additionally, it is required that a police officer has an athleticism value of at least 0.5. Again, this means that anyone with less than a 0.5 level of athleticism is automatically disqualified from being a police officer.

Once all the possible components (careers, clubs, etc.) have been examined for the given person, they can then be assigned to one or more of those components from the probabilities calculated from this algorithm.

3.7.5 Groups

In the real world, nearly every friendship between two people begins by some kind of commonality between the two, whether they attended the same school at the same time, worked together, or met at an extra-curricular activity (spiritual or religious group, games clubs, sports, etc.). It is fairly rare that two people become friends randomly without being involved in some kind of activity together or living in close proximity to one another. For this reason, the society simulation uses **groups** from which friendships are formed. Each group represents some form of community, from a school, workplace, organized religion, or club. If this simulation contained a geographical component, where the agents were given houses within a city or region, then each neighbourhood would be put into a group as well, since people are likely to become friends or acquaintances with people who live near to them. This is not part of this simulation, but a future consideration.

The groups are organized into a tree structure to allow for quicker access to nodes. The top node of the tree is a generic “All Groups” node. Its children nodes are the various categories in which the groups arise from. Those category nodes include “Schools”, “Workplaces”, “Religious Communities”, and “Clubs”.

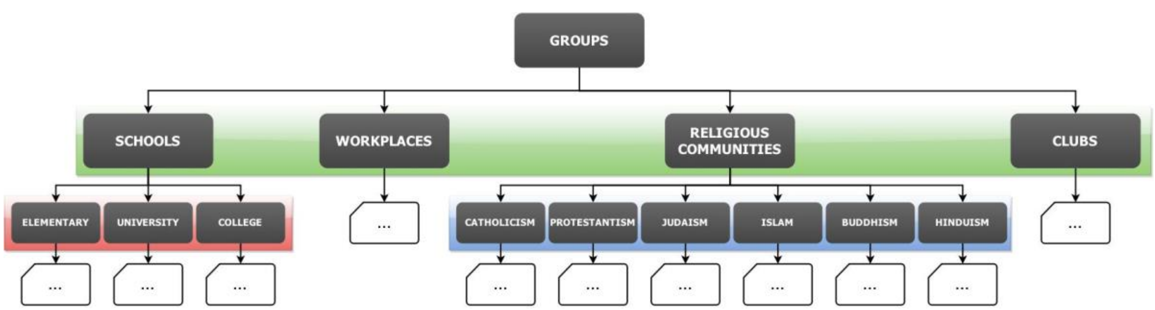

Figure 3.12: Tree structure for set of Groups in the society.

The schools node is then comprised of a sub-tree with “Elementary”, “University”, and “College” sub-categories.

The religious communities node is also a sub-tree with each of its nodes representing the various religions (i.e. Judaism, Catholicism, Islam, etc.). Each of those religions represent a Group (i.e. the Catholic population of the society), and each is also a sub-tree containing the various temples under that religion. For example, the Catholicism node has the following children: *Catholic Church*, *St. Mary’s Catholic Church, St. Thomas Catholic Church*, *Mother Teresa Catholic Church*, and *St. Anthony Catholic Church*.

A tree diagram is presented in Figure 3.12 representing the top levels of the group tree structure. All leaf nodes are omitted from the diagram so that the main structure can be clearly and neatly displayed.

The workplace and club nodes are not broken down into further sub-trees. From each of the nodes that are currently leafs, every individual group is added to the appropriate node. Some of these groups are added singularly but most are added as **timeline groups**. Timeline groups are those that exist for multiple years. The group must distinguish itself from one year to the next, so a series of groups from $year_{min}$ to $year_{max}$ is created. For example, consider an elementary school, generically labelled ‘School A’, which began in 1900 and is still active in 2015. Then, under the Schools node, and then under the Elementary sub-category, there will be added “School A” with groups “School A (1900)”, “School A (1901)” …, “School A (2015)”. Thus, a student who attended School A in 1930 will not be in the same group as a student at School A in 2015.

The timeline groups in the school sub-tree are important so that two students from different eras are not considered schoolmates and thus potential friends, as mentioned above. However, limiting each student to just their own year implies that students from a year above or a year below the student in question would not qualify as schoolmates. In this simulation, a delta year parameter allows students to be considered schoolmates with the students within [$year_{student}$-$year_{delta}$, $year_{student}$+$year_{delta}$]. We chose to use a delta of 1so students from the school within 1 year are potential school friends. For example, a student in School A in 1995 should be considered schoolmates with the students at School A in 1994 and 1996 as well.

3.7.6 Potential Friends

Each agent contains a list of potential friends, along with their respective friendship probabilities and the relationship roles (parent-child, teacher-student, friend-friend, etc.). Other agents are added to this list from being in shared groups (as explained in the **Groups** discussion in Section 3.7.5). All people who share in any groups with an agent are put into this list of friend candidates. They begin with a low probability of becoming friends, but the more groups they have in common, the higher their friendship probability will be. The probability accumulates, so agents who are in several of the same groups will have a high probability of becoming friends. The group size also inversely affects the probability of two people becoming friends from that particular group. For example, consider two arbitrary persons in the society. The two are much less likely to become friends if they are both in a large group, such as a university campus with 5000 students, compared to being in a bowling club group of 10 people.

As mentioned above in Section 3.7.1: **Network Size**, the probability of a person forming a friendship with another agent is scaled by the first person's level of extraversion, in addition to the number of members in the shared group. The probability of person A and person B becoming friends solely from their involvement in group G is calculated as:

$$p_{friendship(A,B,G)} = \frac{\rho\kappa}{|G|}$$

$\kappa = 0.5\Phi + 0.75$, Φ = A's Extraversion, $|G|$ = number of members in group G

3.7.7 Friendship Selector

During the friendship selection phase, each agent's list of potential friends is examined. The probability of each candidate is directly used as thresholds with random numbers to determine whether or not the two become friends at this point. Thus, those who have high probabilities will likely become friends while those with low probabilities will not, however because of the randomness, even those with low probabilities may become friends and those with high probabilities may not become friends. Each agent examines every one of their candidate friends in this way to create their friendships. Every friendship is reflexive so as soon as one person befriends another, they form a mutual friendship.

3.8 Summary

This chapter covered a lot of details of the components, processes, and models for generating an artificial society. We began by defining the atomic unit of the society, the agent, and the various attributes associated with it, from personality traits and current status information to their historical records and social relationships. We proceed to explain the agents' involvements in their different formations (monads, dyads, and groups), their workplace assignment process, as well as romantic relationships between agents. We then take a step back to explain how the agents are created, which comes with slight variation depending on the agent formation or type. Next, we discuss the different models of society generation which range from bottom-up approaches, including the Monadic Model, Dyadic Model, and Genesis Model, to a very different, top-down approach, namely the Ego-Centric Model. Lastly, we describe the social aspect of the population, describing the different kinds of relationships and the processes for group involvement and friendship selection.

Chapter 4

4 Implementation

In light of the exciting new opportunities that artificial intelligent societies introduce into video games, we wanted to address this largely untapped field and examine the possibilities when creating an artificial society. This chapter describes the prototype for generating artificial societies, based upon the design explained throughout Chapter 3.

We first introduce the implemented framework generally in Section 4.1. The following two sections provide details on the development, and the use of external configuration files, respectively. We then wrap up this chapter in Section 4.4 with a description of the approach we use and the kind of society we generate through the prototype system.

4.1 Prototype

For this research, we developed a prototype artificial society generator following the design model proposed in Chapter 3. The simulation is considered a framework as it has a set procedure and numerous parameters from external files; however it is developed in a black-box fashion such that the various steps and sub-steps of the procedure are modular routines that can be modified or replaced by new modules for case-specific needs. Many configuration parameters are stored in XML files, so there are a number of population types that can be created simply by changing those parameters in the files. If one requires a large change in the way the population is created, then the code would likely have to change since parameters will not handle those structural changes. Again, the methods were intentionally created modularly so that one could swap out a function and put in a replacement and with just a few small modifications, the program should work properly with the newly added functions.

4.2 Implementation Specifics

The implemented simulation for this research was programmed in Java version 1.8, using the Eclipse IDE. This implementation was programmed and run on the Windows 7 OS.

4.3 External Files

There are several external configuration files for the simulation program, all of which are formatted as XML. The main configuration parameter file stores all the parameters relating to the population composition; demographics, probabilities of various events, distributions of various events, statistical limits (mean, minimum, maximum, standard deviation) of various events, and several other population-governing factors. In addition to that primary file, there are XML files for specific lists of entities in the society. These include files for schools, post-secondary institutions, religious temples, careers, workplaces, and clubs.

The XML files can include mathematical distributions to indicate how values should be selected for a particular parameter. The distributions are stored as strings of the format *distr*(σ_1 ,σ_2 ...), where σ_i is the i^{th} distribution parameter, which are then parsed by a DistributionParser method in the program. For example, the parameter representing the number of children that a married couple has, is defined as normal(2.0, 1.4) so the number is selected from a normal distribution with a mean of 2.0 and variance of 1.4. There are supplemental limits to ensure the selected number lies within a valid range.

4.4 Micro-Level Artificial Society

The purpose of this simulation is to create a population of agents. Many artificial society simulations work in the macro-level, mainly focused on population and subset attributes like size, composition, and connectivity. In this research, we are developing a model for micro-level agent-based society generation, in which the population itself is not the focus but rather than the individuals within it. In the presented model, the population is composed of individual agents who are modeled after a simplified human.

4.5 Society Visualization

In addition to the primary society generator that was developed for this research, we created a stand-alone visualization program to display the generated networks. This visualization tool was also programmed in Java, and uses the JUNG (Java Universal Network/Graph Framework) library for displaying the network visually.

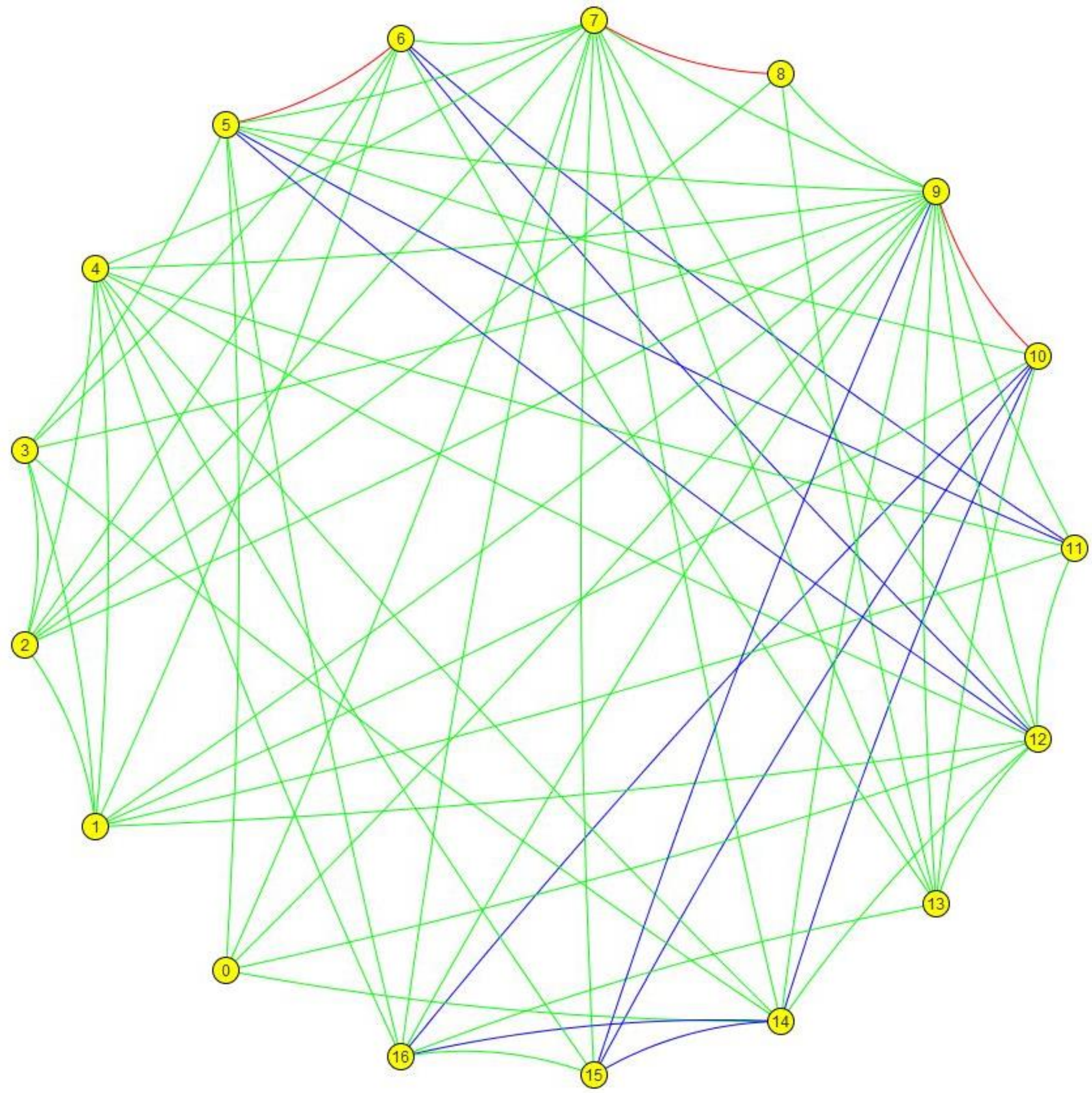

Figure 4.1: Small society network in circular layout.

The visualization tool provides several layouts for displaying the given social network. The network in Figure 4.1 is a circular layout of a small society of 16 agents in their relationships. Red edges represent marriages; blue edges indicate family connections (i.e. sibling-sibling, parent-child); and the green edges are for all other forms of friendships. The same small society is displayed in Figure 4.2 using a Fruchterman-Reingold layout for a force-directed graph (included in the JUNG library). The program was also designed to focus on a selected node's individual closed network, as shown in Figure 4.3.

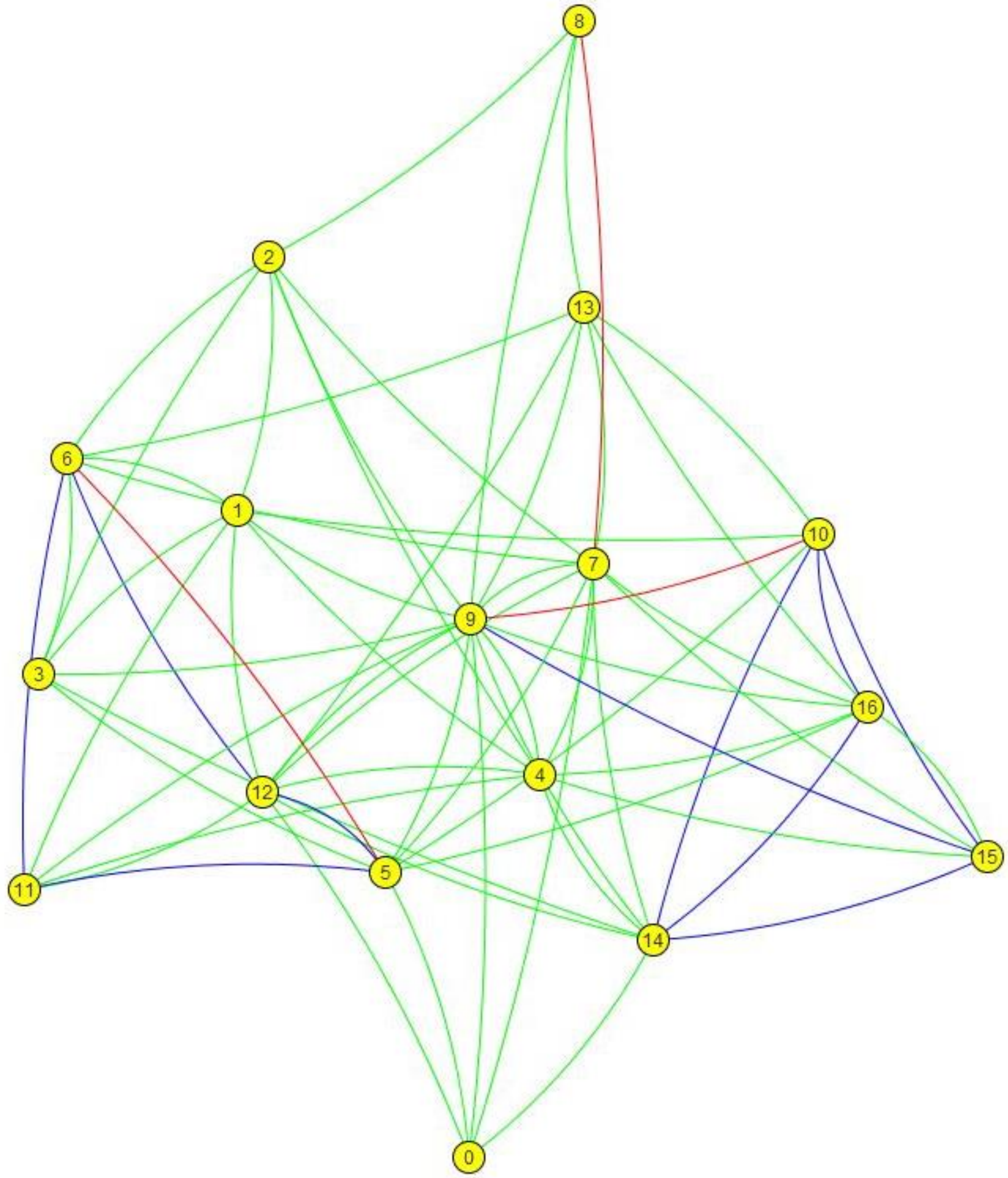

Figure 4.2: Small society network in Fruchterman-Reingold layout.

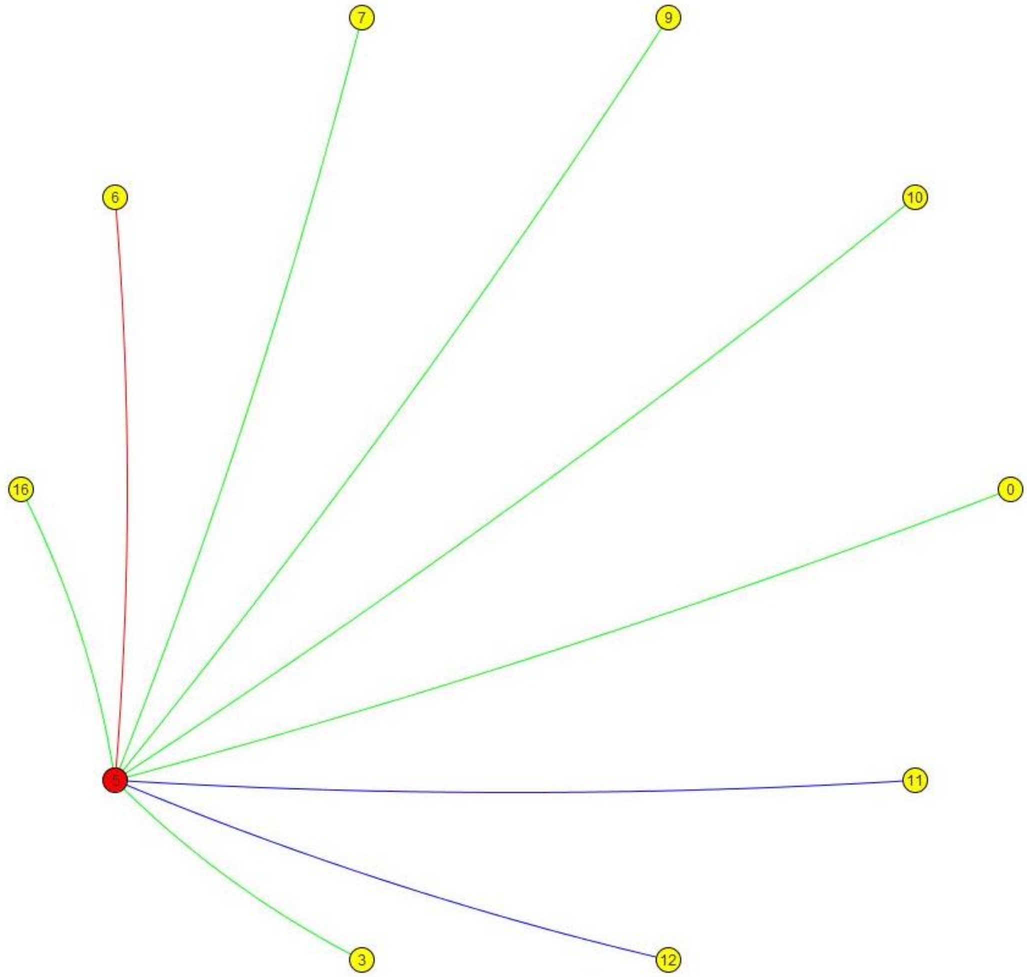

Figure 4.3: Closed social network of one focus agent.

Chapter 5

5 Experimental Results

We began with three questions around which our research was formed. Our implementation was the primary medium for addressing the foundational questions.

5.1 Question 1

Can we make an artificial society?

5.1.1 Discussion

The answer to the first question is simple. We created a program for this research that generates an artificial society. To be more thorough we must revisit the definition of an artificial society. An artificial society is a virtual system consisting of two vital components: the population of agents, and the social network connecting those agents. In our simulation framework, we generate a number of agents and we create a social network of relationships formed from the circles and groups in which the agents participate. Therefore, we have indeed created an artificial society.

5.1.2 Threats to Validity

We considered the definition of an artificial society as a population of agents in conjunction with a social network. Our work focuses on fulfilling those criteria to generate a society; however, artificial societies may be defined with additional parameters or components. The authors of [12] describe an artificial society simulation with three required components: agents, environment, and rules. Now, the rules that they refer to are commonly used to create social connections and communities among the agents. Although they may look different from those in Epstein's work, we do in fact use rules in our simulation to govern the formation of the society. Our simulation uses various algorithms with stochastic elements to determine the population and network, while Epstein uses a cellular automata approach to rules for interactivity and evolution. Still, by his definition, our simulation needs an environment in which the agents would dwell and interact with one another. We acknowledge this missing component, but argue that it is not a necessary component for our research purposes. The focus of our work is to generate a society, rather than to animate or play out a society. Additionally, it would be straightforward to embed our generated society into a virtual world for a video game, which is part of the future work (see Section 6.4.5).

Aside from the definitional differences of a society, we recognize that although we are able to generate an artificial society, there are limitations on the society. Due to the large amount of agent-based information and the agent-agent relationship formation process, the simulation execution time becomes noticeably slower when the population reaches about 500 people. The overhead and resource consumption of this process is explained in more detail below in Section 5.3 on **Question 3**.

Experimental results show that in some cases, the produced societies' compositions are unpredictable and erratic. In Section 5.4 the evaluation results are presented, and they show that the Monadic model and the Genesis model are particularly unstable.

5.2 Question 2

How well can our society "look" like a realistic society?

5.2.1 Discussion

We begin the discussion on the realism of our society by re-iterating four key points.

Firstly, the current work is a simplification that focuses on a relatively small subset of a real society's elements. Thus, the level realism of the virtual society will vary between different aspects, depending on how much we focused on each portion.

The second important point is that many of the society's elements are parameterized from configuration files. We are proposing a framework for a society generator, and not positing that this is a comprehensive system with the utmost realism. Many concepts and statistics were borrowed from psychological and sociological literature to help model the artificial society after the real society; however, there were also several parameters that had to be estimated or evaluated through trial and error.

We also point out that the purpose for this research is to create a society of NPCs for video games, which also solidifies the need for having a parameterized system. We developed a framework that generates societies of a wide variety of sizes, forms, demographics, and looks. Game developers who would use a framework like this would be in control of the society to meet their needs, which may or may not look like a real society. Even in the cases when they do require a realistic society, they would be responsible for optimizing the parameters to suit that need.

Finally, this is the first system of society generation of its kind, so there is no basis for comparison from the literature. This makes it difficult to evaluate the realism of our society since it has not been done before for a society like this.

Three methods of evaluation have been employed to measure specific aspects of the society, which are presented below in Section 5.4. These evaluation approaches are used to determine the validation and verification of the generator system, but the results are transferrable to indicate the realism of the produced society as well.

5.2.2 Threats to Validity

As prefaced in the above discussion, the generated society is modelled after a real society only in a small number of aspects. A subset of parameters and features were incorporated into the current work to provide an introductory generator framework. A real society has uncountable factors, features, and aspects that govern or represent its composition. Although we were not striving to achieve a perfectly realistic society, this is certainly a threat to the validity of our work; that what we generate does not resemble a realistic society. This threat is acknowledged, and we recognize that there are many opportunities for future work to improve the realism of the society being generated.

5.3 Question 3

Can the society be generated in real-time or production-time? What is the overhead in creating a society?

5.3.1 Discussion

The overhead induced by generating a society is certainly an issue for developers trying to implement such algorithms into video games. This is especially problematic in creating large populations, but even in the current simulation, we begin to observe noticeable slowdowns with populations of about 500 agents. Some features within the simulation will cause a linear growth in time with respect to population size, while some cause quadratic growth with the increase of population sizes; so there is an overall major increase in execution time for bigger populations. Figure 5.1 displays the execution time for generating societies of various sizes, compared to the time to load in the society from file. As clearly shown in the scatterplot, the generation times become drastically slower as population size increases, while the loading times are almost consistently low.

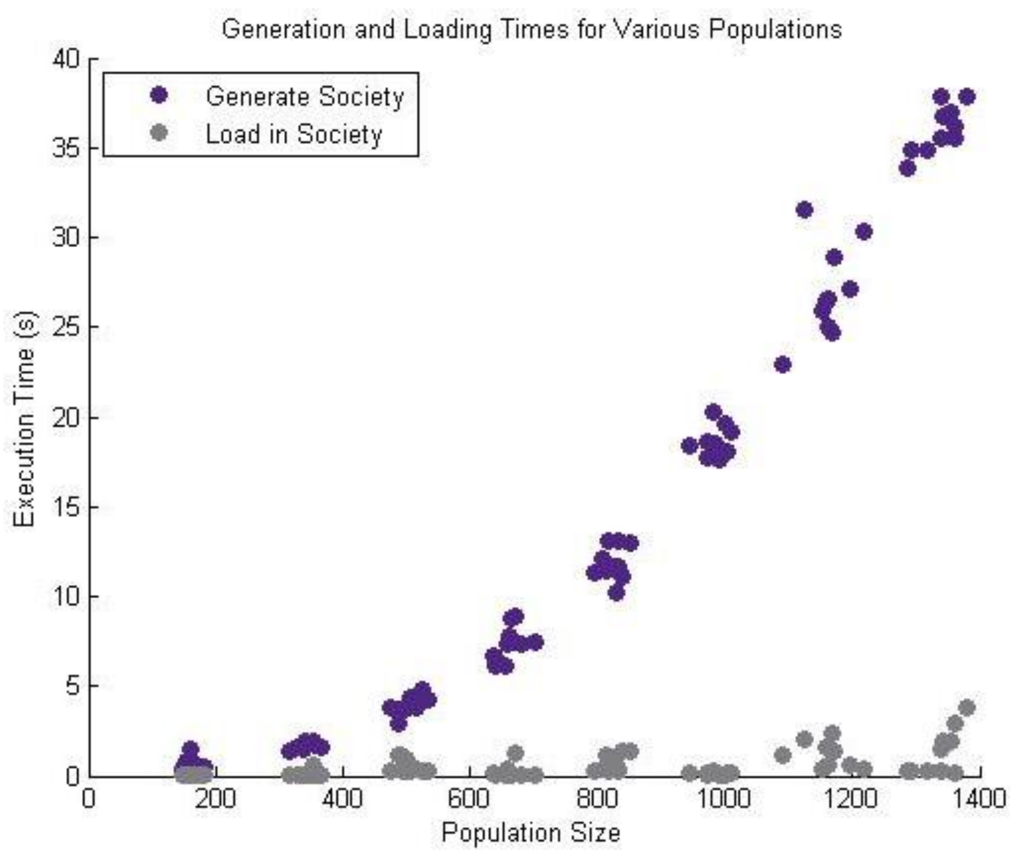

Figure 5.1: Execution times for generating and loading societies of various sizes.

In cases where a large population is required, developers would be encouraged to generate the society during the production phase so the game would read it in from file rather than having to perform the lengthy generation at run-time. On the other hand, games may be designed such that the population is new and dynamic for each play through the game, and then a real-time society generation may be necessary. Developers will have to weigh out their options for each game's needs to provide an optimal solution for the society generation.

The memory is another overhead restriction with increasingly large populations. Saving and loading in the society cannot alleviate the memory constraint the way it can for the timing. The memory usage for varying population sizes is displayed in Figure 5.2 below, and show a drastic growth in relation to increasing population sizes.

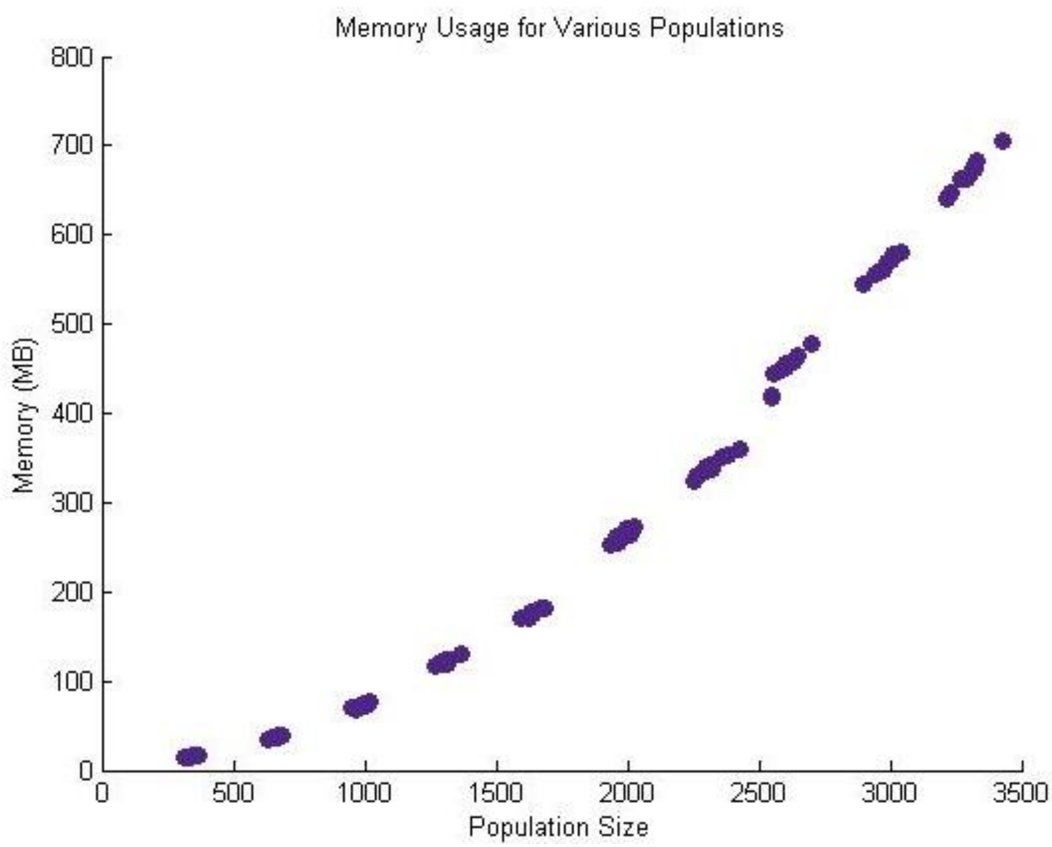

Figure 5.2: Memory usage for generating societies of various sizes

In addition to the generation overhead, we evaluated several simple queries on the society. We came up with seven such queries that would likely be called frequently within a society-based video game. The queries were timed separately and the results are presented in Figure 5.3. Because these queries are very quick to execute, each one was run 10 times consecutively for each trial, and there were 10 identical (no parameter change) trials. The queries are as follows:

1. Add new agent to society.
2. Add new couple (dyad) to society.
3. Kill off random agent from society.
4. Get race distribution from society.
5. Get total number of friendships in society.
6. Get number of agents living locally.
7. Get number of single friends of a random agent.

The graph indicates that query 3 (killing off a random agent from the society) took increasingly longer for larger populations, while the other queries had no significant changes in relation to the population size. Some have a slight increased execution time (see query 5 and 6) and one even has a decreased execution time with relation to an increasing population size, but in either case the trend is insignificant. Query 3 likely yields the drastic time increase because this query involves removing an agent completely from the society which means every agent in the society must ensure they have no social connections to the killed off agent. Thus, as population increases, there are more agents to evaluate possible connections to the deceased agent and so the overall query time increases.

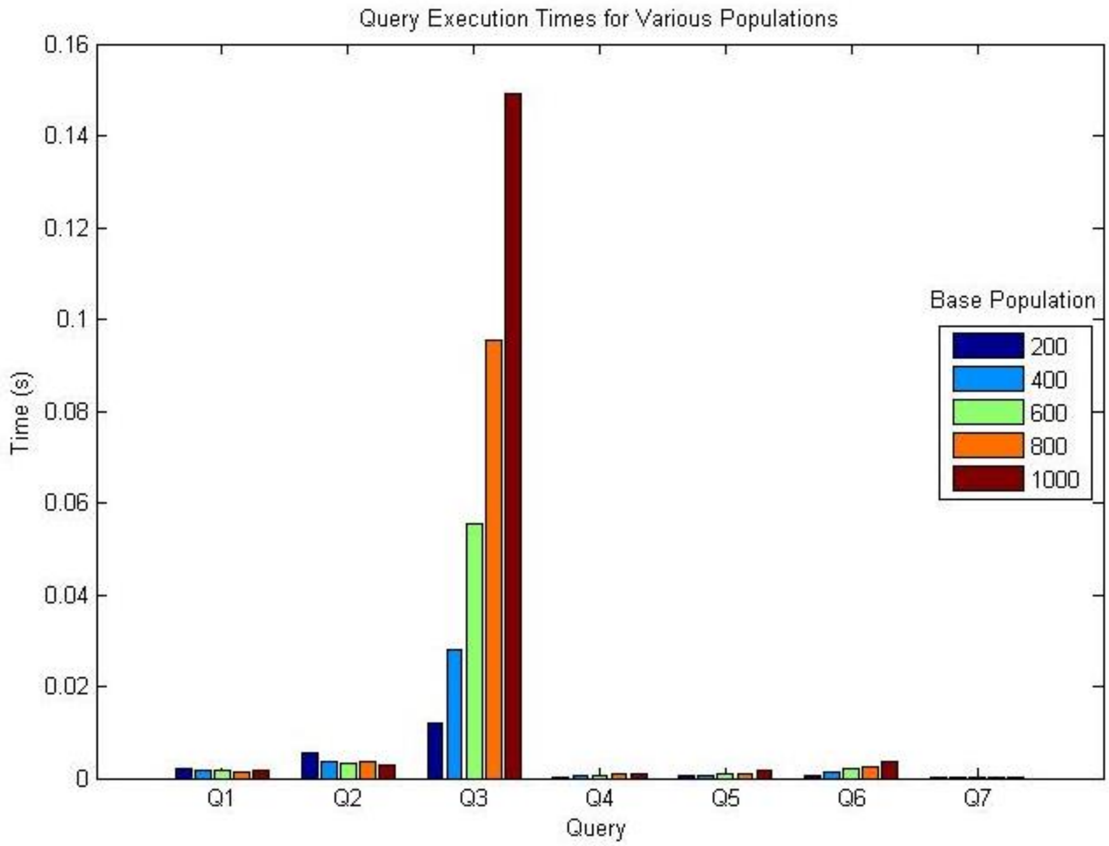

Figure 5.3 Execution times for various simple queries on societies of varying sizes.

5.3.2 Threats to Validity

As population size increases, the generation time increases drastically which becomes especially problematic when generating a large-scale society. In cases where a video game requires a large population, say 1,000,000 people, the generation time will be unreasonably high. In those cases, the developers will have no choice but to generate the society during in-house and just let the game load it in from file; which would disallow the possibility of dynamic populations in the games.

5.4 Evaluation

In order to determine the degree of realism of a society, we must use validation and verification techniques that measure the integrity of the model. The authors of [32] describe several such techniques for determining the credibility of agent-based simulations. We employ some of those presented techniques for our simulation to evaluate how well our generated society looks like a realistic society.

5.4.1 Graphical Representation

The Graphical Representation technique is a more subjective one rather than quantifiable. For this, we graph the output values of particular variables to visually analyze the data produced from the society. These graphs could be compared to sample data distributions where applicable, and others could be examined as functions of tweaked parameters to indicate the effects of the specific parameter on that variable. We compared the artificial society's racial distribution with the given distribution from file. Figure 5.4 shows the similarity between the artificial society's actual racial distribution and the given racial distribution from the configuration file.

In Figure 5.5, we illustrate the effect of the children distribution, in how it impacts the fate of the society. The two cases begin with the same base population – the same number of married couples – and are both run for 30 years. The purple line represents the one-child policy, as has been tried in China, in which married couples can have at most one child. The gray line represents a regular or free distribution, estimated to match a typical North American distribution. It takes a normal distribution with $\mu = 2.0$ and $\sigma^2 = 1.4$, and is then clipped to [1.0, 5.0] which means any values out of that range will be assigned the value of the closest interval limit. Married couples can also have 0 children in this case, but a separate parameter is used to determine whether or not they have children. This line graph shows the vast difference between the two children distributions.

We also wish to depict the effects of the agents' friendship factor parameter, ρ, on the overall connectedness of the society. We use our visualization tool to display three societies of equal size, but with varying ρ values. These are displayed in Figures 5.6, 5.7, and 5.8 below. With $\rho = 0.1$, the network is quite sparse with only a small number of connections between the agents. This changes significantly when the value of ρ is increased to 0.5 and the network becomes more filled in. In the final society, for $\rho = 0.9$, the network is more dense as indicated by less visible white sections that instead contain many more agent-to-agent connections.

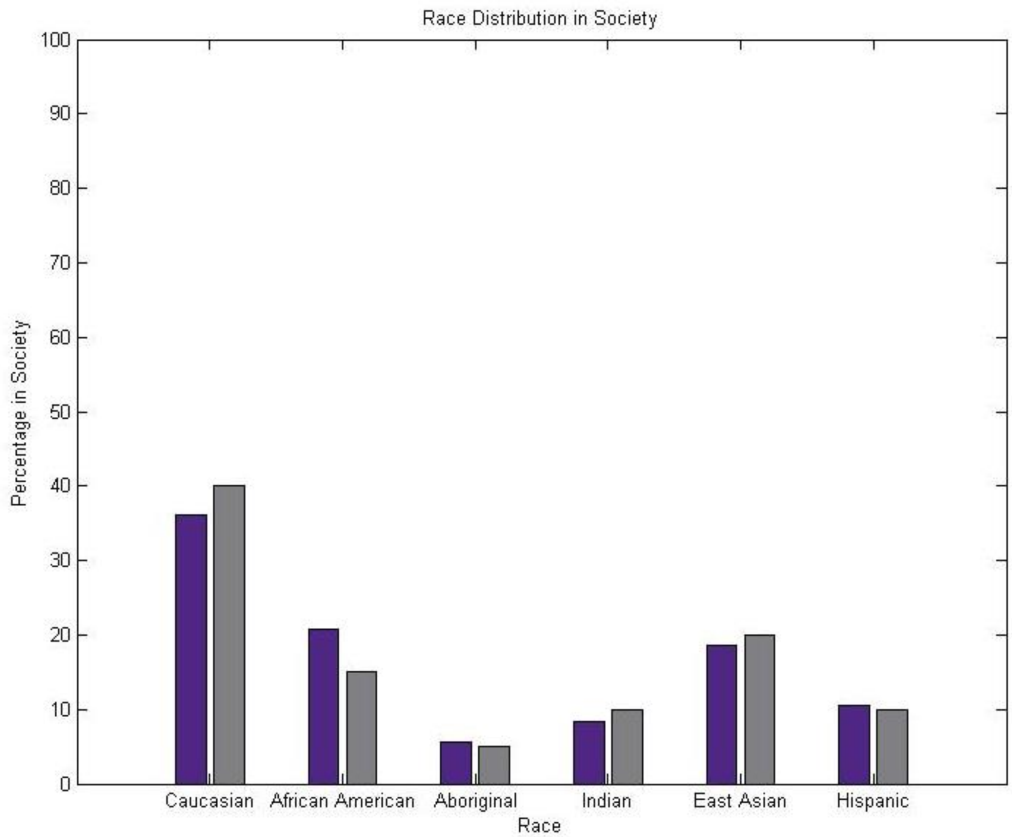

Figure 5.4: Distribution of racial demographics in the artificial society.

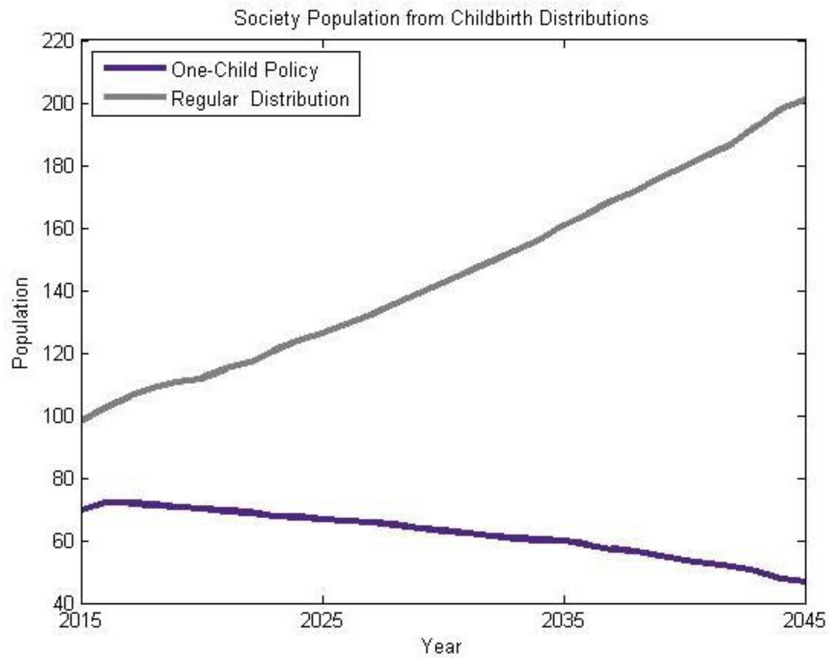

Figure 5.5: Society population over a 30 year simulation for a one-child policy and a regular distribution

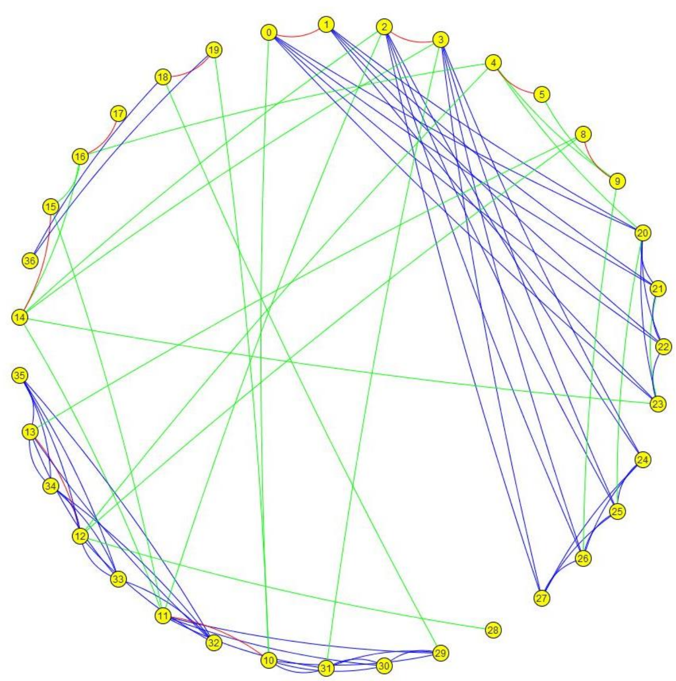

Figure 5.6: Visualization of small society network with ρ = 0.1.

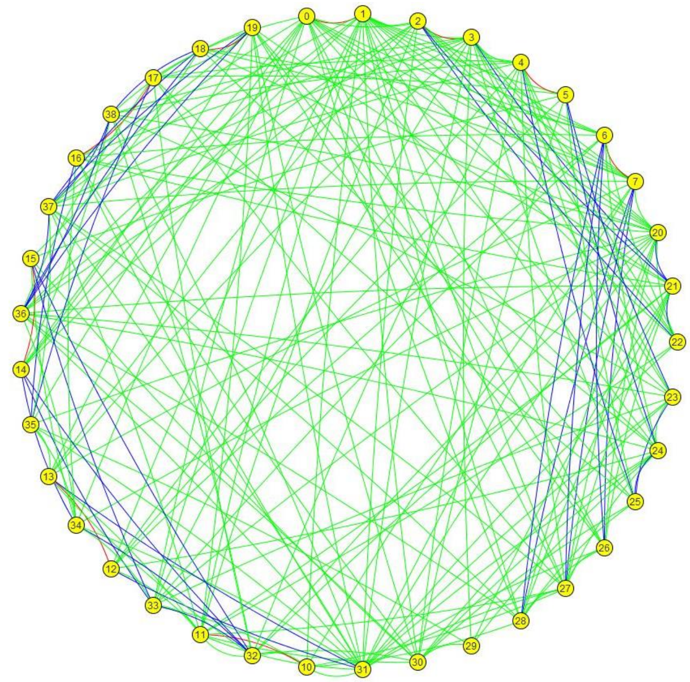

Figure 5.7: Visualization of small society network with ρ = 0.5.

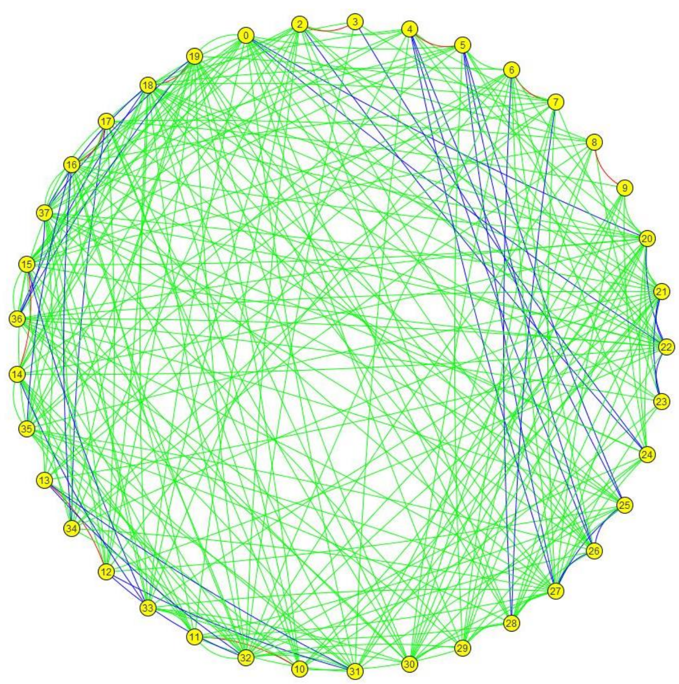

Figure 5.8: Visualization of small society network with ρ = 0.9.

5.4.2 Internal Validity

The internal validity of a stochastic simulation is the measure of its consistency. This is determined by running several trials and comparing the results between them. Ideally, the model should be relatively consistent from trial to trial. Of course there will be some variability especially with complex systems like this that heavily rely on stochasticity, but an integral model should yield reasonably consistent results. Now, in a simulation like this, the notion of "results" is vague and ambiguous. For these tests, we will select specific variables in the society to represent the results that are being examined.

For each of these tests, we ran 100 identical (no parameter changes) trials and calculated the mean and standard deviation from those trials. The results are presented below in Table 5.1. All of these trials use a base population of 30 married couples. The first query is the total population after evaluating the society with this fixed base population of 60. Elderly people may die, and married couples may have children, and thus the final population will diverge from the base population. The second query looks at the connectedness of the society, by counting the total number of relationships. Since all forms of relationships are reflexive, the final tally is halved so that we count each friendship once rather than twice. The third query for the internal validity tests analyzes the percent of agents in the entire population who are currently living in the local society. Since agents will move in and out of the main society, this number fluctuates constantly.

	Mean ± Standard Deviation
Total population from fixed base	98.0900 ± 7.9172
Total number of friendships	331.8900 ± 45.7259
Percentage of population living locally	0.4082 ± 0.0778

Table 5.1 Various query results for internal validity testing.

These mostly large standard deviations reveal the system lacks consistency. This is likely due to the system's complexity and large amount of stochasticity, which causes a lot of variety in many of the produced society's elements.

5.4.3 Model-to-Model Comparison

The Model-to-Model comparison is just as it sounds. It involves comparing the results from the various model types to determine how well they match with one another. We examine the population size of the societies produced by each model. The Monadic Model begins with 50 single adults; the Dyadic model begins with 25 married couples; the Monadic-Dyadic model begins with 20 married couples and 10 single adults; and the Genesis Model of course begins with one married couple and is then run for 50 years.

	Population (Mean ± Standard Deviation)
Monadic	50.0000 ± 0.0000
Dyadic	83.6000± 6.6513
Monadic-Dyadic	75.3000 ± 5.4599
Genesis	338.3000 ± 292.1938

Table 5.2 Population sizes from the different generation models.

Each of these models was run 10 times and we calculated the mean and standard deviation of the resulting society sizes, as displayed in Table 5.2.

The Monadic model yields a consistent population identical to that which was created initially. This is not the intended result, as there should be agents forming relationships and building families, as well as older agents dying off so that the population varies from trial to trial. Because of these erratic results, we performed some post-trial diagnostics on the society generation. It was observed that these single agents have trouble finding partners with whom they are compatible, particularly with regards to age ranges (i.e. they cannot find a mate around their age range). This lack of relationship forming means that they will not get married and have children, and thus the society is consistently stagnant.

The Dyadic and Monadic-Dyadic models present reasonable results with dynamic society sizes, caused by those dyads having children together.

The Genesis model was very unpredictable, in which many trials became depleted quickly or at least seemed headed in that direction, as suggested by low population sizes (under 10 people) after the 50 year simulation. Other trials with the Genesis model produced societies of varying sizes, as high as 807 people. The large standard deviation testifies to the unpredictability.

5.5 Summary of Results

Several evaluation techniques have been performed and presented in the previous discussions, but we will summarize the results here. It must be re-iterated that there is no basis for evaluating a society like this currently, as this is a new field of research.

Graphical representations allow us to visually analyze certain aspects of the society. Figure 5.4 reveals that the racial composition of the produced society closely follows the sample race distribution from the configuration file. The impact of the childbirth distribution on the fate of the society is shown in Figure 5.5. In Figures 5.6, 5.7, and 5.8, the society connectedness is shown to increase in conjunction with the ρ parameter.

The system's internal validity is measured by its consistency over several trials. Table 5.1 presents the results of three distinct queries on the society, which yield quite large standard deviations. We argue that the inconsistent results are due to the complexity and large amount of stochastic functionalities used in the generation process.

Finally, we compare the four implemented models to evaluate the system. We conclude from the results that the Monadic and Genesis models are erratic and require additional parameters to stabilize them. The Dyadic and Monadic-Dyadic models show more promising results with more consistently reasonable population sizes.

Chapter 6

6 Concluding Remarks

In this final chapter of the thesis, we conclude with a summary of what was researched and what discoveries were made. We then explain how this work contributes to the field of study, and then describe many opportunities for future work that build on this research.

6.1 Conclusion

The purpose of this thesis was to address the research gap in combining the attribution of human elements in virtual agents and social relationships among these agents to generate a realistic society. We merged ideas from various different approaches to AI and agent-based simulations along with borrowed concepts from psychology and sociology. The social sciences are vital to research in the field of believable societies. Since we are striving to generate a society that looks realistic on the macro-level (the population as a whole and its connectedness) as well as the micro-level (individual agents having human-like characteristics), it is necessary to use real-world research in the social sciences to model the artificial society.

We designed a detailed framework for generating such a society, and examined several models for the generation itself. A prototype system was also implemented from the blueprints, which shows potential as a society generator.

Although simplified, the system generates a society as is expected, according to the given configuration parameters. However, not all models within the system are successful. In particular, the Monadic model and Genesis model produced erratic or unexpected results.

The Monadic model does generate a society of single individuals as expected, but the agents are unable to find suitable mates and thus that the society is composed solely of the initial population, since families of children are never formed.

The Genesis model yields very inconsistent results as societies often go extinct before they can attain a stable population; and the times that the society *does* survive, the resulting population sizes vary greatly. These unpredictable results show that this model in its current form is not suitable for generating a society; however, a set of additional governing parameters and factors may help to stabilize this model too.

6.2 Contributions

This work was the first of its kind, to the author’s knowledge, and it laid a solid foundation for approaches to creating a networked population for video games. Our research offers multiple contributions to the field.

For one, we propose a model for giving virtual agents a social aspect, manifested in their ability to form friendships and romantic relationships. These are not just random connections either; they are formed on the basis of social circles, or groups, in which the agents are involved. The romantic relationships are even more personalized as the agents will look for partners with whom they are compatible.

We also contribute the series of models for generating societies. Although some of these are not completely novel concepts, they are incorporated into a new mechanism, and so used in a new way.

One particular contributing focus of this thesis is on creating the personified agents as well as their various relationships, without pre-scripting any of it.

The prototype simulation that was implemented as a tool for showcasing this thesis’ work is another large contribution to this field of study.

6.3 Discussion

Some of the computation time of large populations may be able to be alleviated through one or multiple techniques. Several methods have been proposed for dealing with computationally expensive believable AI in video games in [33].

Since the population is agent-based, several of its components would also qualify for parallelism using one or multiple multicore parallel hardware devices, most commonly on multicore CPUs (central processing units). This would allow several agents to be evaluated simultaneously, by dividing up the population amongst the available processors, and so ideally decreasing overhead time. The authors of [11] report three industry level video games that yielded decreases in overhead times on dual-core machines compared to single-core machines. Depending on game genre and graphical requirements, there may be an additionally opportunity to distribute a portion of the workload to GPUs (graphics processing units). Although GPUs have recently become streamlined to handle non-graphical data parallelism, their primary purpose is for graphics and thus should be used for that purpose first and foremost rather than be bombarded with AI computations. In most graphical video games the GPUs should work solely on those graphics computations, but in games that contain no or little visuals, the GPUs may be able to accommodate some of the AI workload. When dealing with parallelism on CPUs, GPUs, or any other kind of parallel hardware, it is important to note that there is **not** a guarantee of speedup. The overhead associated with the I/O transfer of data to and from the parallel hardware may outweigh any potential gain from the simultaneous executions, and thus each module in the simulation must be examined methodically for optimal efficiency. Furthermore, some modules may be better suited on one form of hardware (i.e. multicore CPUs) while other modules run more efficiently on a different form of device (i.e. GPUs), so the optimal efficiency may be a heterogeneous approach, using both kinds of parallel hardware [34].

6.4 Future Work

Our research has introduced a new approach for generating realistic societies for video games. Our work is a stepping stone that just scratched the surface of an untapped field of study. As researchers strive for perfect realism in AI, we recognize that there are numerous opportunities to improve that which is presented in a simplification in the current work. The presented framework provides an approach (or multiple approaches) to generating a society under a set of predetermined or parameterized specifications. A variety of society types can be created solely by adjusting the external parameters. For an even wider variety, the individual modules within the framework can be modified. Some future opportunities may even require a large-scale change in infrastructure.

Several ideas for future work will follow under the broader categories: Psychological; Relational; Environmental; A System of Societies; Incorporation the Society in Video Games; and Dynamic Parameters.

6.4.1 Psychological

Human behaviour is extremely complex and combines a multitude of factors from the person's past experiences and memories, to their emotions, traits, and attitude, as well as environmental and societal factors. The current research incorporates a limited number of personality traits and attributes, and selects them in a mostly independent fashion, whereas human traits are often interdependent. Humans are so complex that computer simulations may never be able to achieve a perfect system to model them, but there's certainly a lot that can be added to improve the current model. Several researchers have explored various other aspects to adding believability to agents through emotion, memory, or moods [4, 35]; none of which were implemented in this proposed simulation, but are elements that would help to further the realism in a virtual society.

Most of the traits and attributes used in this research were selected randomly according to a particular distribution or set of weighted options. Human's traits are often not only dependent on other traits, but also dynamic and will evolve over a person's life, and often affected by those around them. Studies are also finding that children may inherit certain personality traits from their parents genetically [36], and that their personality is also shaped by their upbringing and environment [36]. These personality dynamics are not examined in the current research, but are elements that can be included in future work.

6.4.2 Relational

We simplified the forms of romantic relationships and the processes to form them. The current simulation only allows heterosexual relationships, and these relationships are limited to courtships or marriage. Furthermore, we avoid many of the messy complications in relationships and families by removing the possibility of divorce from marriage and by only allowing married couples to have children. We recognize that these are large simplifications, especially in modern society where there are all kinds of open relationships, broken relationships, pre-marital child bearing, homosexual mating, and many other forms of relationships. To create a realistic society, the framework should be flexible to allow many of these non-traditional relationships, families, and lifestyles. In future work, these options should be addressed.

6.4.3 Environmental

An environment is one of the three key components of an artificial society [12], but is not incorporated into the current simulation. We did not feel a need to include one, since this research deals with the population and network generation rather than the gameplay. There are still opportunities for an environment that could be explored in future society generation frameworks. The society could contain a set of houses and apartments in which the agents could be assigned to live in. This feature would then lead to creating a new set of Groups for districts and smaller-scale neighbourhoods, and even individual buildings. Then agents will be candidates for friends with those who live nearby. There could also be seasonal or one-time environmental events or occurrences that impact the agents' lives to some degree. These could include severe weather phenomena (i.e. hurricane) that cause individuals to evacuate for a period; disease epidemics that cause mass deaths or illnesses; or rises in criminal activity requiring a greater number of police officers in the area. Many of these environmental factors would impact the numbers of career openings for specific jobs (i.e. police officers, doctors, paramedics, etc.), which thus will affect the agents' career path selections. The number of career openings is being used in the current simulation, as described in Section 3.1.3.5, but is only dynamic with respect to the population size. These environmental impacts on career requirements could be directly implemented by having them trigger a change in the table of career opening numbers.

6.4.4 A System of Societies

The presented framework is for generating an artificial society, but this also creates an opportunity for generating several societies from which the agents can travel or move back and forth, just as in real world cities. In the current framework, the agents can move into and out of the society, but only the local activity is recorded in detail and used. A game could incorporate several of these societies and, with some modifications to the framework, could connect them all to create a more realistic and flexible system.

6.4.5 Incorporating the Society in Video Games

The ultimate purpose of this research is to create a society of realistic agents to dwell in a video game world. A great next step would be putting the society created from this framework into a video game, even if just a simple one. This coalescence would require additional components to give the agents the ability to interact with one another, but once that is complete, the agents would already have their personality, interests, and life archives which would guide their conversations in style and content. For example, consider a person, Alice, who has lived in several different locations throughout their life, and another person in the society, Bob, who is looking to take a vacation with his family. Alice and Bob meet and converse together, and Alice suggests that Bob take his family to Vancouver – a location that Alice once lived in. Consider another scenario with a person Charles who is looking for someone to create a website for his business. Charles must think through his list of friends about if any one of them has a career as a web designer or computer programmer, and consults them about the request.

Recent research has begun to examine player-NPC dialogue and the effects of personality and moods on the style and content of the conversations [37]. An engine like this could be combined with the agents in this society to allow such conversations and interactions to take place amongst the agents.

There has also been research into agents' personalities and emotions affecting the goals they create for themselves and the approaches to achieve such goals [20, 38]. This would also be a valuable component in a video game that incorporates the aforementioned AI, so that the agents are actively pursuing goals and filling their lives with meaningful activities, according to their personality and cultural identity.

6.4.6 Dynamic Parameters

Many of the simulation's governing rules and parameters are static and generally independent from one another. In a real society, there are several evolutionary trends that occur; some as reactive and adaptive changes that are invoked by specific conditions while others evolve from random fluctuations or unexplainable causes. In future work, it would be ideal to create a more flexible system that can accommodate dynamic parameters. For example, the Genesis model requires a greater child distribution than do the other models, however the increased distribution is only required for the first several generations so the population can expand drastically quickly. An ideal system would allow these distributions to be used for a certain period, but then a more relaxed distribution to then take effect replacing the now excessive distribution, since the population does not continue to need growth once it has reached a sufficient level.

7 Bibliography

[1] R. W. Picard, Affective Computing, M.I.T. Media Laboratory Perceptual Computing Section Technical Report No. 321, 1995

[2] K. Isbister, Better Game Characters By Design: A Psychological Approach, Elsevier Inc., 2006

[3] C. Bailey, J. You, G. Acton, A. Rankin, M. Katchabaw, Believability Through Psychosocial Behaviour: Creating Bots That Are More Engaging And Entertaining. Appeared in Believable Bots: Can Computers Play Like People? Edited by Philip Hingston, Spring, 2012

[4] C. Rosenthal, C. B. Congdon, Personality Profiles for Generating Believable Bot Behaviors, 2012 IEEE Conference on Computational Intelligence and Games, 2012

[5] A. Iglesias, F. Luengo, Intelligent Agents in Virtual Worlds, Proceedings of the 2004 International Conference on Cyberworlds, 2004

[6] I. Umarov, M. Mozgovoy, P. C. Rogers, Believable and Effective AI Agents in Virtual Worlds: Current State and Future Perspectives, International Journal of Gaming and Computer-Mediated Simulations, vol. 4. Iss. 2, 2012

[7] G. N. Yannakakis, Game AI Revisited, Proceedings of the 9th conference on Computing Frontiers, 2012

[8] J. Schaeffer, V. Bulitko, M. Buro, Bots Get Smart, IEEE Spectrum, 2008

[9] Canadian Oxford Dictionary, Second Edition, Oxford University Press Canada, 2004

[10] C. M. Macal, M. J. North, Tutorial on Agent-Based Modelling and Simulation, Journal of Simulation (2010) 4, 2010

[11] J. Tulip, J. Bekkema, K. Nesbitt, Multi-threading Game Engine Design, Proceedings of the 3rd Australasian conference on Interactive entertainment, 9-14, 2006

[12] J. M. Epstein, R. Axtell, Growing Artificial Societies, 1996

[13] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast Unfolding of Communities in Large Networks, 2008

[14] A. Buscarino, M. Frasca, L. Fortuna, A. S. Fiore, A New Model for Growing Social Networks, 2012

[15] B. Lacroix, P. Mathieu, Automated Generation of Various and Consistent Populations in Multi-Agent Simulations, 2012

[16] D. Ballas, et al., SimBritan: A Spatial Microsimulation Approach to Population Dynamics, 2005

[17] Ge et al., A Framework of Multilayer Social Networks for Communication Behavior with Agent-Based Modeling, 2013

[18] K. Harland, A. Heppenstall, D. Smith, M. Birkin, Creating Realistic Synthetic Populations at Varying Spatial Scales: A Comparative Critique of Population Synthesis Techniques, Journal of Artificial Societies and Social Simulation **15** (1), 2012

[19] A. B. Loyall, Believable Agents: Building Interactive Personalities, 1997

[20] F. Bouchet, J.-P. Sansonnet, Influence of Personality Traits on the Rational Process of Cognitive Agents, 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, 2011

[21] M. R. Rodrigues, A. C. da Rocha Costa, R. H. Bordini, A System of Exchange Values to Support Social Interactions in Artificial Societies, AAMAS ’03

Proceedings of the second international joint conference on Autonomous agents and multiagent systems, 2003

[22] S. Sardiña, S. Shapiro, Rational Action in Agent Programs with Prioritized Goals, AAMAS ’03 Proceedings of the second international joint conference on Autonomous agents and multiagent systems, 2003

[23] M. Ponsen, H. Muñoz-Avila, P. Spronck, D. W. Aha, Automatically Generating Game Tactics through Evolutionary Learning, AI Magazine, Vol. 27, No. 3, 2006

[24] Q. Gemine, F. Safadi, R. Fonteneau, D. Ernst, Imitative Learning for Real-Time Strategy Games, 2012 IEEE Conference on Computational Intelligence and Games, 2012

[25] E. Haasdijk, P. Vogt, A. E. Eiben, Social Learning in Population-based Adaptive Systems, IEEE Congress on Evolutionary Computation, 2008

[26] C. Sun, S. Wang, Modeling Adaptive Behaviors on Growing Social Networks, 2008

[27] C. Kadushin, Understanding Social Networks, Oxford University Press, 2012

[28] T. B. Heaton, Factors Contributing to Increasing Marital Stability, Journal of Family Issues, Vol. 23, No. 3, 2002

[29] J. Greenwood, N. Guner, G. Kocharkov, C. Santos, Marry Your Like: Assortative Mating and Income Equality, National Bureau of Economic Research, 2014

[30] C.-Y. Huang, Y.-S.Tsai, T.-H. Wen, Simulations for Epidemiology and Public Health Education, Journal of Simulation (2010) **4**, 2010

[31] M. Selfhout, W. Burk, S. Branje, J. Denissen, M. v. Aken, W. Meeus, Emerging Late Adolescent Friendship Networks and Big Five Personality Traits: A Social Network Approach, Journal of Personality, 78, 2, 2010

[32] X. Xiang, R. Kennedy, G. Madey, S. Cabaniss, Verification and Validation of Agent-based Scientific Simulation Models, 2005 Agent-Directed Simulation Symposium, 2005

[33] A. Rankin, G. Acton, M. Katchabaw, A Scalable Approach To Believable Non Player Characters In Modern Video Games, Proceedings of GameOn 2010, 2010

[34] M. P. Wachowiak, B. B. Sarlo, A. E. Lambe Foster, High-Dimensional Adaptive Particle Swarm Optimization on Heterogeneous Systems, Journal of Physics: Conference Series, **540** 012007 (9pp), 2014

[35] A. Heuvelink, M. C. A. Klein, J. Treur, An Agent Memory Model Enabling Rational and Biased Reasoning, 2008 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, 2008

[36] D. P. Schultz, S. E. Schultz, Theories of Personality, Tenth Edition, Cengage Learning, 2013

[37] C. M. Rose, Realistic Dialogue Engine for Video Games, Master’s Thesis, 2014

[38] A. Camurri, G. Volpe, A Goal-directed Rational Component for Emotional Agents, 1999 IEEE International Conference on Systems, Man, and Cybernetics, Vol. 4, 1999

Curriculum Vitae

Name:	Bryan Sarlo
Post-secondary Education and Degrees:	Nipissing University North Bay, Ontario, Canada 2009-2013 B.Sc.
	The University of Western Ontario London, Ontario, Canada 2013-2015 M.Sc. Candidate Expected graduation: June 2015 Supervisor: Michael J. Katchabaw
Honours and Awards:	Ontario Scholar 2008-2009
	Graduate Student Teaching Award 2013
Related Work Experience	Research Assistant Nipissing University 2011-2013
	Teaching Assistant The University of Western Ontario 2013-2015

Publications:

Sarlo, B. B., Foster, A. E. L., Wachowiak, M. P. (2012). A Visualization Framework for Simulation Fuel Consumption Through Serious Games. 17th International Conference on Computer Games (CGAMES), 103-107.

Wachowiak, M. P., Sarlo, B. B. (2013). Interactive Visualization of Dynamic and High-Dimensional Particle Swarm Behavior. IEEE International Conference on Systems, Man, and Cybernetics (SMC), 770-775

Wachowiak, M. P., Sarlo, B. B., Lambe Foster, A. E. (2014). High-Dimensional Adaptive Particle Swarm Optimization on Heterogeneous Systems. Journal of Physics: Conference Series, **540** 012007 (9pp).