Traffic Dynamics in Scholastic Environments

Alex Katkova
TJHSST
Computer Systems Lab

June 12, 2007

Abstract

A great quantity of simulations have been conducted since the beginning of computer time, studying everything from global population patterns to local climate. This simulation attempts to undertake the relatively small area of traffic simulation, more specifically within a school environment. The purpose of this project is to create a simulation of students and teachers at Jefferson moving around the building. The program will be coded in Java, using MASON, and a realistic and accurate simulation is expected at the end.

Keywords: Multiagent, dynamic simulation, probability, group mentality

1 Introduction

As far as it is known, no such simulation yet exists. A realistic representation needs to be constructed to reflect the need for a new school building and to allow users to determine paths of travel during their day at school. A graphic output shell exists through MASON, and the necessary classes will be written to detail the adventures of the students throughout their day at school. The students follow real schedules of current students at TJ, and the floorplan of the school closely resembles that of TJ.

The research being addressed is the study of how human beings move about buildings and to a certain extent how they interact with each other and with their environment. This study will take the approach of using real, current data, to model the movement and then compare it with what is actually observed. This research will bear the most importance to those at TJ and almost none to the outside world. The scope of this project is to model observed phenomena and allow some user control of environment variables.

2 Background

This project belongs to the family of agent-based modeling projects. Agent-based simulations rely on the characteristics of the objects involved in the simulation to produce results. The objects control their own actions based on a predetermined set of rules and are not subject to the direct control of the overall governing program. The most relevant area currently being studied is traffic modeling. Such simulations endeavour to find patterns of traffic to promote better development of intersections and traffic light control. These projects provide a basis for my project because I've traded the cars for students and defined specific paths for the students to follow.

The central algorithm that governs the movement of the students is the breadth-first search. In each timestep, the student knows his current location, the location of his destination, and the distance to the destination. He surveys select points around himself, each equal distances away from him, and determines which would be the most desirable in getting him nearer to his goal. He only looks one time-step ahead, and takes the path that would get him closer to his goal in the next time step. For such small dimensions as those of a school building, this is sufficient and is often quite accurate.

3 Development Sections

3.1 Requirements

This project was deemed succesful by the accuracy of the portrayed motion of the students. If the students went to the correct class when expected, and exhibited normal student behavior during breaks, talking to others, getting food, using the bathroom, running around, then the simulation was succesful. Upon the initation of a fire drill, the students should also proceed to their proper locations. The accuracy was judged by the difference between the simulation and real-life observations. The project should also allow user manipulation of variables, allowing him to set the class and break lengths to experiment.

3.2 Overview

The project started out as a tutorial in MASON. I used the graphical output capabilities of that tutorial and created my own classes to replace the ones in that tutorial. I coded the student objects, which move around the school, and acquired the necessary data from the school servers for creating the floorplans and schedules.

3.3 Planning

Informal tactics of assessing the components including simple observation of the studens to see if they behaved as they were programmed to behave. Formal tactics have not been employed yet. The lifecycle model used was Evolutionary Prototyping. The goal for first quarter was simply to get the simulation running, with all of the floor inputs written and students moving randomly around the school. During second quarter, I acquired the actual schedules and programmed the students to follow them. The endeavor for third quarter was to create invisible rooms so that students would not be able to move through courtyards or other classrooms as they travelled. For fourth quarter, I finished programming the fire drill and added in the appropriate student behavior for lunch. I also added in the option to run a red, blue or anchor day.

3.4 Testing and Analysis

For the most part, the type of testing I use now is informal testing by sight. I watch the simulation to see if the students are moving where they are supposed to be moving. The students have to start in the required places, and move through the doors, not the walls, to get into the building. They then have to start heading out in different directions, some toward their classrooms, and some toward the stairs to get to the second floor. The students may not enter any courtyards or walk through any classrooms unless they're going into the their next scheduled class. When the fire drill button is pressed, the students must head out of class and toward their designated locations.

In order to check whether students were indeed making it to their correct locations by means of a formal test, I graphed the distances of the students to class with respect to time. Every time step that the students move in, the simulation loops through all the students and finds their average distance to their destination. This test has confirmed what I already suspected based on the informal tests, that as time goes by, more and more students have made it to their proper locations. I also graphed the number of students that were in class over time, and that graph ended up looking much like the distance graph. I also want to test the step size by seeing how it changes with respect to the student density around the person. If the density is high, the step size should decrease, because clearly a student can't move quickly if there are many people around him.

3.5 Visuals

This is what the first floor looks like during a run. The various rooms can be seen and it should be noted that students are only located in areas in which they are allowed to be.

3.6 Developmental Procedures

Using the GUI provided by MASON, I created a SimState that contained several SparseGrids as private variables. The SimState read in data files that included the locations of the rooms, the initial locations of the students, and the schedules. It then entered all the data into the SparseGird and created all the Student, Room, and Wall objects. It then started the simulation and called the step methods of each of the Students. The Students determine their new locations based on a defined set of criteria and move there. See the Visual section for a diagram of relationships.

A sample run of the program would go as follows. The GUI class is run, and the class that it displays reads in the student locations, schedules, and room locations and properties. Once the simulation is started, the step method of each of the students is called. The student surveys an area around itself, depending on the step size, and determines which location in that area will bring it closer to its goal. If a student needs to switch floors, he looks for a nearby staircase. If a location that he is considering falls inside a classroom

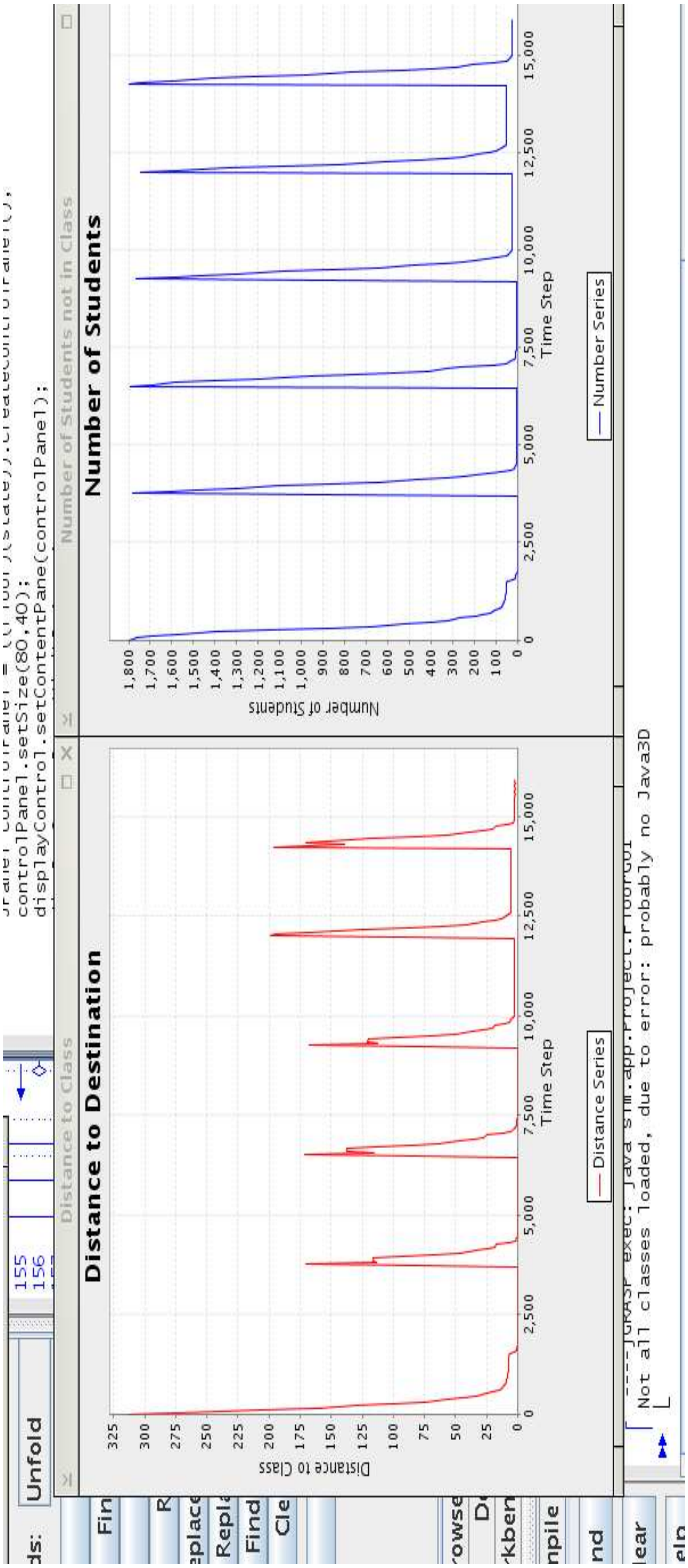

Figure 1: Student Distances to Class with Respect to Time

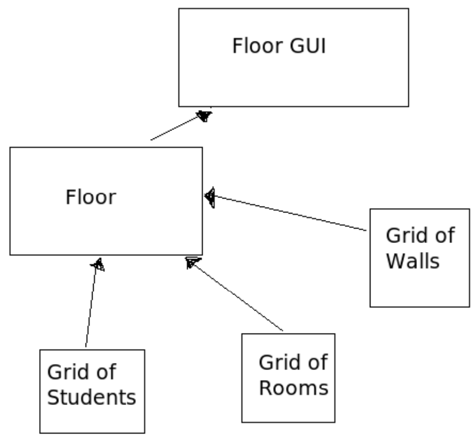

Figure 2: Diagram of class relationships

or invalid area, such a courtyard, that location is deemed invalid and the student moves on to considering the next location. At any point in time, the user has the option of pressing the fire drill button, which initiates a fire drill. Each room has an assigned area to which it goes during a fire drill, and students move to an area based on the assigned area of the room they are supposed to be in.

4 Quality Assessment

The characterists of the program that were tested were the accuracy of student movement and room placement, and the resemblance of school wide behavior to real life.

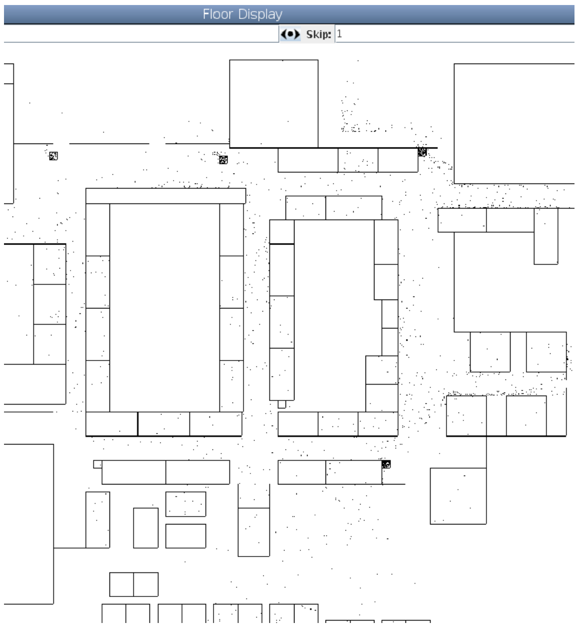

Figure 3: Sample run of the first floor

5 Results and Discussion

The point of the project has been to demonstrate that student behavior can be modeled through the use of a simple breadth first search. It means that generally when beings need to navigate through some form of maze or building when they know where their goal is, they decide where to go by means of what next step will get them closer to their goal. The project has been largely successful. The places in the school that should be the most crowded, such as hallway intersections, are indeed the most crowded. The fire drill works as planned; the students exit the building and stay there for a certain period of time until it is time to come back inside.

6 Conclusion

The purpose of this project was to develop a simulation that modeled student behavior at TJ. The goal was to show a typical day at TJ and demonstrate that like in the real world, trouble spots exist in the program. The fire drill simulation was an interesting insight into the way the fire drill runs in the real world. The simulation reflects the time that it takes students to get out of the building, because the school is empty in approximately four time steps.

References

[1] R. Axtell, “Why Agents? On the Varied Motivations for Agent Computing in the Social Sciences”, *Center on Social and Economic Dynamics*, November 2000.

[2] E. Bonabeau, “Agent-based modeling: Methods and techniques for simulating human systems”, *National Academy of Sciences of the USA*, 1999.

[3] C.M. Macal, M.J. North, “Tutorial on Agent-based Modeling and Simulation, *2005 Winter Simulation Conference*, 2005.

[4] M.W. Macy, R. Willer, “From Factors to Actors: Computational Sociology and Agent-Based Modeling”,*Center for Computational Analysis of Social and Organizational Systems*, August 2001.

[5] G. N. Beligiannis, C. N. Moschopoulos, ”Applying evolutionary computation to the school timetabling problem: The Greek case”, *Department of Computer Engineering and Informatics, Univeristy of Patras*, October 2006.