Using NetLogo to Simulate Building Occupancy of a University Building Environment

Dr. Sanish Rai, West Virginia University Institute of Technology

SANISH RAI is an Assistant Professor in the Department of Computer Science and Information Systems at West Virginia University Institute of Technology, Beckley, WV. He received his Ph.D. degree from Georgia State University in 2017. His research interests include simulation and modeling, agent and graph based systems, artificial intelligence, data assimilation and machine learning. His email address is sanish.rai@mail.wvu.edu

Mr. Thomas Keith Carter
Mr. Bimarsh Sharma

Using NetLogo to simulate building occupancy of a university building environment

Sanish Rai, Thomas Carter and Bimarsh Sharma
Department of Computer Science and Information Systems
West Virginia University Institute of Technology
410 Neville St., Beckley, WV, USA
sanish.rai@mail.wvu.edu

ABSTRACT

Building occupancy simulation is an interesting area of research for not only in the construction planning but for other applications such as security monitoring, crowd management, and occupant behavior analysis. Simulating an environment such as a university department allows a simulation model that will help in the study of scenarios that will help in evacuation during fire, earthquake or threat management. We can use the simulation system to study the behavior of students, teachers and analyze how an effective evacuation can be done in the current environment in real time. Agent-based modeling is the most popular method for developing models where there is an active interaction with the components of the model. Considering the need for agent-based models, we used NetLogo, an open source agent-based simulation software to build a simulation model for a building environment with occupants in it. The open source nature of the software and convenience in building Graphical User Interface (GUI) applications makes NetLogo a popular use to develop models for agent-based system requiring an interactive interface.

INTRODUCTION

Computer simulation in general means building a computer model of a process or a system. It allows us to use computer resources to simulate various activities that go into a system and get the results or observations that would occur in the real system. There are multiple advantages to computer simulation. It allows building models which otherwise would be costly to build in the real world. It allows to visualize a concept of a process without building it physically and check if it is correct, viable or needs modification. It allows to model the real-world systems such as city planning, traffic management, fire control and to implement new changes or add new applications to those systems. After the successful study and analysis in simulation, they could be implemented in the actual real system, thus saving a lot of cost and resources. It also allows simulation of various medical tests on live organisms, which is not possible in real world. In this work, we focus on a computer simulation model for a normal building.

Building simulation means a computer model of a real building structure along with the occupants who interact in the building. It consists of building walls, entrances, rooms, stairs and other physical structures as its environment. The occupants and their activities depend on the type of building being simulated. For example, in a house, there will be a family, and they will do basic activities such as sleeping, eating, etc. In a normal office environment, the occupants will normally come to the office, attend meetings, go to the cafeteria, and leave office, whereas, in a school, the students will go to their classes, have breaks, and go home. In a hospital, there will be people 24 hours, but the main activities will be during the day. In a train station, there will be a lot of people coming and leaving the station during the rush hour. In an airport terminal, there will be thousands of people moving around waiting for their flights, staffs helping clients and similar other activities.

Building occupancy simulation finds its application in a variety of fields leading to increasing emphasis on the development and study to build occupancy dynamics model. A model of occupancy dynamics in a building needs to correctly model the structure of the building, occupants and their activities in it. The model will allow us to analyze the behavior of occupants in various applications such as occupant behavior and movement simulation, evacuation simulation, threat analysis or stadium evacuation simulation. Modeling these kinds of behaviors helps to understand their behavior and is useful for conserving energy or designing smart buildings. In other application, evacuation simulation can be used to model the occupant's behavior in case of emergencies (fire, explosion, toxic gas threat). The evacuation process is dependent not only on the building structure (passage width, exit placements, obstacles) but also on the crowd's behavior (speed, exit information, time to react). As such, it becomes important to have a model, which can simulate the process correctly and analyze the performance of evacuations in a specific building for various types of occupants.

Many methods are used for building occupancy simulation, out of which agent-based is popular for the system in which there is high interaction in its components, and it is necessary to model it. There is an agent who represents an occupant, and the dynamic process of occupants is simulated repeatedly over time to generate the complex and intriguing emergent behavior [1]. The agent-based model works at a detailed level with a focus on agent's properties and interaction and has the advantage of being able to represent each occupant's behavior and decision making in detail at the cost of computational resource consumption increase with the increase in the number of agents. In this work, we try to simulate a building simulation similar to a university department. As such agent-based models satisfy our requirement as there are not many occupants but it allows us to observe the interaction between the occupants.

For our work, we used Netlogo [2] which is an open-source agent-based programming language. It has been used by researchers to build agent-based models for various applications conveniently. Two undergraduate sophomore students self-learned the programming language to understand the concepts of simulation, agent-based programming and built a preliminary building model to simulate interactions between two different groups of people. Here, we present the model, the simulation achieved and also briefly discuss the development process.

RELATED WORKS

Agent-based models are well suited for smart environment simulations as agents can model the human and their characteristics naturally. Agents have been used in [3] to simulate user behavior in domestic settings dynamically and to identify the context, beliefs, and facts impacting energy-related behavior. The agent-based framework is used to develop energy efficient strategies implemented through social campaigns, ubiquitous computing or centralized and distributed approaches. In [4] a multi-agent simulation framework has been presented for the study of human behavior during building emergency evacuations. The framework includes a visual sensor which analyzes the environment and helps agents to make decisions. The system models the emergent human behavior by simulating the behavior of human agents at the microscopic level. In [5], the authors used agent-based framework using Java to build a building simulation model for occupants in a university department. Sensors were also modeled to extract data and use in occupancy estimation.

NetLogo is popular in simulation research for being convenient to build agent-based models quickly. In [6], the authors used NetLogo to build an evacuation model and observed behavior patterns during the evacuation, mainly arching, clogging and other herding phenomena. In [7], the authors build an integrated modeling framework for measuring the usability of designs before the buildings are constructed. In the model, the human factors were modeled using NetLogo. In another work, the authors in [8] used NetLogo to build a model to study interactions of human in an environment during and immediately following a natural disaster. We found NetLogo widely used for building agent-based simulation and modeling in academia and research area. Also, the open-source nature of the software made it attractive to adopt in our work compared to other commercial agent-based modeling languages.

AGENT-BASED MODELING

An agent-based model (ABM) is a model for simulating the actions and interactions of single or multiple agents to analyze their emergent and individual behavior in the overall system. A single agent is defined as a discrete entity with its goals and behaviors with a capability to adapt and modify its actions depending on the environment. Agent behaviors are defined as a simple set of rules based on which agents perform their action, interact with other agents and the environment. These interactions over the course of time and space give rise to various system behaviors, patterns, and structures which give a better understanding of a rather complex system. They are widely used for modeling occupants and their behavior in a building structure. An agent in the model can represent a real-world occupant entity with appropriate property and behavior which makes the model reasonably accurate to represent occupant movement dynamics in a given environment. An agent can be modeled to behave like a person by assigning certain speed, size, goal, and decision-making property. Heterogeneous agents can be created as different agents with different properties representing a mixed population of various gender, size, age, speed, motive, interactions, and other properties.

ABMs are suitable to model occupancy dynamics since we can represent occupants easily model agents with human-like behaviors and decision-making capabilities. The building where the occupants exist is set as the environment, and the interaction between agents and environment can be defined as a set of rules. The agents perform the behavior of the real occupants under situations that the system is meant to execute. For example, simulation can be performed where occupants are given destination as goals. Now the agents will move towards their destination following some rules which govern their direction and movement. They might interact with each other and exert behaviors like cohesion or avoidance. The environment might be a building structure with various rooms as their locations and destinations.

NETLOGO

NetLogo is a suitable programming language for developing agent-based models as it allows to model complex interactions between the agents in real time and to observe the emergent behaviors. It is convenient to simulate hundred to thousands of agents and give instructions externally when they are simulating. It allows users to observe simulations and interact with them to explore the model's behavior in various scenarios. This is an important feature as it allows you to introduce new agents and even properties in an ongoing simulation to create new conditions. It is easy to install and comes with preinstalled models which are the sample models that could be used for learning and reference for new models. The main feature of NetLogo is its GUI which is important in simulation models for building occupancy as we need to observe the occupancy behaviors and not only compute it. Behaviors in buildings such as follow, avoid, reach, walk need to be observed to understand and perform analysis easily.

As seen in Figure 1, NetLogo provides an easy interface to add various tools to the GUI to interact and analyze the system. For example, we can add buttons to call procedures from the code, slider to enter variable numbers to system, monitors, and plot to analyze the system parameters in real time. We can also increase or decrease the system run time speed by modifying it from GUI. We can type the code in Code tab, and switch to GUI tab to run the system and check the changes conveniently.

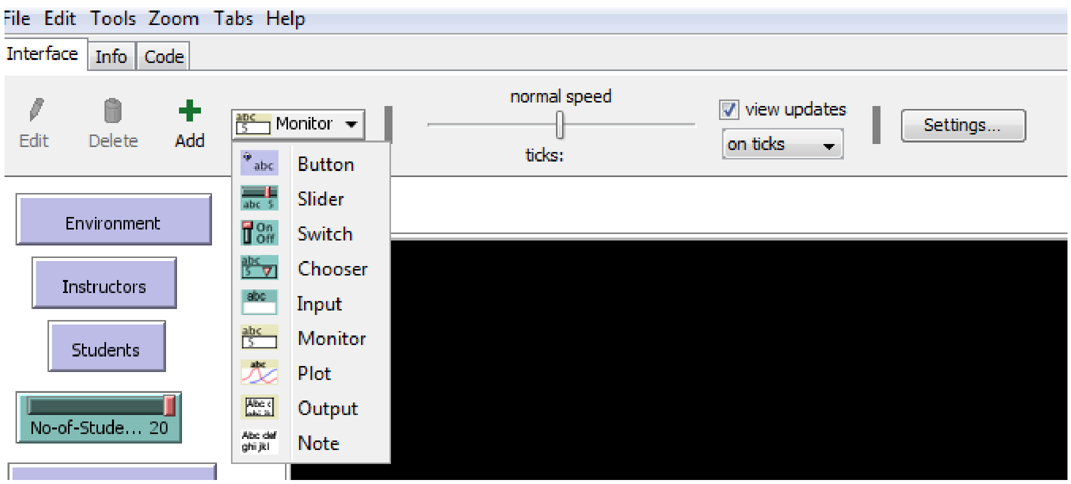

Figure 1. A NetLogo GUI

BUILDING OCCUPANCY SIMULATION MODEL USING NETLOGO

In this work, two sophomore students built a basic simulation model for a university style building. One of the student was supported by XSEDE Empower program [9] and another student was supported by NASA West Virginia Space Grant Consortium [10]. The building structure represents a layout similar to a general department or a community space. The model built is a general model, but we use the application for a university environment as it was our motive for this application. However, the model can be best described as an application to simulate interactions with two groups of people, where one group is seeking for people in another group.

For example, in a shopping market where people seek cashier to check out, or in a student career fair where students seek to communicate with prospective employers. For our convenience, we divide them into two groups: students and instructors.

Figure 2 (a) represents the building environment, where the black area represents the area where people can walk, and the green area represents the walls or areas where people cannot walk. The layout mimics our Computer Science department where green areas are the closed rooms where people do not go frequently and do not need to be simulated. In NetLogo, *turtle* are the agents that can move around so we will use it to create our occupants: instructors and student agents. Instructors will be positioned in a particular location (their office or a common area) and students will enter from an entrance seeking for an instructor to meet. After meeting, instructor students will leave the area. Instructors are represented in blue color with the human figure and students in red color. Figure 2 (b) shows a snapshot of the simulation where the instructors are in the office spaces and corridor while students are just entering the area from the entrance.

The state space (the floor surface) in the NetLogo is managed by the agent *patch* where the minimum area is 1 by 1 pixel. As such the agents will walk 1 pixel per time step (simulation time). To manage the basic properties of occupants in the model, we implemented three behaviors for the agents. The first one is to avoid an obstacle, where the occupants will avoid the green obstacle. In their path, if they find an obstacle in front of them, they will randomly move to a side and continue their path until there is no obstacle in their path. Second, they will avoid each other. If two occupants tend to move into the same *patch*, they will randomly move to a new direction to avoid each other. Third, each student will seek an instructor. It is maintained by creating a link for each student with a random instructor. Figure 3 (a) shows the lines connecting each student with an instructor. This relation is created using the agent *link* from NetLogo which allows two agents to be linked and interact. In our model, we will make the students to seek an instructor, move towards them, meet them and exit the area after some time.

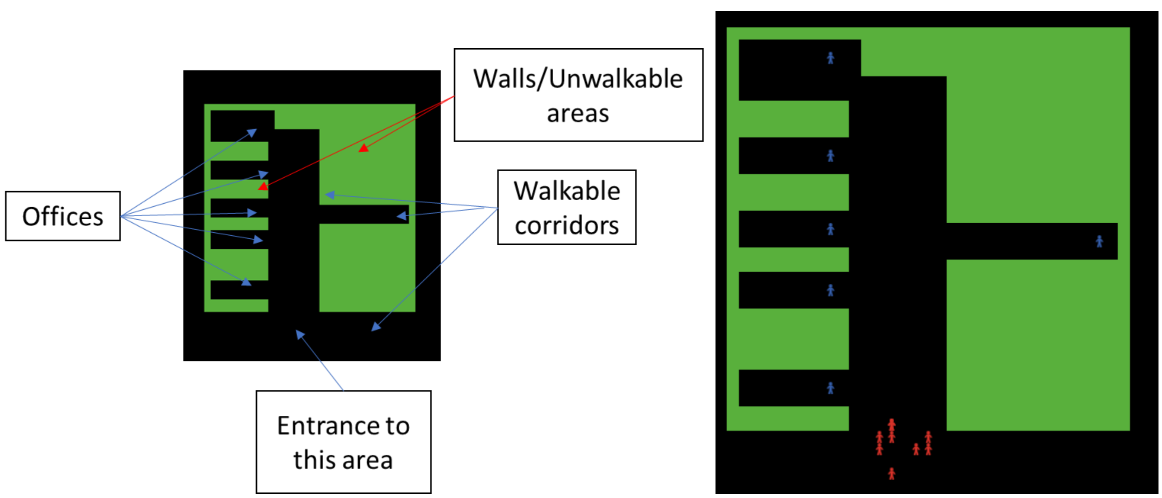

Figure 2. (a)Environment of the building in NetLogo (b) with instructors and students

To make the simulation interactive, there are seven main buttons that allows to interact with the simulation model. The buttons are shown in Figure 3 (b). The button Clear, clears and resets the GUI for restarting the simulation. The Environment button allows creating the building obstacles.

If we do not click it, we can run the simulation in a space without any obstacles. The button Instructors allows to create six instructors in the environment. In the beginning, they are created in their offices as shown in figure 2 (a), if we create more, they are created in random locations. The button Students allows to create the number of students selected by the sliding option No-of-Students. The minimum number is 1 and the maximum is 20 and they are all created near the entrance in random locations. The button Remove Instructor and Remove Students allows to remove a random instructor and student from the model. This will allow to simulate different scenarios and observe the behaviors that occurs in the system. The button Go is to run the simulation, which makes the students seek their instructor, meet them and leave the area.

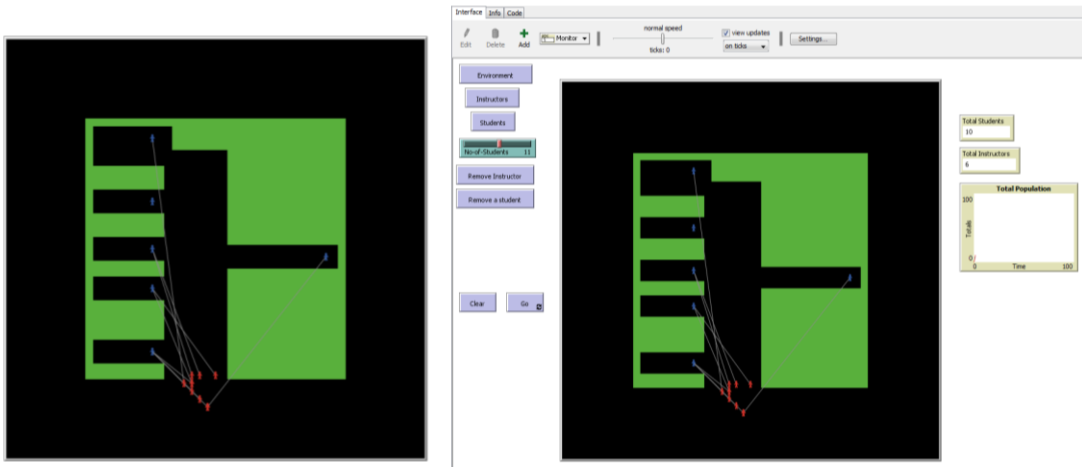

Figure 3 (a) Links between students and instructor (b) Simulation model GUI interface

Since the simulation is randomized every time, during each simulation there will be different behaviors observed. To analyze the output of the simulation, we used Monitor and Plot tools easily available from GUI. We counted the total number of students and total instructors in a given time using the monitors. We used the plot to compare the number of occupants in each time step. Figure 4 (a) shows a snapshot where there are 9 students and 6 instructors and the simulation was started with 11 students. We can observe the information in the monitors and plots on the right-hand side of the GUI. Figure 4(b) shows a plot for a total of 251 simulation steps which starts with 24 instructors and 100 students and the number of students gradually drops, while some of the instructors are removed manually.

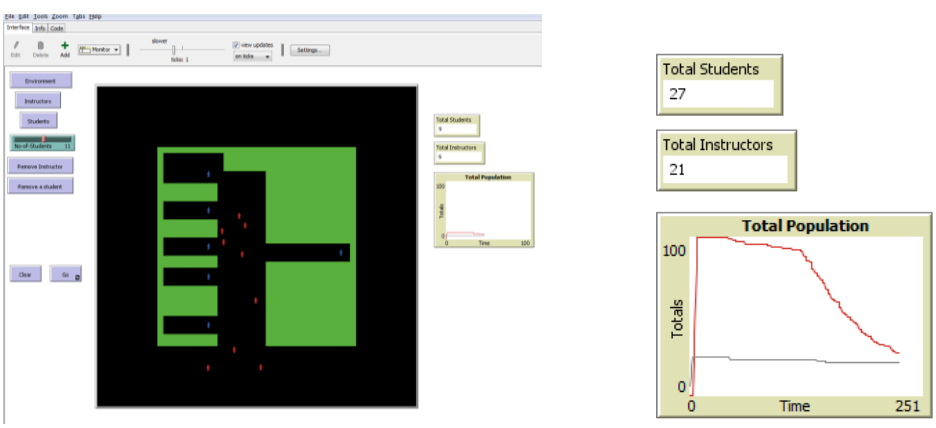

Figure 4(a) Snapshot of a simulation step (b) Plots for a certain duration of the simulation

DISCUSSION

The model is in the preliminary phase due to limited time but can be easily expanded with more features. Properties of the agents can be added as rules and depending on the required analysis more plots, or data reporting can be done. We found some bugs in the program as well. For example, some agent when facing an obstacle in large size, when tries to turn to a non-obstacle region would still be inside the obstacle. And as the agent is always moving it will always try to find a non-obstacle path and instead get inside the big green patch and keep rotating in a random direction. Better rules can be written in the code to avoid those scenarios.

The students had learned only java programming language, and it took a lot of time to understand the new language syntax of NetLogo. Students had to invest a lot of time in understanding the basic concepts of agent-based and simulation before starting the coding process. Each student would meet weekly for one hour with the research advisor to discuss their progress and get help with concepts and programming. We found that documentation of NetLogo to be helpful but not extensive. The knowledge base for NetLogo was limited, and it took a lot of effort to get a new concept working with the code.

Students are quite motivated to be able to develop the current simulation model and are interested in keeping working on it. Future work would be to fix the bugs and write a better algorithm for agent collisions. The actual dimensions of the department will be used to build the environment. Instructors will be modeled based on their office hours, and student interaction will be simulated. The application will be expanded to include other building simulation such as interaction in shopping malls and multi interaction areas like career fairs or a convention.

CONCLUSION

In this work, NetLogo, a multi-agent based programming language was used to build a building environment and simulate occupants. A basic spatial-temporal model has been developed where instructors and students can interact with each other by moving around the environment space. The instructors and students can be added and remove from the system, and the number of occupants can be monitored using plots in real time. In the future, we plan to expand the model in various other applications where there is continuous agent interaction.

Acknowledgement

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562. We would like to thank XSEDE Empower program for providing scholarship for student to work on the research.

We would also like to thank NASA West Virginia Space Grant Consortium for providing undergraduate research fellowship to student to work on the research.

REFERENCES

1. Macal, C. M., and North, M. J. Agent-based modeling and simulation. In Winter Simulation Conference, Winter Simulation Conference (2009), 86-98.

2. Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
3. A. Kashif, X. H. B. Le, J. Dugdale, and S. Ploix, “Agent based Framework to Simulate Inhabitants' Behaviour in Domestic Settings for Energy Management” in ICAART (2), pp. 190-199, 2011.
4. X. Pan, C. S. Han, K. Dauber, and K. H. Law, “A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations” in Ai & Society, 22(2), pp. 113-132, 2005.
5. Rai, Sanish, and Xiaolin Hu. "Behavior pattern detection for data assimilation in agent-based simulation of smart environments." Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on. Vol. 2. IEEE, 2013.
6. Almeida, J. E., Kokkinogenis, Z., & Rossetti, R. J. (2012, June). NetLogo implementation of an evacuation scenario. In Information Systems and Technologies (CISTI), 2012 7th Iberian Conference on (pp. 1-4). IEEE.
7. Andrews, C. J., Yi, D., Krogmann, U., Senick, J. A., & Wener, R. E. (2011). Designing buildings for real occupants: An agent-based approach. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 41(6), 1077-1091.
8. Liu, Z., Mehdi Jalalpour, C. Jacques, S. Szyniszewski, J. Mitrani-Reiser, James K. Guest, T. Igusa, and B. W. Schafer. "Interfacing building response with human behavior under seismic events." (2012).
9. John Towns, Timothy Cockerill, Maytal Dahan, Ian Foster, Kelly Gaither, Andrew Grimshaw, Victor Hazlewood, Scott Lathrop, Dave Lifka, Gregory D. Peterson, Ralph Roskies, J. Ray Scott, Nancy Wilkins-Diehr, "XSEDE: Accelerating Scientific Discovery", Computing in Science & Engineering, vol.16, no. 5, pp. 62-74, Sept.-Oct. 2014, doi:10.1109/MCSE.2014.80
10. “NASA WVSGC Undergraduate Research Fellowship Program.” NASA West Virginia Space Grant Consortium, www.wvspacegrant.org/programs/students/undergraduate-research-fellowship/.