Proceedings of the 1998 Winter Simulation Conference
D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds.

APPLICATION OF A GENERAL PARTICLE SYSTEM MODEL TO MOVEMENT OF PEDESTRIANS AND VEHICLES

Lisa A. Schaefer
Gerald T. Mackulak
Jeffery K. Cochran
Jennifer L. Cherilla

Systems Simulation Laboratory, Modeling and Analysis for Semiconductor Manufacturing Laboratory
Department of Industrial and Management Systems Engineering
Arizona State University, Tempe, Arizona 85287, U.S.A.

ABSTRACT

This paper discusses the development of a series of models that ultimately leads to an unusual simulation approach to modeling automated guided vehicles on a factory floor.

A pedestrian model was initially created from ideas taken from some models of traffic and computer animation. The first stage of this model was a cellular automata model of pedestrians crossing a street. After a model of "beasts" was located on the internet, some modifications were made to the beast model to take advantage of the object-oriented features of the beast model to model the pedestrian scenario. Not only could the pedestrian model be integrated into a traffic simulation, but it could be extended to model robots on a factory floor. The algorithms developed for this research are an example of systems modeling in which a set of rules can be applied to a variety of systems by making simple modifications.

1 INTRODUCTION

In the early days of cartoons, paths of flocks or herds of animals were scripted, or placed frame by frame according to the whim of the animator. About a decade ago, the scripting paradigm changed due to advances in computer animation. An innovative approach to defining the paths of birds in a flock was invented by Craig Reynolds (Reynolds 1987). The flock is a simulated particle system in which birds are treated as individual objects. Each bird is an independent actor, or intelligent agent, that navigates according to rules on how to avoid nearby obstacles.

Particle system theory is applicable to many areas of mathematical and computer modeling. Models of insect swarms have been documented in the literature. Research on motor vehicles that drive themselves requires complex algorithms to calculate routes and prevent collisions in real time. Pedestrians and vehicles interacting at an intersection could be modeled as a particle system. A manufacturing application for particle system algorithms is a model of automated guided vehicles on a factory floor.

2 TRAFFIC SIMULATION

Individual particles in particle systems could be modeled as points or as cells which exist over a given amount of area. Sanford et. al. (1995) developed a preliminary model of vehicle traffic as points which uses the animation algorithms developed by Reynolds as a feasibility study. The model consisted of one vehicle which implemented behavioral rules within a "flock" of scripted point vehicles. Sanford et. al. determined it was indeed feasible to develop a traffic simulation model based on principles of complex systems rather than programming global rules to describe the traffic stream.

Currently available traffic software uses points in a procedural programming design to simulate vehicular traffic. The rules are based upon traffic flow theory. CORSIM (Kaman 1996) is a microscopic traffic simulation software developed by the Federal Highway Administration, written in FORTRAN. SimTraffic is another traffic simulation software, developed by Trafficware (Husch 1998). Advantages of these softwares are that they run on a PC, they models vehicles as individual entities, and they have the option to enter data with a graphical user interface or as a text file. Paramics by Quadstone is also available for modeling large traffic systems on Unix workstations (Quadstone 1998).

Although vehicles are considered individual entities, pedestrians are modeled by delaying vehicles at an intersection. This may be insufficient for modeling intersections with heavy pedestrian volumes.

One problem with modeling vehicles as points, rather than areas, existing in 2-D space can be detected by viewing the output with animation software. Even though the points representing vehicles do not collide, the areas over which the vehicles exist overlap. This problem is demonstrated by the screen shot in Figure 1.

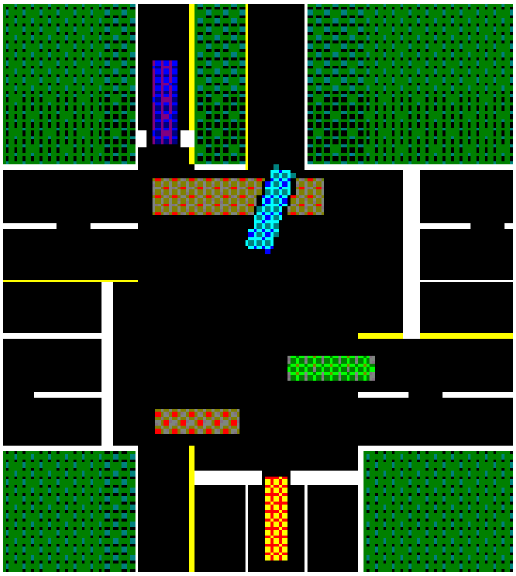

Figure 1: Screen Shot of Traffic Simulation Animation

Cellular automata models of traffic have been developed (Blue et. al. 1996, Nagel and Rasmussen 1995). The term "cellular automata" comes from defining discrete squares of space, or cells, and each cell has a set of rules that governs its state at each time step automatically. Since the cells occupy areas rather than points, the problem of vehicles colliding due to modeling them as points, rather than rectangles, is eliminated. Cellular automata models are characterized by four features:

1. size of the state space,
2. number of attributes per cell which define the state of the cell,
3. neighborhood of a cell, and
4. local rules.

Although cellular automata models account for vehicles existing over areas rather than points, they still do not account for pedestrians.

3 PEDESTRIAN MODEL

Schaefer and Cochran (1997) created a preliminary cellular automata model of pedestrians crossing at an intersection. The pedestrians were modeled with a procedural architecture in Visual Basic macros for Excel. A screen shot of the pedestrian animation is shown in Figure 2.

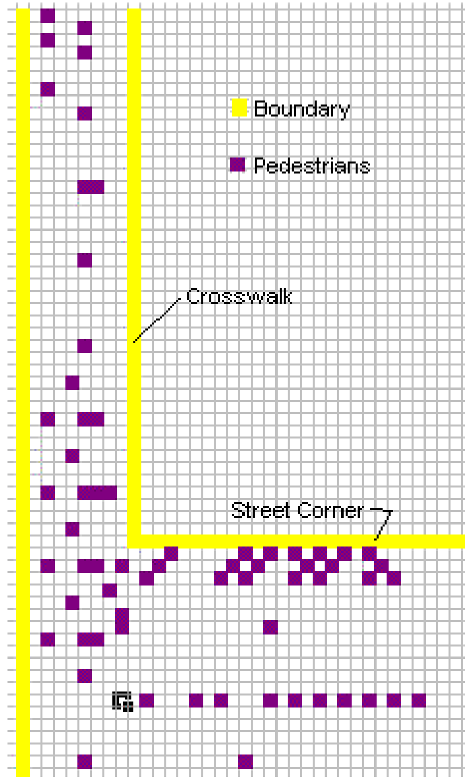

Figure 2. Screen Shot of Cellular Automata Pedestrian Animation

The algorithm for pedestrians crossing at an intersection is as follows.

```
Insert initial pedestrians into the system
For each signal cycle
{  Insert new pedestrians
   For each time unit in a signal cycle
   {  If less than 5 time units has elapsed during
        this cycle
      {  Insert pseudorandom pedestrians
         For each cell within the boundaries
         {  If pedestrian exists in cell
        {  While (no obstacles exist hgap +1 cells
               to the right, hgap < 5, & staying right of
               goal cell)
           {  Increment hgap    }
               While (no obstacles exist vgap +1 cells
               up from cell hgap cells to the left, and
               vgap < 5)
               {  Increment vgap    }
           Place pedestrian hgap cells to the left
               and vgap cells up
}  }  }  }  }
```

Basically, pedestrians move toward a goal which is first the crosswalk, then the other side of the street, while avoiding collisions by stopping behind the other pedestrians that are immediately to their left or immediately above them.

Pedestrians can be modeled easily with cellular automata, however the difficulty with integrating pedestrians into a cellular automata model of traffic is the difference in area that a vehicle occupies versus the area that pedestrians occupy. Even if vehicles are modeled as many-celled objects, the difficulty of modeling turning vehicles arises since turning vehicles do not occupy rectangular space parallel to roadways in real world systems. Some type of diagonal rectangular space must be defined or approximated.

4 BEASTS

A herd of beasts can be simulated as a particle system. Source code for a Java applet of an animation of a herd of beasts was downloaded from the internet (Brogan 1998). The beast animation is modeled as an object-oriented system. The beasts may represent anything from a herd of buffalo to a crowd of people walking in a pedestrian mall. Figure 3 is a screen shot of the animation created by Larry O'Brien.

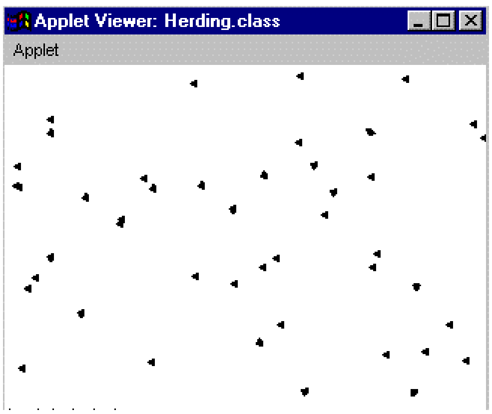

Figure 3. Screen Shot of Animated Beasts

The algorithm for the navigation of the beasts was adapted from the flocking rules found in Reynolds, 1988. The rules are shown below.

For each time step:
Collect locations, speeds, & angular direction of beasts in "sight" range
For each beast apply Reynolds's rules:

Rule 1) Match average speed of those in the list of nearby beasts
Rule 2) Move towards the perceived center of gravity of the herd
Rule 3) Maintain a minimum distance from those around you
Update x & y coordinates according to beast's speed and direction

Since the beasts are modeled as points rather than existing over an area, rule 3 prevents the perimeter of a beast from colliding with the perimeter of another beast or obstacle.

5 BEASTLY PEDESTRIANS

By changing the rules of the Beast model, the simulation can be transformed to resemble the pedestrian model. A screen shot of the pedestrians is shown in Figure 4.

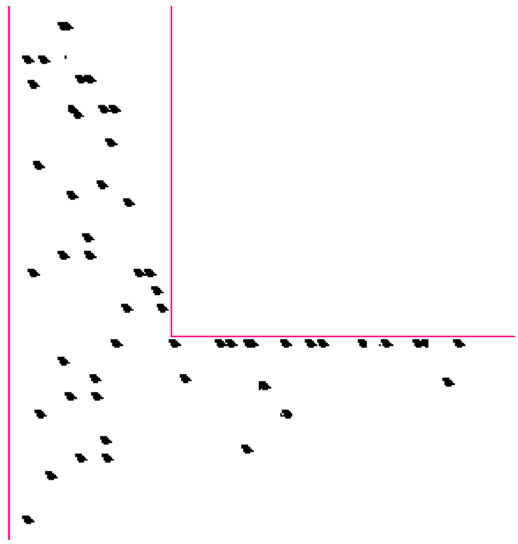

Figure 4: Screen Shot of Object-Oriented Pedestrian Animation

The main difference between this model and the cellular automata model is that pedestrians in this model avoid collisions with other pedestrians by adjusting their speed according to locations, speeds, and angular direction of pedestrians in "sight" range. Sight range is defined by a distance and an angle of vision. In the cellular automata model, pedestrians avoid collisions by stopping behind the other pedestrians that are immediately to their left or immediately above them.

The pedestrian simulation was created by removing Reynolds's navigation rules of the "Beast" code and substituting the following navigation rules.

Rule 1) Maintain a desired speed
Rule 2) Move towards a "goal" defined as the other side of the street

Rule 3) Maintain a minimum distance from those around you
Rule 4) Mom's Rule: Stay out of the street

Rule 4, "Mom's Rule," causes pedestrians that would otherwise move to a forbidden zone, in this case the street, to move along the edge, or sidewalk.

The advantage of using an object-oriented approach to model pedestrians is that each object can have an attribute, or variable, which defines its size and shape. Thus pedestrians and vehicles can easily be differentiated from each other. Each object may also have its own set of rules by which it navigates around its environment. The rules are easy to change since each rule consists of only a few lines of code.

6 AUTOMATED GUIDED VEHICLES

Another advantage of object-oriented programming is the ability to insert different types of objects in the simulation program. The pedestrian rules can be inserted in a model of free-range Automated Guided Vehicles (AGVs) on a flow shop floor. A screen shot of the AGVs is shown in Figure 5. The rectangular figures are the machines to which the AGVs are transporting objects (jobs) that need to be processed.

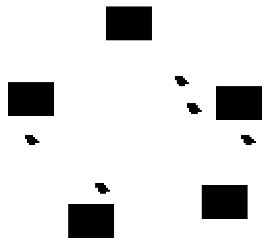

Figure 5: Screen Shot of AGVs

The "goal" mentioned in Rule 2 of the pedestrian simulation becomes the next machine in the job's sequence rather than the other side of the street. Instead of staying out of the street as in Rule 4, forbidden areas may be defined as the space over which machines exist. Rule 4 could become:

Rule 4) Stay within the designated pathways

The pathway would be the maze of floorspace around machines and other obstacles.

Instead of only AGVs being objects, jobs are also objects. This becomes a scenario in which jobs with due dates are dynamically scheduled to be transported by AGVs to machines. The intention of the simulation is to minimize the sum of the tardiness of all jobs. The flowchart for the logic applied to each AGV is shown in Figure 6.

7 CONCLUSION

With simple modifications, a model of "beasts" can be used to model pedestrians on a street corner. The pedestrian model can be integrated into a traffic simulation software to determine feasible signal timing strategies at intersections with heavy pedestrian volumes. The rules for pedestrian navigation can also be used as navigation rules for AGVs on a factory floor.

When creating a model of a particle system, one must be careful when choosing the manner in which to represent entities. If the entities are modeled as points but they represent areas in the real system, the rules must account for appropriate collision detection, otherwise the simulation results may not be valid. If the entities are modeled as cells, the rules may need to account for entities of varying sizes.

The rules used for the simulations described in this paper are not yet designed to accurately model pedestrians or AGVs, rather to show that systems which seem unrelated at first glance can be modeled with the same simulation program by simply changing the rule sets. The authors intend to continue work in this area by improving the parameters and rule sets.

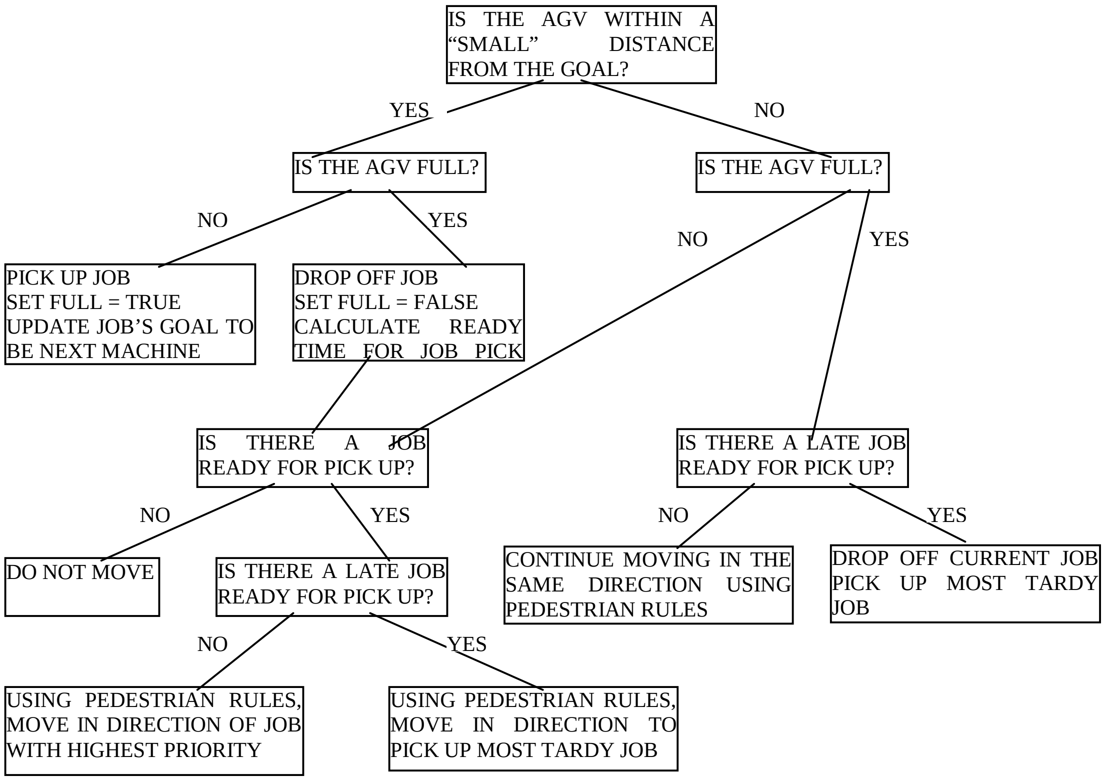

Figure 6: Flowchart for AGV Simulation

ACKNOWLEDGEMENTS

This work was partially supported by the Federal Highway Administration Office of Advanced Research.

REFERENCES

Blue, V., F. Bonetto, and M. Embrechts. 1996. A Cellular Automata Model of Vehicular Self-Organization and Nonlinear Speed Transitions. *Proceedings of the Transportation Research Board 75th Annual Meeting*. Paper No. 961336.

Brogan, D. 1998. *Artificial Life and Flocking/ Herding/Schooling Behavior* [online]. Coded by Larry O'Brien, code no longer available, web site still exists `http://www.cc.gatech.edu/gvu/People/PhD/David.Brogan/herding.html` [web site accessed June 11, 1998].

Husch, D. 1998. `SimTraffic` [software online]. Trafficware, Berkeley, California. Available as `http://www.trafficware.com` [accessed June 11, 1998].

Kaman Sciences Corporation, 1996. *TSIS User's Guide Version 4.0 beta.*

Nagel, K., and S. Rasmussen. 1995. Traffic at the Edge of Chaos. *Transportation Science*. **28 (2)**, 222-235.

Quadstone. 1998. `Paramics` [software online]. Quadstone, United Kingdom. Available as `http://www.quadstone.com`
[accessed June 11, 1998].

Reynolds, C.W. 1987. Flocks, Herds, Schools: A Distributed Behavioral Model. *ACM SIGGRAPH Computer Graphics*. **21 (4)**, 25-34.

Sanford, K. L., J. A. Wentworth, and S. McNeil. 1995. A-Life in the Fast Lane. *Proceedings of the AI 1995 15th International Conference*. 431-440.

Schaefer, L., J. K. Cochran. 1997. *The Development of a Framework for Particle motion Laws in Support of an Emergent Behavior Model of Pedestrians and Vehicles Interacting at an Intersection*, Final Report to the Federal Highway Administration, United States Department of Transportation, December 1997.

AUTHOR BIOGRAPHIES

LISA A. SCHAEFER is a Ph.D. student in the Industrial & Management Systems Engineering Department at Arizona State University. She received her B.S.E. and M.S. in civil/ transportation engineering from Arizona State University. Ms. Schaefer has performed research on intelligent agents in manufacturing systems at Wright Patterson Air Force Base. She is currently supported by a research grant from the Federal Highway Administration to explore the potential for simulating pedestrians and traffic as intelligent agents.

GERALD T. MACKULAK received his B.Sc., M.Sc., and Ph.D. degrees from Purdue University in the area of Industrial Engineering. He has been at Arizona State University since 1980 were he is currently an Associate Professor of Engineering and Co-Director of the Systems Simulation Laboratory (SSL). The SSL was founded in 1985 with the charter of extending the methodology of simulation to a broader user base. Major funding for the SSL has been provided by Motorola, Hughes, McDonnell Douglas, IMAR, NSF, and NIST. Dr. Mackulak has held positions with U.S. Steel, Burroughs Corporation, and Pritsker and Associates as well as consulting for over 50 national companies. His primary areas of expertise are in the modeling of manufacturing systems, CIM analysis methodologies, and production control. He has published over 75 articles and has taught over 25 seminars on simulation methods, languages, and IDEF modeling techniques. Since 1990 he has received two teaching awards for excellence in both graduate and undergraduate instruction.

JEFFERY K. COCHRAN is Associate Professor of Industrial & Management Systems Engineering at Arizona State University, where he is also a Director of the Systems Simulation Laboratory. Dr. Cochran was educated at Purdue University (BSE 73, MSNE 76, MSIE 82, and PhD 84). His industrial experience includes positions at Battelle Northwest Laboratory, Los Alamos National Laboratory, and NASA's Laboratory for Applications of Remote Sensing. Professor Cochran has directed research projects for the National Science Foundation's I/UCRC for Manufacturing and Automation, the National Institute of Standards and Technology, and the Institute for Manufacturing and Automation Research. His primary research interests are simulation modeling and analysis, computer integrated manufacturing, operations research, and artificial intelligence. Dr. Cochran is the author of over sixty scholarly publications, and is a Professional Engineer.

JENNIFER L. CHERILLA is an M.S.E. student in the Industrial & Management Systems Engineering Department at Arizona State University. She received her B.S. in mathematics from John Carroll University in Cleveland, Ohio. Ms. Cherilla's area of emphasis is operations research, and she is currently working on simulation projects at the Intel Corporation.