Agent-Based Modeling of Urban Society and Interactions
TJHSST Senior Research Project Proposal
Computer Systems Lab 2009-2010

Andrew Imm

October 26, 2009

Abstract

This project uses an agent-based modeling approach to simulate human society in an urban environment. After constructing a two-dimensional representation of a city, the simulation is populated with goal-oriented agents with unique personalities that guide their decisions. These agents are assigned daily schedules to follow, and they are forced to interact with other agents in order to complete their goals. This interaction allows the agents to build a social network of friends and colleagues, and it also allows the spread of a virtual "virus" — a final that allows this simulation to be used to accurately model the transmission of an infection through an urban environment.

1 Purpose and Scope

The ultimate goal of this project is to create a simulation of urban society that is realistic enough to be used in the analysis of infections disease spread. In its most basic form, this program will model the day-to-day movements of people within an urban environment. The simulated citizens, or agents, will follow routines, travel around the city, and interact with each other through "conversation" and the sharing of knowledge. Their actions will partially be guided by their preferences; each agent will have numeric values that somewhat control how likely they are to work, sleep, or interact with others. As the program ages, the agents will develop social networks — groups of people whom they prefer to interact with. These networks, by necessity, will be constructed of co-workers, neighbors, and other people who are often encountered throughout the day. Finally, when the agents have been established in their routines and travel around the city in a realistic manner, I will introduce the ability to spread a virus through agent interaction. Because the city will — at that point — resemble a real-life urban environment, the results following the introduction of the infection can be extrapolated to a real urban center. Due to the extensible nature of the program, I will also be able to implement various quarantine methods to produce preliminary results on how best to combat an epidemic in a bustling metropolis. Currently, such detail-oriented agent-based models are not used to model infection. The CDC and WHO use mathematical models that treat the entire population as one enormous individual. This modeling technique lacks insight into the decisions and behaviors of individuals, though; human beings cannot be expected to act the same, nor can they be grouped into discrete categories for the sake of modeling. By taking into account the movements and actions of individuals, this model should be able to produce more accurate results than those seen in the non-agent-based models.

2 Review of Current Literature

I have been reading several articles concerning agent-based modeling of an urban environment. However, those articles are chiefly concerned with path-finding, and do not go anywhere near the depth I hope to implement. I have also been reading papers by Dr. Stephen Eubank, a professor at Virginia Tech. He has researched the spread of epidemics by modeling them with agent-based systems, and is part of a CDC group that is working on accurately modeling the spread of disease through the nation. His research has mostly been with models that represented entire nations, though, so his scale is far larger than that of my highly-detailed model.

3 Project Structure

This project uses the Python programming language to handle the backend processing of the model and — optionally — the Tkinter library to handle the running of a graphic display. Additionally, all tools used to expedite testing and development (such as the map builder) are written in Python.

The project is divided into three separate areas of code: the first includes all code that is central to the actual simulation, the second includes programs such as the map builder which are key to the progress of the project, and the third includes all tests used to measure the speed, robustness, and practical limitations of the program. The majority of the code falls into the first category. All code used at simulation runtime, including code used to draw the graphic simulation display, is included in this piece of the project. As the project progresses, the need for certain tools arises. For instance, in order to easily manage the large, complex map files that the simulation uses, a map editor is necessary. It provides the ability to create map files with the aid of a user-friendly graphic interface that displays the map as it will render in the simulation. These tools are large pieces of code that are not part of the actual simulation at run-time, but they help to speed up the process of developing the entire project. As such, they still require a large amount of development time, but they are not usually seen when the simulation is finalized.

4 Testing and Analysis

Because of the complexity and size of the project, each piece will require separate testing. The primary round of tests will measure the program's limitations concerning the size of the model. These tests will gauge the efficiency of algorithms such as the path-finding method used by agents to navigate the city, and will determine the maximum number of agents that can be handled by the simulation. Results from these tests will be used to design the later experiments, and may be used to determine necessary reduction of processing — for instance, queueing navigation requests so that only a certain number of path-finding algorithms are run during each program cycle. The secondary round of tests will determine how well agents are able to follow their schedules by comparing their pre-written schedule with the times at which they actually arrive at their destinations. These tests will be used to fine-tune the creation of schedules for agents, and will determine how much time is necessary to traverse the map. Finally, when social networking has been implemented, tests of these networks can be run. One of these tests will create a visual map of the social network to determine how agents are connected by their relationships. Another test of the social-networking system will compare the time spent socializing with "friends" to the time spent interacting with "strangers." All three stages of testing will ensure that the final program is efficient, stable, and accurate.

5 Algorithms and Methods

In order to properly implement the urban society, a number of algorithms need to be implemented. The most important is the agents' path-finding algorithm, which will be implemented as a grid-based A* search. Branching outwards from their current position, agents will use a manhattan-distance heuristic to locate the shortest path to their destination. This navigation algorithm will also be tailored to work around moving obstacles — other agents who temporarily block the path until they move out of the way. Other algorithms used in the project will include the decision-making algorithm. This method will take into account an agent's schedule, current location, proximity to other agents, and personality factors in order to determine the agent's actions. Ideally, the personality factors will tailor the agent's response to the other three variables. These algorithms, along with others, will create the sense of realism of the simulation.

6 Expected Results and Value to Others

Due to the realistic nature of the final model, the results after the virus has been introduced into the population should be fairly accurate in determining how an epidemic spreads through an urban network. Hopefully, this program will be an excellent starting place for further research into accurate modeling of viral spread — such information would be helpful to the CDC, the WHO, and other health organizations whose current simulations fail to take into account the individuals that make up a population.