EpiViz: A Visual Simulation of an Epidemic Model using a Cellular Automaton

Matthew J. Farmer* and Tina V. Johnson
Department of Computer Science
Midwestern State University
Wichita Falls, TX 76308
matthew.farmer@mwsu.edu, tina.johnson@mwsu.edu

***Abstract*— Cellular Automata (CAs) are often used to simulate complex systems. One such application of a CA is to simulate the spread of disease through a susceptible population. This paper describes EpiViz, a CA-based epidemic simulator which uses a variety of input parameters, such as probability of infection, infectious period, vaccination rate, mode of infection, and probability of recovery to influence the resulting state of each entity in a CA over the course of a simulated disease outbreak. EpiViz then produces an animated representation of the outbreak. EpiViz allows a user to set disease related parameters prior to running a simulation to enable experimentation with various disease factors and to observe the effects on a susceptible population. Incorporating the traditional Susceptible-Infected-Removed (SIR) disease model, EpiViz is able to visually display an outbreak on a day-to-day basis.**

Keywords—cellular automata, computational epidemiology, SIR model, disease simulation

Submitted to: MSV '13

I. INTRODUCTION

Cellular automata (CAs) are powerful models for simulating discrete systems. John von Neumann is credited with defining a CA as a self-reproducing system in which each cell can be in one of several distinct states and the state of each cell as time progresses is dependent upon its own state and the states of its neighbors at the previous time-step [1]. Although CAs are simplistic in design, they have proven to successfully model diverse, complex systems, including power consumption [2], pedestrian flow [3], wildfire spread [4], and areas in applied physics [5] [6].

CAs are particularly well-suited for modeling epidemic behavior since the disease state of an individual (susceptible, infected, etc.) is dependent upon its own state as well as the state of its neighbors. Several researchers have developed CA models to investigate disease propagation [7] [8] [9] [10]. This paper presents an epidemiological CA model, EpiViz, which allows variation of disease parameters and visual animation of a simulated outbreak.

II. MODEL DESCRIPTION

EpiViz is a CA that was developed using the Susceptible-Infected-Removed (SIR) paradigm. SIR is a widely accepted mathematical model for simulating disease spread [11] [12]. In the basic SIR model, all individuals in a closed population belong in one of three states: Susceptible (S), Infected (I), or Removed (R). Individuals may move from S to I based on a transmission probability and from I to R based on a removal rate. Other states may be added to the basic SIR model to more accurately represent disease dynamics.

EpiViz was created in C++ based on the following considerations and rules:

- The grid is a two-dimensional array with a dimension of n x n entities.
- Each cell represents a single entity which is capable of being in one of the following five states:
  - Susceptible: The default state of all entities; vulnerable to infection. Depending on the disease, it may be possible for an infected individual to return to the susceptible population if immunity is disallowed.

*Contact Author

- Infected: The entity is harboring the disease and is able to infect other entities. After the infectious period has elapsed, the entity will either become immune, return to the susceptible population (if immunity is not possible), or die. Infected is a temporary state.
- Immune: The entity was once infected but is no longer infectious and cannot die, receive vaccinations, or be infected again. Immune is a permanent state.
- Deceased: The entity was once infected, but has died and is no longer able to infect others or be infected. Deceased is a permanent state.
- Vaccinated: The entity is immune to infection permanently; only susceptible entities may be vaccinated. Vaccinated is a permanent state.

• Contacts are established using the Moore Neighborhood, which includes top, bottom, left, right and diagonal neighbors. Only infected and vaccinated entities may affect the state of nearby entities.

• Each infected or vaccinated entity has a probability of changing the state of individuals within its Moore neighborhood. An entity in either of the aforementioned states will have the opportunity to cause anywhere from 0 to 8 entities to change states in a given day. Therefore, the likelihood of a state change occurring is greater if more of the surrounding entities are either infected or vaccinated.

• If entities are capable of traveling, each infected entity has an additional opportunity per day to infect a random entity anywhere in the grid. Vaccinated entities are not capable of travel.

• When an entity becomes newly infected, the entity is not infectious until the following day. The same applies for newly vaccinated entities.

• The grid allows wrapping; entities on the borders of the grid may affect entities on the opposite side of the grid. In this way, the grid is less like a flat square and more like a sphere.

• The simulation ends when either no entities are infected or two hundred days have elapsed.

When an EpiViz simulation begins, the user is greeted with a menu that allows selection from a predefined list of diseases to simulate. Each disease has the following characteristics: disease name, length of infection, probability of infecting other entities, probability of killing an infected entity, probability of an infected person traveling, whether immunity is allowed, and which day of the simulation vaccines are made available to the population. In addition, the user may create a custom disease to simulate by supplying the various characteristic values.

Once a disease has been selected, EpiViz will show the user all of the characteristics associated with the chosen disease and will ask for confirmation to begin the simulation. EpiViz provides the ability to conduct multiple simulations on the same data set to average the results. Once the animation is complete, a CSV file with average population data for the various states by day is produced. The user is returned to the main menu from which another simulation may begin.

Initially, a single entity is randomly chosen to be the first infected. The infected entity has the opportunity to infect others, and the day is concluded. Each day, the infected entities have the opportunity to infect other entities. Some diseases may have a day in which vaccinations are made available. On the predetermined vaccination day, four random susceptible entities are selected for vaccination. From each day forward, all vaccinated entities have the

opportunity to vaccinate susceptible entities within the Moore Neighborhood (vaccinated entities may not travel). Vaccinations cannot cure an infected entity. Only on the first available day of vaccine availability are four random entities chosen for vaccination.

The grid consists of entities that are capable of knowing their current state and how long they have been infected, if applicable. Once the infection period has fully lapsed, the entity must change to one of three states: susceptible (only if immunity is not allowed), immune, or deceased.

At the conclusion of each day within the simulation, a new frame of the animation is rendered which consists of *n* x *n* cells. Each cell is filled with a color that represents the current state of a given entity within the grid (see Table 1). The animation visually demonstrates how the disease spreads each day. The CSV file contains the average state population data for each day of the user-specified number of simulations. The CSV data is used to create the SIR Graph.

A spreadsheet program, such as Microsoft Excel or Apple Numbers, can open the CSV file to produce a table of values and create a line graph similar to the one shown in Section IV.

III. Algorithm Overview

Initially, a two dimensional array is allocated and initialized with an entity object at each index. Each entity object consists of attributes used for storing its current state and length of time in the infected state. Next, an entity within the array is randomly chosen to enter the infected state, and is placed in a first-in first-out (FIFO) infected queue. For each day of the simulation, each entity in the infected queue has the opportunity to infect eight neighboring individuals and one distant individual, if traveling is allowed. After an entity has had an opportunity to infect others, its days-infected count is incremented by one, and the next individual in the queue is allowed to attempt infecting others using the same method.

Table 1: Color/status Association

Status (Color)	
	Susceptible (yellow)
	Infected (green)
	Immune (blue)
+	Vaccinated (red)
	Dead (black)

If an infected entity successfully spreads the infection, the newly infected individual is placed in the FIFO queue reserved for recently infected entities, but not in the infected queue. Two queues are necessary because newly infected individuals must not be allowed to spread their infection the same day their infection occurs, else the entire population may succumb to infection within one simulated day. After all infected entities have completed their attempts to infect others, the recently infected entities are merged at the tail end of the infected queue. A simulation day concludes once the two queues are merged, a new frame of an animated grid is generated based upon the current status of each entity in the array, and population totals for each possible state are recorded. The process then repeats until no entities are in the infected queue or two hundred simulation days have been processed.

Given that the infected state is temporary, the user-defined days in which an entity may be in the infected state dictates how long an individual remains in the infected queue. After an entity has attempted to infect all other entities within reach, its days-infected counter is incremented, indicating that the time an entity stays in the infected state is limited. When the infection counter has reached its limit, the entity must enter either a removed or susceptible state depending upon the parameters specified for the disease

by the user. When transitioning to the removed state, entities which were once infected enter either the immune or deceased state. In the event both death and immunity are allowed by the simulation, a random number is generated to determine which of the two removed states an entity enters based upon probability parameters for both states provided by the user.

IV. Example Simulation

For the example simulation, a custom disease was created to demonstrate parameter specification and user interaction with EpiViz. Selecting the custom disease option menu allowed the user to choose the parameters shown in Fig 1.

The animation created by this example run of EpiViz reveals that the disease spread slowly at first, but quickly began to infect the entire susceptible population. Because vaccinations were not made available until day 25, the disease had already spread too far for the vaccination stations to be created. Four of the thirty-three tables are shown in Fig 2. The graphical table for the last day of the simulation indicates that all entities were either immune or deceased.

The CSV file was used to generate the SIR curve as shown in Fig 3. The line graph shows a pattern that is characteristic of an SIR curve. The susceptible population begins high, and then begins to drop as the removed population begins to increase. The infected population starts low, peaks midway, and dies out at the point which the population of

```
+--------------------------+
|                Name: Silly_Pox
|        Days Infected: 6
|  Immune After Removed: 1
| Infection Probability: 20%
|       Death Probility: 5%
|    Travel Probability: 10%
|  Vaccinations Allowed: 1
|Day Vaccinations Begin: 25
|   Vaccine Probability: 0.3
+--------------------------+
```

Fig. 1. Parameter selection of a custom disease

immune entities exceeds the population of susceptible entities. The graph does not explicitly show the removed population, but instead shows the immune, deceased, and vaccinated entities from which the removed population is composed.

V. Future Work

EpiViz was created to provide a CA model for epidemic simulation in a manner that provides a visual representation of an outbreak, input parameter customization, and outbreak data collection. While EpiViz meets that goal, there are areas for improvement. The following ideas have been identified for future enhancements of EpiViz:

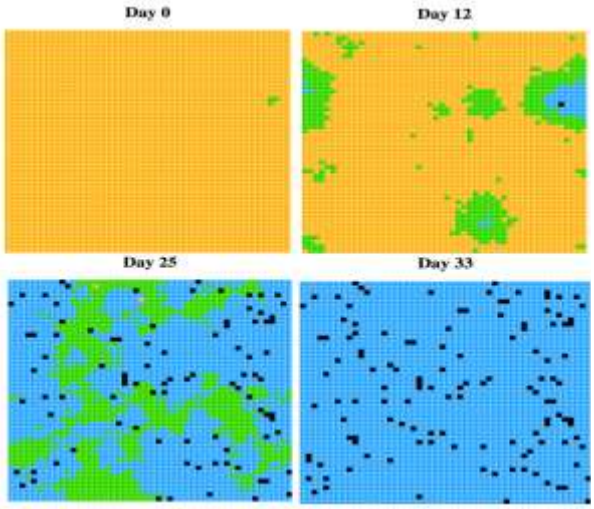

Fig. 2. Sample screenshots from an EpiViz simulation.

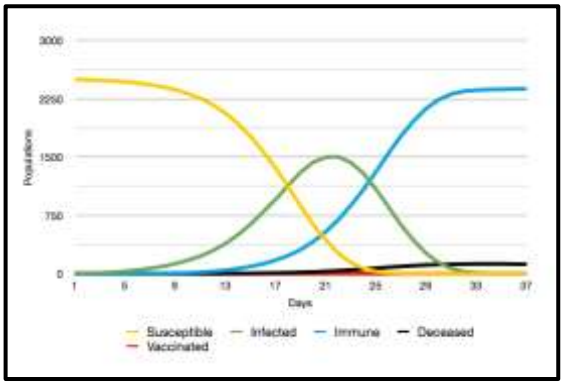

Fig. 3. Outbreak curve from an EpiViz simulation.

- Modify the artificial intelligence of the vaccinated entities to better combat the spread of diseases.
- Allow some cells in the grid to represent flora, structures, water, or other objects that are not affected by infections but instead, provide natural barriers that can alter the spread of a disease.
- Include a latent period to enable SLIR Model data collection.

VI. Conclusion

The SIR mathematical disease model can be effectively implemented using a CA approach. EpiViz is an epidemic simulator that allows a user to specify several disease related parameters and then models an outbreak using a set of rules for each entity in the CA. The simulation begins with one individual in the infected state and all others in the susceptible state. The outbreak proceeds as infected entities probabilistically infect susceptible neighbors. The simulation continues for two hundred days or until there are no longer any entities in the infected state. The primary benefit to EpiViz is the ability to visualize the day-to-day statistics along with data collection.

VII. References

[1] C. E. Shannon, "Von Neumann's contributions to automata theory," *Bull. Amer. Math. Soc.,* vol. 64, no. 3, pp. 123-129, 1958.

[2] G. C. Sirakoulis and I. Karafyllidis, "Cellular Automata and Power Consumption," *Journal of Cellular Automata,* vol. 7, pp. 67-80, 2012.

[3] P. Zhang, X.-X. Jian, S. C. Wong and K. Choi, "Potential field cellular automata model for pedestrian flow," *Phys. Rev. E,* vol. 85, no. 2, Feb 2012.

[4] G. A. Trunfio, D. D'ambrosio, R. Rongo, W. Spataro and S. Di Gregorio, "A New Algorithm for Simulating Wildfire Spread through Cellular Automata," *ACM Transactions on Modeling and Computer Simulation,* vol. 22, pp. 1-26, Dec 2011.

[5] I. Amlani and A. O. Orlov, "Digital logic gate using quantum-dot cellular automata," *Science,* vol. 284, no. 5412, p. 289, 1999.

[6] P. D. Tougaw and C. S. Lent, "Dynamic behavior of quantum cellular automata," *Journal of Applied Physics,* vol. 80, no. 8, 1996.

[7] A. R. Mikler, V. Sangeeta and K. Abbas, "Modeling infectious diseases using global stochastic cellular automata," *Journal of Biological Systems,* vol. 13, no. 4, pp. 421-439, 2005.

[8] S. C. Fu and M. George, "Epidemic modelling using cellular automata," *Proceedings of the Australian Conference on Artificial Life,* 2003.

[9] H. F. Gagliardi and D. Alves, "Small-world effect in epidemics using cellular automata," *Mathematical Population Studies,* vol. 17, no. 2, pp. 79-90, 2010.

[10] K. M. Bohannon and T. V. Johnson, "Cellular Automaton as an Epidemiological Model: A New Twist on Old Ideas," *In Proceedings of CAINE,* pp. 127-131, 2010.

[11] E. Allman and R. J., Mathematical models in biology, an introduction, New York, NY: Cambridge University Press, 2004.

[12] R. M. Anderson and R. M. May, Infectious diseases of humans, Oxford, NY: Oxford University Press, 2006.