Carleton University

Department of Systems and Computer Engineering

Modelling Discrete-Event Systems Using Cell-DEVS (2016 Fall)

Assignment 2

Food Chain

Name : Yamini Nibhanupudi

Student ID : 101050830

Email ID : yamininibhanupudi@cmail.carleton.ca

Part I

The CELL DEVS model in this assignment will be dealing with a very simple Food Chain. The Food chain has only 4 participants, namely, bear, fox, rabbit and vacancy. The basic idea is based on the paper “A Minimal Spatial Cellular Automata for Hierarchical Predator-Prey Simulation of Food Chains” by K.A. Hawick and C.J. Scogings.

We develop a cellular automata model of a hierarchical predator-prey system and show that even a minimal automaton is able to capture the dynamical behaviour of real physical systems.

We are primarily exploring predator-prey and predator predator-prey behaviours as they might pertain to realistic spatially distributed participants in a food chain.

Part II

**INTRODUCTION**

Cellular Automata (CA) models often provide a simple yet insightful encapsulation of the key microscopic behaviours of a complex system. CA’s have been used to study: system dynamics, growth, emergence, chaos, randomness, complexity and other whole system properties such as the eﬀect of diﬀusion. CA’s models can be explored through explicit enumeration of deterministic rules and behaviours and explicit starting conditions or by using methods such as statistical mechanics, whereby computer generated random numbers provide diﬀerent starting conditions or probabilistic pathways through diﬀerent deterministic rule sets.

The values that can occupy the cell at any time are 4, 3, 2 and 1. In cell-DEVS, these are represented by entities as given below:

4 – Bear

3 – Fox

2 - Rabbit

1 – Vacancy

The simulation is done based on certain rules. These rules are given in the paper mentioned above.

**CA RULES**:

The rules governing this model are given below in the form of an algorithm (figure 1).

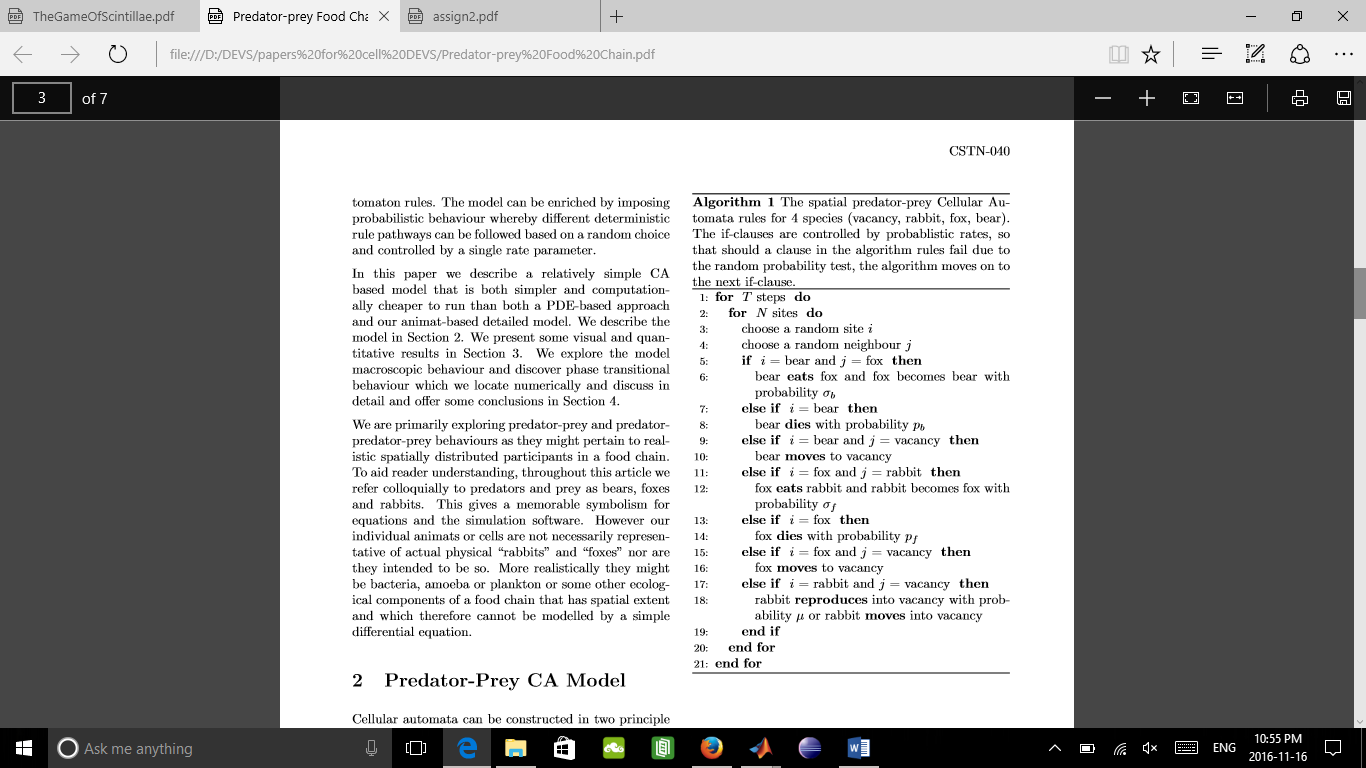


Figure 1: CA Rules

If none of the rules above are satisfied, then the present value in the cell is retained.

Figure 2 shows the von Nuemann neighbourhood as the results for the square 4 nearest neighbour case only is given in the paper.

Figure 2: von Neumann square 4 nearest neighbourhood

(0,1)

(0,-1)

(-1,0)

(1,0)

(0,0)

**Cell-DEVS Formalism**

M = < Xlist, Ylist, I, X, Y, η, N, {r, c}, C, B, Z, select >

Xlist = {4,3,2,1}

Ylist = {4,3,2,1}

I = <Px, Py> Px={Ф} , Py ={Ф}

X = Ф

Y = Ф

η = 5

N = {(-1, 0), (0, -1), (0, 0), (0, 1), (1, 0)}

r = 20

c = 20

S = {0, 1}

B = Ф (wrapped)

Pij Y1 → Pi,j-1 X1 Pi,j+1 Y1 → Pij X1

Pij Y2 → Pi+1,j X2 Pi-1,j Y2 → Pij X2

Pij Y3 → Pi,j+1 X3 Pi,j-1 Y3 → Pij X3

Pij Y4 → Pi-1,j X4 Pi+1,j Y4 → Pij X4

Pij Y5 → Pij X5 Pij Y5 → Pij X5

SELECT = {(-1, 0), (0, -1), (0, 0), (0, 1), (1, 0)}

**EXECUTION AND RESULTS**

The test cases are such that the cell space initially looks like figure 3 below:

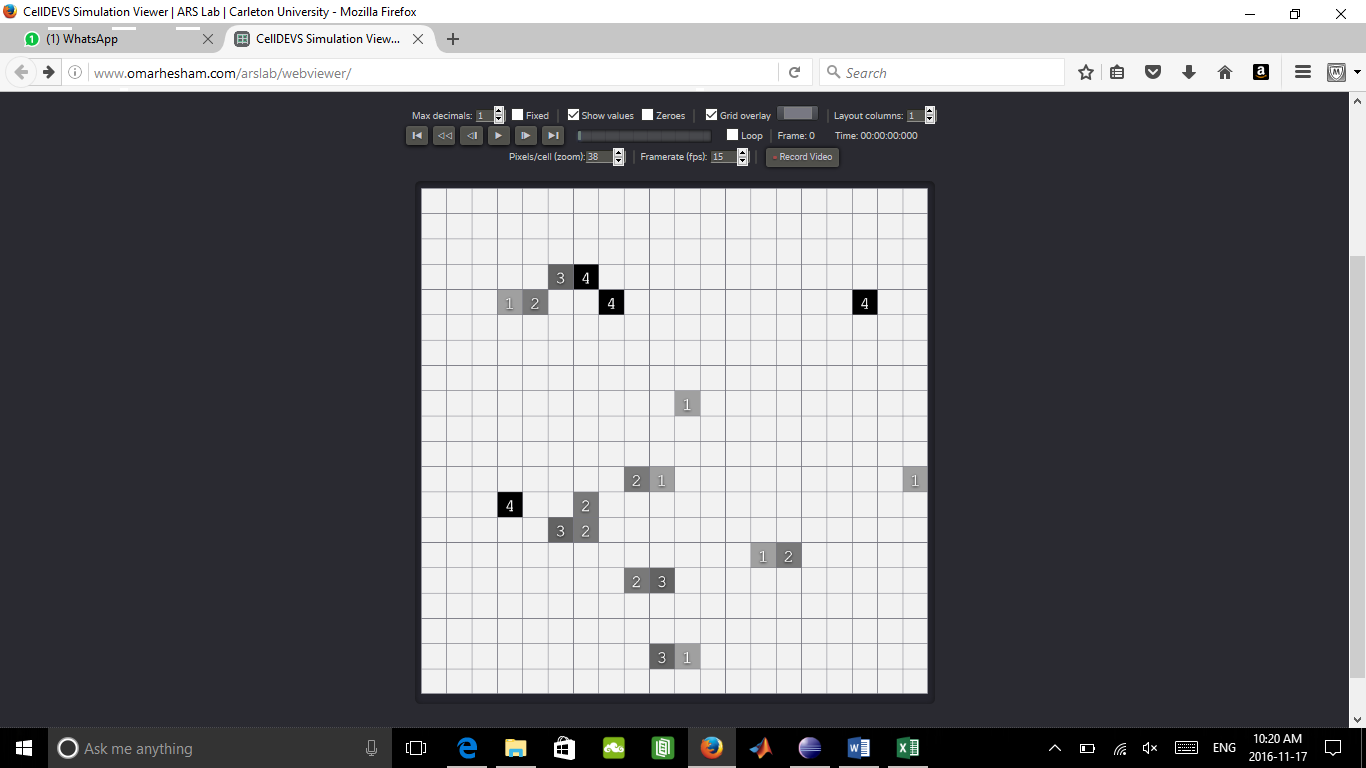


Figure 3: Initial cell space

When the model is run on <http://www.omarhesham.com/arslab/webviewer/> , we get the results as below, in three frames.

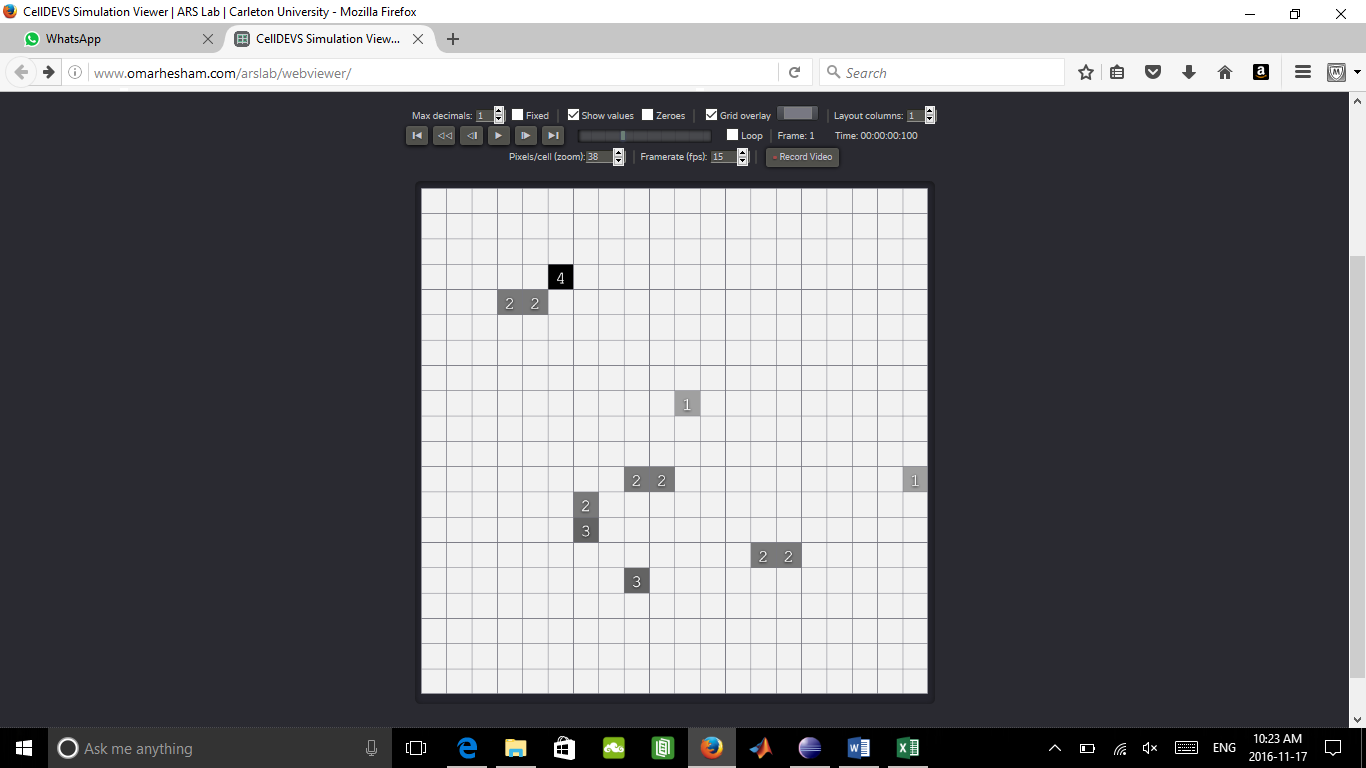


Figure 4: frame 1

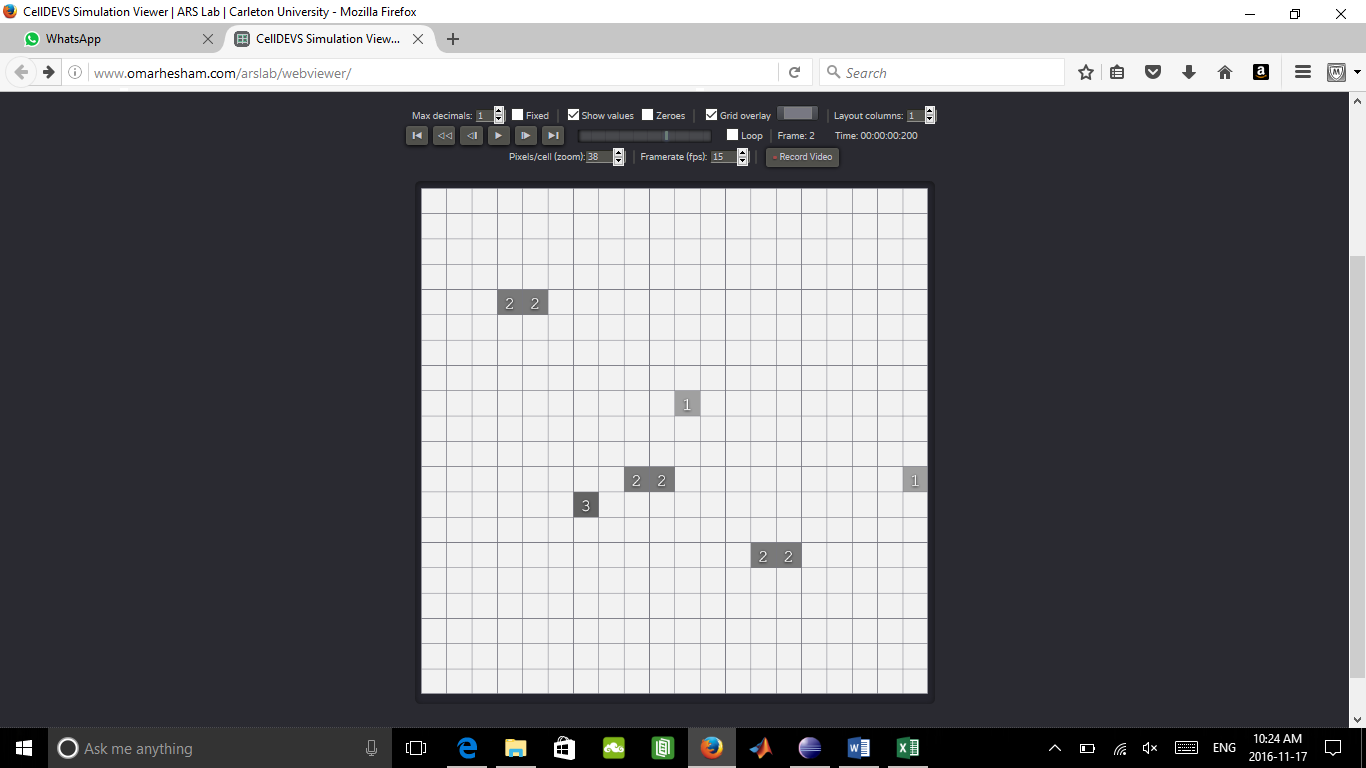


Figure 5: frame 2

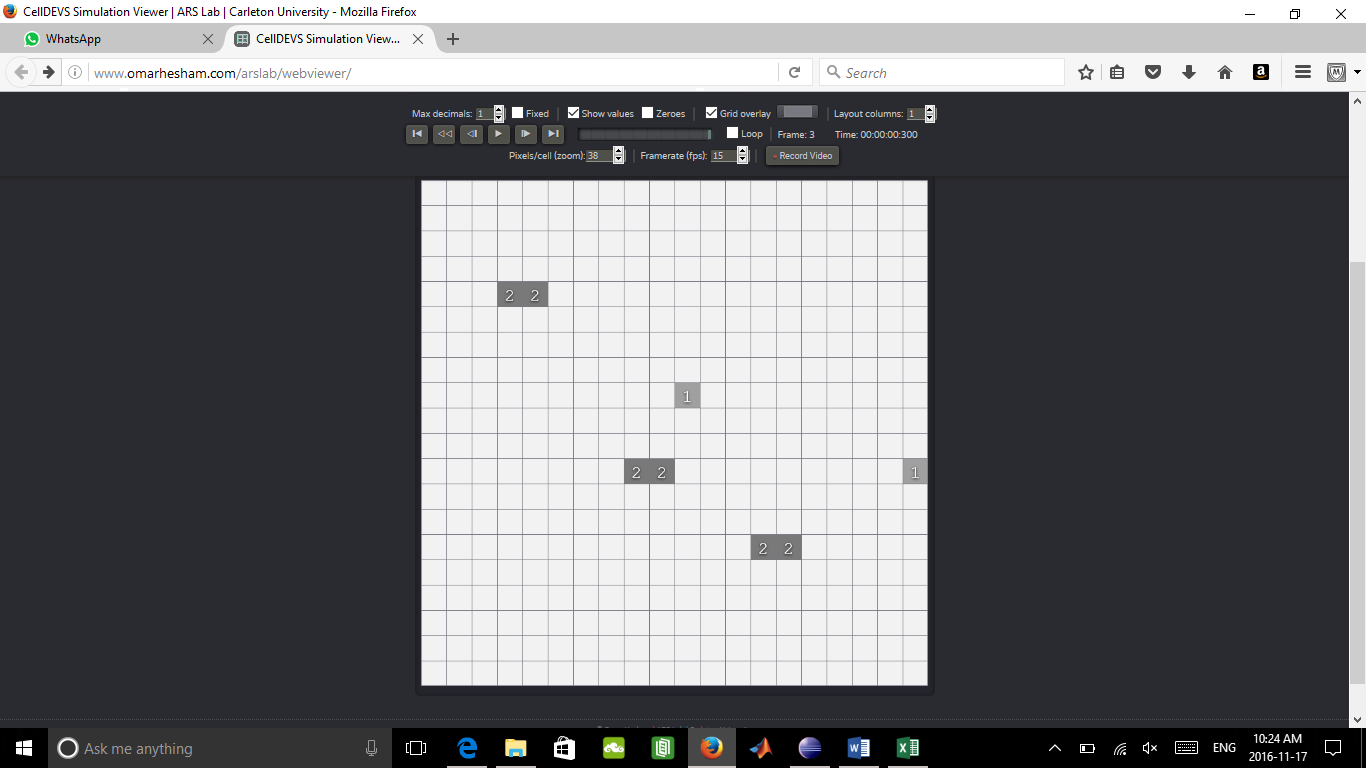


Figure 6: frame 3

The bear eats the fox, the fox eats the rabbit, and the rabbit produces into the vacancy. In any other conditions, the present values in the cell is retained.

But, an ambiguity arises when the model is designed this was as some cases will not be satisfied as the simulator does not know which one has more priority. Hence, we could use probability in that case.

**CONCLUSION**

From the above frames, we can say that the model is working perfectly. Hence, the food chain simulation model has been successfully simulated.