**Bi-Direction Pedestrian Movement with Cell-DEVS  
Assignment 2**

By : Ronnie Farrell

Student # : 100717103

To : Prof. Wainer

Course : SYSC 5104

Date : November 15, 2011

**Introduction**

The paper that I chose for this assignment is “Simulation of bi-direction pedestrian movement using a cellular automata model”. The paper presents a cellular automata model to simulate the bi-direction pedestrian movement.

The model is described in the two-dimensional system. The underlying structure is a W × W cell grid, where W is the system size. Each cell can either be empty or occupied by exactly one pedestrian. The size of a cell corresponds to approximately 0.4 × 0.4 m2. This is the typical space occupied by a pedestrian in a dense crowd.

The update is synchronous for all pedestrians. Each pedestrian can move only one cell per time-step. Since, the average velocity of a pedestrian is about 1.00 m/s, one time-step in this model is approximately 0.4 s, and it is of the order of the reaction time.

In the paper, the Von Neumann neighbourhood is used (ie. can move left/right/up/down). In the model, there are two components of walkers including the up walkers moving from the bottom to the upper boundary and the down walkers moving from the upper to the bottom boundary. The model uses a wrapped boundary.

Since the update is synchronous for all pedestrians at every time-step, there exist two problems at every time-step. The first problem that needs to be resolved is the route choice. Figure 1 shows all the possible configurations that the paper describes in which the up walker may encounter.



Figure : Pedestrian Behaviour

In this model only one cell can be occupied by one pedestrian at a time-step. This brings us to the second problem that needs to be resolved, the confliction when more than one pedestrian is vying for a cell. To solve this problem of collision, the neighbourhood will be increased from the proposed Von Neumann neighbourhood to a larger neighbourhood as shown in figure 2.

**Rules:**

This section will define the behaviour of a pedestrian. All of the rules below are pseudo code specified for the up walker. The same logic applies to the down walker. To make the simulation more interesting and realistic, I have added obstacles as a third component to the simulation.

**Rule 1 – No walker/obstacles ahead, move forward**

The upper adjacent cell is unoccupied, the up walker will select it to move into whether his left and/or right adjacent cells are occupied or not. To avoid two pedestrians colliding, this rule will be extended to: move to the upper adjacent cell if the upper adjacent cell is unoccupied, and the upper adjacent cell to that one is not a down walker. This allows the up walker to move forward one cell if an obstacle lays in the cell two spaces ahead. This rule has the highest priority.

**Rule 2 – Walker/obstacle ahead and right side available, move to the right side**

The upper adjacent cell is occupied by another walker or obstacle, move to the right adjacent cell if that cell is unoccupied and another walker isn’t going to take that cell in the next time-step. This rule has the second highest priority.

**Rule 3 – Walker/obstacle ahead and to the right, move to the left side**

The upper and right adjacent cells are occupied by other walkers or obstacles, move to the left adjacent cell if that cell is unoccupied, and another walker isn’t going to take that cell in the next time-step. This rule has the third highest priority.

**Rule 4 – Move right if two walkers are vying for the same cell**

When an up walker and a down walker are trying to walk on the same cell at the in the next time-step, both walkers move to the right adjacent cell if that cell is unoccupied, and another walker isn’t going to take that cell in the next time step to avoid collision. If the right cell is occupied, or another walker is going to take that cell in the next time step, don’t move. This rule has the fourth highest priority.

**Rule 5 – Default**

The walker does nothing and waits where they are.

**Cell-DEVS Formal Specification**

**Neighbourhood**

Figure 2 is a graphical representation of the neighbourhood. As mentioned, the neighbourhood had been increased from the Von Neumann neighbourhood to the neighbourhood in Figure 2 to avoid collisions. Cell (0,0) represents the core cell where the pedestrian can move to the adjacent cell either north, south, east or west based on the rules specified above.

Figure : Model Neighbourhood

**M = <Xlist, Ylist, I, X, Y, ƞ, N, {r,c}, C, B, Z, SELECT>**

* Xlist = { Ø };
* Ylist = { Ø };
* I = { Ø };
* X = Y = {0,1,2,3};
  + 0 means unoccupied
  + 1 means up walker
  + 2 means down walker
  + 3 means obstacle
* Ƞ = 17;
* N = {(0,0), (0,1), (0,-1), (1,-1), (1,0), (1,-1), (2,-1), (2,0), (2,-1), (-1,-1), (-1,0), (-1,1), (-2,-1), (-2,0), (-2,1),(0,-2),(0,2)};
* r = 30; c = 30;
* C = {Cij | i ϵ [0,29], j ϵ [0,29]};
* B = {Ø}; % wrapped
* Z = Inverse neighbourhood of N
* SELECT = {(0,0), (1,0), (0,1), (0,-1), (-1,0)};

**Testing**

This section will display the testing results. A video demonstrating the behaviour of the model with the full grid (30x30) can be found here:

<http://www.youtube.com/watch?v=NFByhzFXWMc>

For testing purposes, the grid size was reduced to 8x8 to show specific scenarios and save space on the page. When describing the behaviours of the pedestrians, the directions mentioned are relative to the direction the pedestrian is facing.

**Rule 1 Testing:**

These tests are carried out to verify that pedestrians move forward, provided that there are no obstacles or walkers directly in front of them. Figures 3 and 4 demonstrate different scenarios for this rule.

Figure : Rule 1 Test 1

Figure 3 displays that since there are no walkers or obstacles directly in front of them (within 2 cells ahead), they move to the cell in front of them. Figure 3 also demonstrates the behaviour of the wrapped border (ie. when a down walker (green) reaches the bottom of the grid, they reappear at the top of the grid).

Figure : Rule 1 Test 2

As shown in Figure 4 since there are no two pedestrians going for the same cell in the next time-step, they move forward even though there are obstacles or walkers of the same type two cells ahead. This is as expected and no collisions result.

Over the course of the simulation, the rule functions very well.

**Rule 2 Testing:**

These tests are carried out to verify that when a walker has an obstacle or pedestrian directly in front of them, they move to the right if it is possible. Figures 5 and 6 demonstrate different scenarios for this rule.

Figure : Rule 2 Test 1

As seen in Figure 5, the pedestrians move right when they are blocked by either another pedestrian or obstacle directly ahead of them.

Figure 6 demonstrates a more complex test for this rule. In the scenario on the bottom left of the figure, the orange cell (up walker) has another pedestrian in front of it. However, since the down walker (green) will be going to the right of the orange cell in the next time-step, the orange cell waits because the down walker has the right of way. Another collision is avoided.

Similarly, in the upper scenario on Figure 6, the green (down walker) cell has a pedestrian in front of it, and an opening to its right. However since the up walker has precedence and will take that cell in the next time-step, green waits.

Finally, even though there are two obstacles NE, and SE of the other orange cell, they pose no threat to a collision in the next time step, so the up walker moves right, in between them.

Over the course of the simulation this rule behaves beautifully.

Figure : Rule 2 Test 2

**Rule 3 Testing:**

These tests are carried out to verify that when a walker has an obstacle or pedestrian directly in front and to the right of them, they move to the left if it is possible. Figures 7 and 8 demonstrate different scenarios for this rule.

Figure : Rule 3 Test 1

As seen in Figure 7, the pedestrians that have obstacles and/or pedestrians in front of them, and to their right, move to the left when the left cell is available and not going to be taken by another pedestrian in the next time-step.

Figure 8 shows a more complex scenario for this rule. In the bottom left scenario of the figure it shows both an up walker (orange) and a down walker (green) with obstacles / pedestrians in front of them and to their right. The orange cell does have the left cell available, but it also has to check if there will be another pedestrian going for that cell as well in the next-time step. In this case a down walker will be going for it, so the orange cell will wait since the down walker has priority. The green cell has two obstacles in the cells NE, and SE of it and the left cell proportional to it is available. Since these obstacles pose no threat of collision in the next time step, the down walker is able to move left. The upper scenario behaves the same way except with the roles reversed.

Over the course of the simulation this rule behaves as expected.

Figure : Rule 3 Test 2

**Rule 4 Testing:**

These tests are carried out to verify that pedestrians that are going for the same cell in the next-time step move right if it is possible to avoid collisions. Figures 9 and 10 demonstrate different scenarios for this rule.

Figure : Rule 4 Test 1

Figure 9 shows two different scenarios. The first scenario involves two sets of pedestrians that have empty cells in front of them, but if they both move forward, there will be a collision. In the bottom left of the figure it shows that the cells to the right of the pedestrians are available, so they move right to avoid collision. In the upper portion of the figure the down walker (green) has an obstacle to the right so it waits and stays where it is.

Figure : Rule 4 Test 2

In the lower scenario on Figure 10, the down walker (green) with a potential collision has an available cell to the right with no one fighting for it, so it moves there in the next time step. However, the up walker (orange) has an available cell to its right, but a down walker in its NE cell that will move to its right cell in the next-time step. As observed, to avoid collisions, the orange cell waits, and does not move right.

In the upper scenario on Figure 10, the up walker has obstacles to its NE and SE cell. Since these obstacles pose no threat of collision in the next-time step, the up walker is able to move to the right cell. The down walker has an available right side, but since it knows that its fellow down walker is going to take that cell in the next time step, it waits to avoid collisions.

Over the course of the simulation this rule behaves as expected.

**Rule 5 Testing:**

This test is carried out to verify the default case that when none of the other rules apply, the pedestrian does nothing and waits where they are. This also covers the behaviour for obstacles.

Figure : Rule 5 Test 1

The lower scenario on Figure 11 shows an up walker (orange) who has obstacles / pedestrians to its right, left, and front adjacent cells. Since none of the other rules apply, the up walker remains where it is until to coast is clear. The upper scenario demonstrates two walkers that will collide if they both move forward. In this case they move right if possible. The up walker does just that, but the down walker has an obstacle to its right, so it waits where it is.

For the purposes of this simulation, the obstacles are meant to be permanently configured to the ground and not moved. This rule provided this.

Over the course of the simulation this rule behaves as expected.

**Conclusion**

To conclude, this rule set behaves very well. Multiple simulations were performed, and there were no collisions or traffic jams observed. The pedestrians flowed very naturally. Much knowledge was learned throughout this assignment. Overall this assignment was a success.

**Reference**

[1] F. Weifeng, Y. Lizhong, F. Weicheng. “Simulation of bi-direction pedestrian movement using a cellular automata model”, 2002.