**Assignment 2 – Conceptual Model Description**

**Cellular Automata Modeling of Pedestrian Movement**

Ali Salaheddin – 100704976

The real system to study is pedestrian movement in an open space. The space matrix is considered to be a mesh of squared cells. The pedestrian can move around in 8 directions around their current cell, unlike a bidirectional corridor, for example.

The model will consider a few pedestrians settled around the space matrix at startup, as well as pedestrians entering the space matrix at different time periods.

The pedestrians try to walk from their initial position to a destination cell.

The model should account for traffic control and collision avoidance. If a pedestrian is found confined and can't move to an adjacent cell, the model would try to provide a mechanism to resolve this.

The model is intended to be represented by a Cell-DEVS formalism. It is intended to be atomic.

The following is an illustration of how the model could behave like:

In Figure 1 the sides of a matrix of cells are numbered

1 through 4 clockwise from the top. In the first time

step shown, entity 11 enters the matrix from Side **1** in

cell (0, 1). The destination on Side ***3*** is through cell

**(3,** 2) which is highlighted for illustration. In the

second time step in Figure **1** entity 12 enters from

Side ***3*** through cell ***(3,*** I), also highlighted. Its

destination is through cell (0, ***3).*** The desired

movements based on the local rules for entities 1 1 and

12 are shown with arrows.

**Part 2:**

C = < X, Y, S, N, type, d, τ, d.int, d.ext, lambda, ta >

X = Y = { }

S =

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

Type =

d =

τ =

Discussion:

Simulation\_3\_ver1

Simulation\_3\_ver2

Simulation\_2

Simulation\_1

As per the description of the model in Part 1, the simulation consists of 4 pedestrians trying to reach a destination from their initial standing point. They follow the set of rules defined in the τ function as described above. For further comments on the rules and cell value set, please refer to the pedest\_walk.ma file.

Simulation\_3\_ver1:

In the simulation video Simulation\_3\_ver1.avi, we can see the starting point of each of the pedestrians on the matrix. The each pedestrian is colored and holds a negative number marking its walking direction to the next free cell. Also, the destination cell holds the color of the pedestrian trying to reach it. Destinations are also marked by a positive number (10, 20, 30 or 40) also related to its matching pedestrian.

During the simulation, it is observed how the pedestrians follow simple walking procedures by continuing to walk on a straight path (horizontal or vertical) and in the same direction until they reach the border of the matrix or come across an obstacle (black cell) or another pedestrian in their way. In these exceptions, the pedestrian makes a turn and follows that new path until another exception arises on its way.

If the pedestrian walks onto a cell which is on a straight horizontal or vertical path (including obstacles, if there are any in the way) to its destination, the pedestrian takes this new path (i.e. makes a turn, if the new path cuts the directional path it was walking on) and walks on it. The pedestrian isn’t modeled to be smart enough to correctly facing its destination. Currently, it’s modeled to take an arbitrary horizontal or vertical direction on this path.

As the pedestrian approaches the destination cell, it then goes to it and disappears. In the simulation, there’s an erroneous behavior where the pedestrian jumps over an empty cell separating it from the destination, to reach its goal. It isn’t yet understood why this happens or where the error in the simulation code lies.