Chapter 10
An Introduction to Pedestrian Modeling Using Spatial Discrete-Event Modeling and Simulation

Gabriel A. Wainer and Ala'a Al-Habashna

Abstract We discuss an approach for spatial discrete-event Modeling and Simulation (M&S) of crowds using the Cell-DEVS formalism, which provides some advantages over existing crowd modeling approaches, as it provides a trade-off between the simulated granularity level and the computational requirements. We explain through simple one- and two-dimensional models how Cell-DEVS is used to build pedestrian models. Furthermore, the usability of the approach is verified by employing it in real-life case studies. We discuss different case studies, employing the Cell-DEVS approach to build a model for a general building evacuation, and a fire evacuation model.

10.1 Introduction

Modeling and Simulation (M&S) is an important tool that has been employed to study the behavior of many complex systems. One of the emerging applications for M&S is studying crowd dynamics. The simulation of virtual crowds has found a certain appeal as an illustrative tool in urban and architectural projects. It allows designers and engineers to visualize the utility of their project's space and facilitates the feedback, discussion, and decision-making among all stakeholders. This is becoming increasingly important considering that the world's population is increasing at a high rate, and moving towards urban areas, which increased the occurrence of the crowd phenomena (Zhan et al. 2008).

Crowd analysis can be used for developing crowd management strategies to increase safety in highly crowded situations (like concerts, convocations, demonstrations, public celebrations, etc.). Crowd analysis can also be used in building design, in order to provide a more efficient use of spaces. Crowd analysis is also important in virtual environments as it leads to better simulation in such artificial

G. A. Wainer (✉) · A. Al-Habashna
Department of Systems and Computer Engineering, Carleton University, 1125 Colonel by Dr., Ottawa, ON K12 5B6, Canada
e-mail: gwainer@sce.carleton.ca

J. Sokolowski et al. (eds.), *Summer of Simulation*, Simulation Foundations, Methods and Applications, https://doi.org/10.1007/978-3-030-17164-3_10

settings. Crowd M&S is now used in many areas such as safety, architectural design, computer games, transportation, etc. (Zhou et al. 2010), as in most situations, it can be very difficult (or costly) to study the behavior of crowds using simulacra. On the other hand, crowds exhibit highly complex dynamics, which makes it difficult to characterize its behavior using pure analytical models. When modeling a small crowd (i.e., one with a few dozen individuals), it is easy to investigate the behavior of each individual. However, in large crowds (i.e., with thousands of individuals), the interest is in the overall emergent behavior.

In this chapter, we present an approach for modeling and simulation of crowds based on Cell-DEVS. As Cell-DEVS is an entity-based approach, it provides a higher level of details of the behavior of the individuals in the crowd than fluid dynamic models. At the same time, the Cell-DEVS models require less computational demands than agent-based models.

10.2 Related Work

There are many models have been proposed recently for M&S of crowds. These models can be categorized based on the modeling approach into fluid dynamics-based models, social force-based models, agent-based models, Cellular Automata (CA)-based models, and others. In this section, we will discuss each of these methods.

10.2.1 Fluid Dynamic Models

Fluid dynamics models study the crowd as a continuum using coupled nonlinear, partial differential equations that can be solved for simple geometries. It has been proposed that crowds move in a similar manner as fluid flows. Based on that, early work in this area (Bradley 1993) suggested that the Navier-Stokes equations could be applied to pedestrian flows. However, this does not take into account many factors that affect the crowd behavior such as various physiological, psychological, and social factors. As such, recent models do not use the Navier-Stokes equations in their entirety; rather the concepts of fluid dynamics are combined with consultation from behavioral scientists.

In Hughes (2003), the author extended the work in this area, where the crowd is modeled using classical fluid dynamics, but with the additional assumption that human flows “think”. Bradley’s model was based on a sociological view of crowds which considered that non-orchestrated crowds are rational and can therefore be expected to abide by scientific rules of behavior (McPhail 1991). Hence, the nonlinear, time-dependent, simultaneous equations representing a crowd are conformably mappable. This property makes many interesting applications analytically tractable. The theory has been used to study the Hajj (annual Islamic pilgrimage to

Mecca), in an attempt to improve the flow of pilgrims over the Jamarat Bridge near Mecca. For further and more up to date examples and applications in the literature on Fluid dynamics models, the reader is referred to Treuille et al. (2006), Xia et al. (2009), and Dogbe (2010). The problem with fluid dynamics is that the models do not provide high-resolution details of the behavior of the individuals.

10.2.2 Social-Force Models

Another method, proposed in Helbing and Molnár (1995) is called *social force*. In this method, the human motion is viewed as a complex behavior subject to a self-driving force and repulsive forces from the environment (other pedestrians and obstacles). Each pedestrian is assumed to be affected by four major factors; the destination to be reached, the distance to be kept from other people, the distance to be kept from borders and other obstacles, and other people or objects that might attract the pedestrian. The completed social force model can be found from these factors, and the paths taken by pedestrians can be predicted.

In Helbing et al. (2000) the social force model was extended to exhibit numerous phenomena to study the buildup of pressure observed during escape panics. The model considers a mixture of socio-psychological and physical forces influencing the behavior in the crowd. For further date examples, the reader is referred to Hoogendoorn and Bovy (2004) and Campanella et al. (2009).

10.2.3 Agent-Based Models

In agent-based models, each individual in the crowd is modeled as separate agent that takes its decisions independently. The local phenomena can affect the decision-making process of each individual, while the entire crowd can produce emerging patterns deducted by the social and physical aspects of each individual.

In Klügl and Rindsfüser (2007), the authors presented an agent-based simulation of pedestrian traffic of the complete railway station of Bern during rush hour. In their simulations, more than 40,000 agents pass through the station during 1.5 h. Furthermore, in their simulations, pedestrians are not only capable of avoiding collisions, but also able to flexibly plan their way through the railway station.

In Ronald et al. (2007), the authors investigated the behaviors that pedestrians may exhibit, and the belief-desire-intention (BDI) architecture, presenting the development of a sample model using Prometheus, an agent-oriented design methodology, and JACK, an agent-oriented programming language.

Agent-based models usually try to simulate crowd at fine scale, which makes them computationally demanding and more suitable for short-term simulations with small-sized crowds. Further examples in the literature on agent-based models can be found in Pluchino et al. (2014) and Liu et al. (2014).

10.2.4 Cellular Automata Based Models

CA is one of the oldest models of natural computing; it was introduced by John von Neumann in the late 1940s (Neumann and Burks 1966; Burks 1971; Wolfram 1986). In CA, the studied space is represented as a lattice of cells, with each cell being a state machine. The states of a cell come from a finite set of states (Kari 2005). Cells change their states synchronously at discrete time steps. The state of a cell at the next time step depends on its current state, and the current states of the neighboring cells according to an update rule. The neighborhood usually contains some or all the adjacent cells, but more general neighborhoods can be specified.

CA have been used recently for M&S of pedestrian movement. In Burstedde et al. (2001), the authors proposed a 2D CA model to simulate pedestrian traffic, and to simulate the evacuation of a large room with reduced visibility. The model introduced the concept of "floor field", which can be thought of as a second grid of cells underlying the grid of cells occupied by the pedestrians. Floor field holds the probabilities of moving from a cell to other cells. Dynamic floor cells can evolve with time so that probabilities change with time depending for example on the presence of pedestrians. Hence, floor field is used to model a "long-ranged" attractive interaction between the pedestrians.

The Situated Cellular Automata (SCA) model (Bandini et al. 2006) is a particular class of Multilayered Multi-Agent Situated Systems (MMASS). SCA provides explicit spatial representation, and defines adjacency geometries. SCA was used to build a small-sized model to simulate an environment with the crowd defined as a Multi-Agent Systems (MAS). In Tao and Jun (2009), an entity-based model was used to represent bidirectional pedestrian flow using CA. Different behavioral factors were considered such as position exchange and step back. The pedestrian CA in Ji et al. (2013) focused on acceleration and overtaking. They divided pedestrians into two categories: aggressive and conservative. The model was used to simulate the movement of pedestrians in a corridor.

In Masuda et al. (2014), a simple CA reproduces oscillation phenomena due to formation and destabilization of arches in 2D flows. This is used to study the jamming of pedestrian crowds that occurs due to the formation of arches at bottlenecks. The model predicts critical bottleneck sizes for particle flows without congestion, and it determines the dependency of the jamming probability on the system size. In Vihas et al. (2013), the authors define a CA in which pedestrians follow leaders, as this phenomenon is a fundamental driving mechanism. The model provides microscopic simulation of the crowd, as all configurations of the model are triggered by simple rules applied locally to each of the group members. They also study the emergence of qualitative attributes such as collective effects, random to coherent motion due to a common purpose, and transition to incoordination (arching) due to clogging.

The proxemic approach (Was et al. 2012), which is the process of acquisition of space in evacuation modeling, is based on a detailed representation of space and floor fields. The model allows for efficient, real time simulation of evacuation from large facilities using detailed representation of spatial relations. In order to reduce the computation cost of pedestrian models, (Steffen and Chraibi 2014) reduce the simulation time by using a multicast approach that performs fast simulation of probable evacuation scenarios. The work deals with the problem of passing agents from a CA to a force-based model, and they provide a CA that addresses the problem at less computational cost, with some possible loss of accuracy.

10.2.5 Cell-DEVS

The Cell-DEVS formalism (Wainer 2000; Wainer 2009), allows modeling discrete-event cell spaces built as n-dimensional grids of cells. Each cell is defined as a DEVS atomic model, and a procedure to couple cells is defined. Figure 10.1a shows the contents of an atomic cell. A cell is only active when an external event occurs, or when an internal event is scheduled. When there are no further scheduled events, the cell will passivate. When an external event occurs, the external transition function is executed, and the local computing function (τ) is activated. When the cell's state changes, the external function will schedule an internal transition, and the state change is transmitted after a delay of d. The local computing function in a Cell-DEVS model computes the next state of a cell depending on its current state, and the states of a finite set of nearby cells (like in CA).

The internal computing function is defined using a set of rules indicating the output VALUE for the cell's state after some time DELAY, when a PRECONDITION is satisfied. The rule format is denoted as <VALUE> <DELAY> <PRECONDITION>, which means that when the PRECONDITION is satisfied, the state of the cell will change to the assigned VALUE, and this new value will be transmitted to its neighborhood after a period time of DELAY.

After a cell is defined, it can be integrated into a coupled model representing the cell space. The CD++ M&S tool provides a development environment for implementing Cell-DEVS models using a built-in specification language (Wainer 2009).

Cell-DEVS is built on top of the DEVS formalism (Zeigler et al. 2000), which provides a formal framework for modeling generic dynamic systems and includes

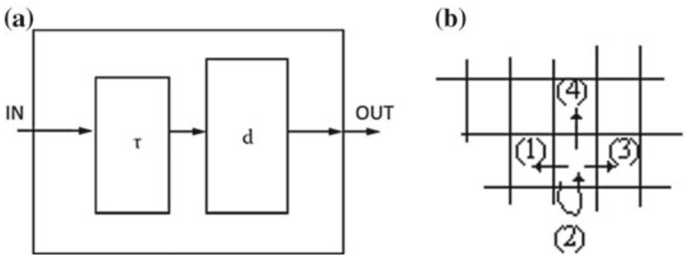

Fig. 10.1 *Cell-DEVS* model: **a** Atomic cell **b** 2D coupled model

hierarchical, modular, and component-oriented structure, and formal specifications for defining structure and behavior of a discrete event model. Coupled component define the hierarchical structure of the system, while each atomic component is the basic building block of the system, which represents its behavior.

Since Cell-DEVS implement entity-based modeling, it provides a higher level of details than fluid dynamics models, and needs less computation demands than agent-based models (Al-Habashna and Wainer 2015; Zhou et al. 2010). Furthermore, as discussed in Sect. 10.1, Cell-DEVS has multiple advantages over CA, which makes it easier to develop larger models, and allows faster execution of models. In the following section, we discuss how Cell-DEVS can be employed to build pedestrian and crowd models, and present various examples.

10.2.6 Centroidal Particle Dynamics

The Centroidal Particle Dynamics (CPD) method is a personal space preserving method, which models pedestrian dynamics based on the concept that a pedestrian, in crowd, tend to maintain its personal space (Hesham et al. 2018). When this concept is employed for modeling pedestrians in highly-dense areas, it produces a very realistic pedestrian behavior. CPD provides a dynamic, autonomous, and adaptive approach that generates realistic pedestrian behavior.

The CPD approach is a variation of the social force approach. With this approach, a pedestrian is modeled as an entity with a personal space, and each pedestrian in the simulation tries to maintain and regain its personal space. This approach very suitable for close-range crowd in areas with high pedestrian density, as the personal space preservation concept produces realistic crowd behavior. The personal space preservation concept was not employed arbitrarily; it is a well-known human behavior in physiology, and a natural reaction that people employ as a mechanism to avoid any uncertainty and unexpected behavior that could be taken by surrounding people (especially strangers).

With the CPD approach, the Personal Space Map (PSM) is first constructed for the pedestrians in the model. This is done by checking the surrounding of each pedestrian within a certain radius (0.8–1.0 m) and calculating the available and violated space for each pedestrian. According to the results obtained in the previous step, the new location for the geometric center (centroid) of a pedestrian with be calculated to regain the full range of personal space. The centroidal force is also calculated, which is a vector pointing from the current location of the pedestrian to the new location.

The movement of a pedestrian can be impacted by various forces such as the movement on a global path to a destination, physical factors such as obstacles, and the different psychological and physiologic factors such as fear or following friends and family. When the centroidal force is calculated, it will be used with the other forces affecting the pedestrian movement to calculate the net force. The net force is then used to calculate the acceleration/declaration experienced by the pedestrian at each time step.

10.3 Models of Pedestrian Behavior with Cell-DEVS

In this section, we discuss how Cell-DEVS can be used to develop pedestrian and crowd models. Then, we provide two case studies of 1D and 2D pedestrian models. The models presented in this section are basic models that provide examples and explanation of how Cell-DEVS are used for modeling and simulation of crowd. More complex and realistic models will be provided in the next section.

As discussed in Sect. 10.2, in order to build a Cell-DEVS model we first need to define the following components of the model:

- The dimensions of the model, i.e., the number of cells in each dimension
- The set of possible states for the cells
- The neighborhood shape
- The set of rules that define the behavior of the cells

A Cell-DEVS model can be built in a modular and hierarchical fashion, by integrating various DEVS and Cell-DEVS models together. The cell's states can be used to represent different aspects of the space used by that cell. For instance, certain values can be representing occupied cells while others can represent vacant cells. Depending on the space of each cell, a cell can contain more than one pedestrian. For small cells (e.g., 0.4 m^2), a cell can only occupy one pedestrian at a time. In such cases, mechanisms to model collision avoidance should be employed. Moving direction is another important aspect that should be taken into consideration when defining cell states. Different values can be assigned to different movement directions. Pedestrians may have different speeds due to different time constrains. Pedestrians might also walk at a constant speed or have to change their speed. As such, the status of cell can be used to reflect the speed of the pedestrian in the cell.

10.3.1 One-Dimensional Movement Model

We will start with a simple one-dimensional movement model. A cell can be considered to be a single-pedestrian space. This is the case considered in all the Cell-DEVS models in this chapter. Figure 10.2 shows the neighborhood used in

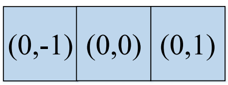

Fig. 10.2 Neighborhood of the 1-D movement model

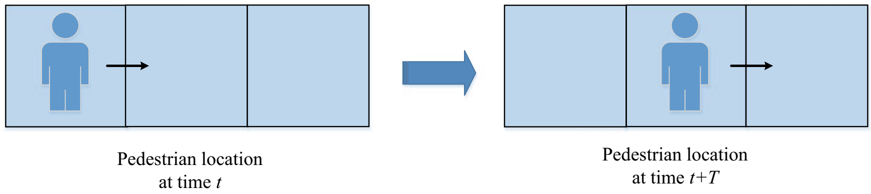

Fig. 10.3 Pedestrian moving forward in one-direction

such a model. The figure shows a simple, 1×3 neighborhood. There are two states for each cell; a cell can be either empty (state 0) or occupied (state 1).

A simple example is shown in Fig. 10.3; which depicts a pedestrian moving east. A pedestrian moves one cell forward at a time. The movement speed of the pedestrian in the model can be controlled through the time delay.

A simple set of rules can be used for this model. For instance, the pedestrian moves forward if the cell ahead is vacant, which can be represented as follows.

```
Rule 1: 1 400 {(0,0)=0 and (0,-1)=1}
Rule 2: 0 400 {(0,0)=1 and (0,1)=0}
```

The rules are checked sequentially starting from the first rule and continuing until one of the preconditions is satisfied. The first rule states that if the core cell is empty and there is a pedestrian to the west (cell (0,−1)), the next state of the core cell will be 1. The second rule states that if there is a pedestrian in the core cell, and the cell to the east (cell (0,1)) is vacant, the next state of the core cell will be 0. The combination of these two rules represents the movement of the pedestrian. Although we have used a fixed delay of 400 ms to represent the time to cross the 0.4 m distance, we can use a function representing the speed for each cell in each of the rules allowing the modeler to easily represent the time it takes for each individual to cross a cell in different directions. As the cell area is 0.4 m^2, which is the average area occupied by a pedestrian, the delay is set to 400 ms. In this way, a constant movement will simulate a walking speed of 1 m/s, which is the average walking speed of pedestrians.

10.3.2 Two-Dimensional Movement Model (Crosswalk Model)

In this section, we explain through an example how to use Cell-DEVS for building a 2-dimensional pedestrian model. The example used is a pedestrian crosswalk model that represents the dynamics of pedestrians crossing a street. Three different scenarios might occur:

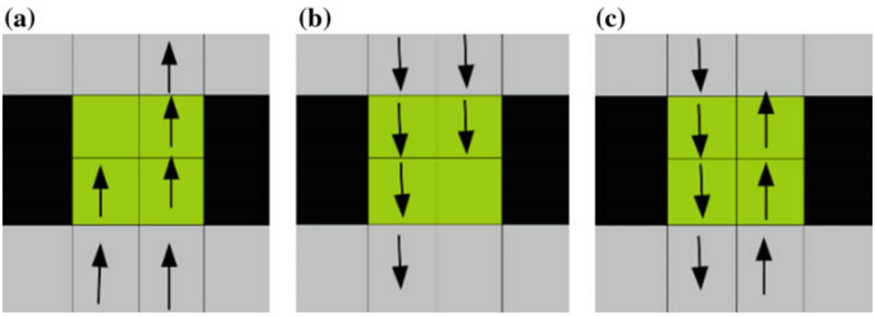

Fig. 10.4 Crosswalk model scenarios. **a** Upward movement **b** Downward movement **c** Bidirectional movement

1. There are pedestrians standing only at one end of the crosswalk, i.e. only upward movement. This scenario is shown in Fig. 10.4a (arrows represent pedestrians).
2. There are pedestrians standing only at the other end of the crosswalk, i.e. only downward movement. This scenario is shown in Fig. 10.4b.
3. There are pedestrians at both ends of the crosswalk, therefore, during their passing, collisions might occur. This scenario is shown in Fig. 10.4c.

The model represents the movement of pedestrians along the crosswalk, and illustrates collisions that may happen when two pedestrians (one moving upward and another moving downward) are vying for the same cell. In this case one, of the pedestrians must sidestep. If sidestepping cannot be done due to unavailability of cells, then pedestrians must wait until a surrounding cell becomes available.

This model uses two columns of cells to represent the crosswalk. Pedestrians will wait at both ends of the crosswalk until the corresponding cells (i.e. crosswalk) turns green. The crosswalk is only safe for a certain amount of time (e.g., 1 min) and once those cells have changed back to black (representing a street with vehicles) no pedestrian is allowed to cross. The pedestrians are allowed to move three cells ahead (2 green cells representing the crosswalk and 1 gray cell representing the side walk) or one cell sideway (sidestepping).

The Cell-DEVS model will be based on a two-dimensional square grid. The grid dimensions are 4×4. Figure 10.5 shows the cell space. The central 4 cells model the crosswalk. When the traffic light is green, and cars pass the crosswalk, the 4 cells have value of zero. The color of the central cells in this case will be red and pedestrians are not allowed to cross the street. Once the traffic light turns red, the color of these cells will change to green and their corresponding value is set to 2.

The gray cells represent the sidewalks at both sides of the street, and they are occupied by pedestrians. Once the crosswalk cells turn green, the pedestrians will pass the crosswalk and get to the other side of the street. All pedestrians can be standing at one end of the crosswalk and move to the other end, or there could be

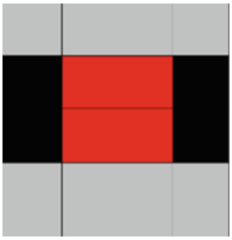

Fig. 10.5 Cell space representing the crosswalk scene. Red cells represent the crosswalk

bi-directional movement. We use a Moore's neighborhood (the 9 adjacent neighbors) and the cell states in Table 10.1.

More rules are needed in the case of two-dimensional movement to handle collision avoidance. For instance, when two pedestrians walk in opposite directions, the rules governing their movement should consider this, and avoid collisions. We use pedestrians walking at a constant speed.

Following, we define the rules that govern the movement of a pedestrian walking upward. The rules of four different scenarios are listed and explained below. The first rule in each case considers the case where the pedestrian leaves the current cell, while the second rule considers the cell where the pedestrian moves. We show the rules for the first 3 cases. Afterwards, we list rules for the remaining cases.

(a) A crosswalk cell is red, turn it to green after some time delay

```
Rule 1 : 2 400 { (0,0)= 0 }
```

Here, if a crosswalk cell is red (has a value of 0), it will turn green (value 2), and after 400 ms, this change is reported to the neighboring cells, which will make the cell be considered as safe to walk into.

(b) The next rules show the case where there is no pedestrian ahead and the cell is green, then we move upward. This is a case where there is no other pedestrian or an obstacle in the cell ahead (−1,0), and there is no pedestrian in cell (−2,0) walking in the opposite direction. This case is illustrated in Fig. 10.4a.

```
Rule 2: 2 400 {(0,0)=1 and (-1,0)=2 and (-2,0)=2}
Rule 3: 1 400 {(0,0)=0 and (1,0)=1 and (1,0)=2}
```

The first rule checks if the current cell has a pedestrian walking upward. In this case, the pedestrian will move forward, i.e., the current cell will be green. The

Table 10.1 State values used for the crosswalk model

State	Value	Color
Crosswalk is not safe to pass	0	Red
Crosswalk is safe to pass	2	Green
Cell is occupied by a pedestrian moving upward	1	Blue
Cell is occupied by a pedestrian moving downward	−1	Blue
Street	3	Black
Sidewalk	5	Gray

second rule checks if the core cell is empty and the cell below has a pedestrian walking upward, in which case the pedestrian will move to the core cell.

(c) If there is a pedestrian ahead, move to the right (east) provided that this cell to the right is green. If the next cell forward contains a pedestrian, we try to move to the east if that cell is green and no other pedestrian is going to move into it.

```
Rule 4: 2 400 {(0,0)=1 and (-1,0)!=2 and (0,-1)=2 and
(-1,-1)!=-1 and (1,-1)!=1}
Rule 5: 1 400 {(0,0)=2 and (0,1)=1 and (-1,1)!=2 and
(-1,0)!=-1 and (1,0)!=1}
```

Rule 4 checks if the current cell has a pedestrian walking upward, and the cell forwarded is occupied (or not green), and the cell to the east is available, green, and no one is moving to it (from the bottom or the top). In such case, the current cell will be green, cause the pedestrian will move out of it. Rule 5 checks the same conditions for the cell where the pedestrian is going to move into.

The remaining rules are listed below.

(d) *Pedestrian/obstacle ahead and to the east, move to the left (west) if possible*. In this case, the cell ahead is occupied by an obstacle or a pedestrian moving in the opposite direction, and the cell to the east is occupied by a pedestrian or an obstacle. In such case, the pedestrian will have to move to the cell to the west, given that, it is empty, and that no pedestrian is walking into it

(e) *Two pedestrians vying for the same cell; move to the east if possible.* When there is pedestrian moving upward and another moving downward, and both are moving into the same cell, the one walking upward will move to the cell to the east, given that it is vacant, green, and no other pedestrian is moving into it.

(f) *Two pedestrians vying for the same cell and there is a pedestrian or an obstacle to the east; move to the west.* When there is a pedestrian moving upward and another moving downward, and both are moving into the same cell, and it is not possible for the one moving upward to sidestep to the east, the pedestrian will sidestep to the west, given that it is vacant and no other pedestrian is moving into it.

When none of the cases above is satisfied, the pedestrian simply does not move.

Following, we present simulation results obtained for this model. First, we simulated the case were two pedestrians walking upward. Step by step visualization of the obtained results are shown in Fig. 10.6. Initially, the pedestrians are the bottom of the crosswalk, and the crosswalk is red, which means that the pedestrians cannot cross the street. In the next step, the crosswalk cells turn into green (rule 1) which means that pedestrians can cross the street now. In the next 3 steps the pedestrians move forward, one cell at a time until they cross the street and reach the sidewalk on the other side.

The following scenario shows two pedestrians moving across the street. One of them is moving upward, and the other is moving downward. Each pedestrian was positioned in a different lane so that no collision will be experienced. The results from this case are shown in Fig. 10.7. As in the previous scenario, the crosswalk cells are red in the beginning, and they turn green afterwards. Thereafter, the pedestrians start walking cross the street, each in the opposite direction of the other.

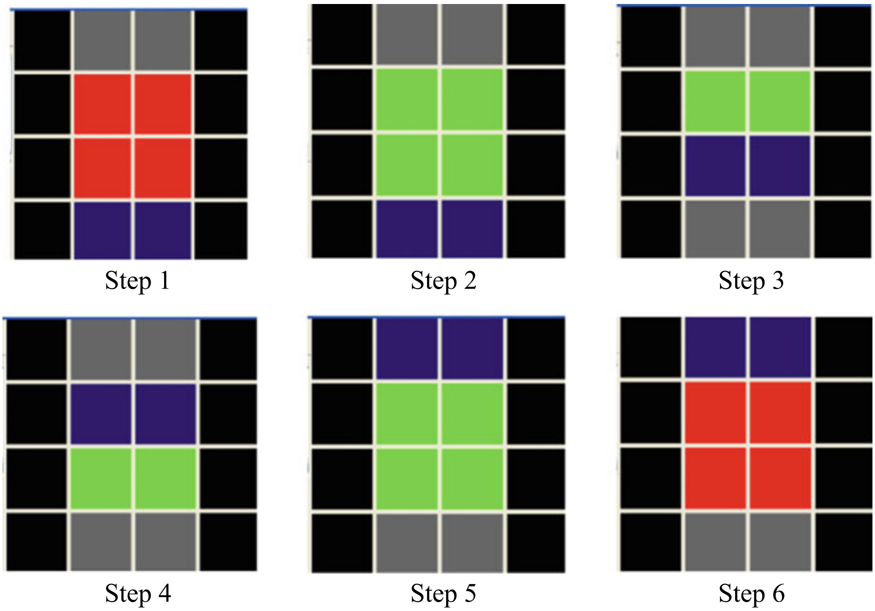

Fig. 10.6 Simulation results: two pedestrians moving upward

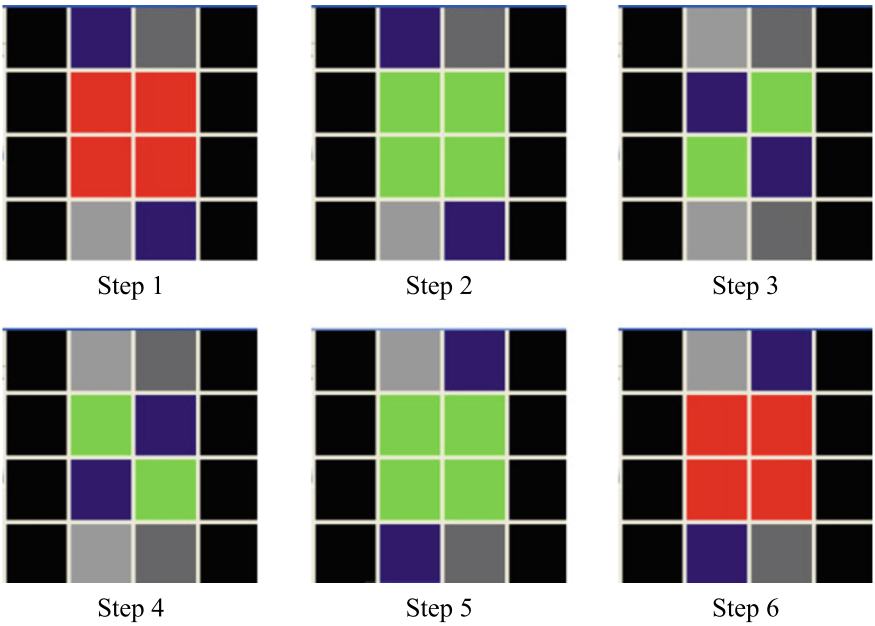

Fig. 10.7 Simulation results: two pedestrians moving upward (no collision)

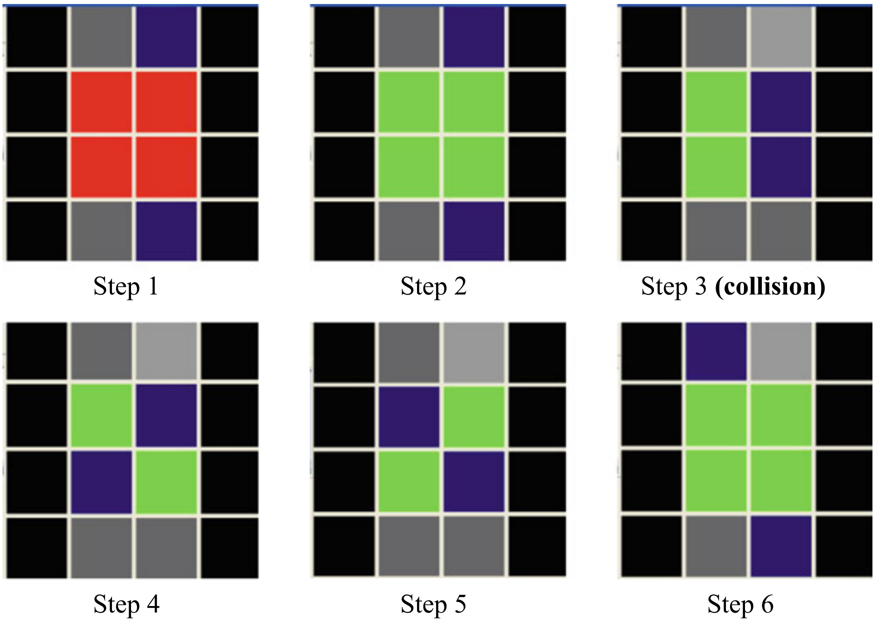

Fig. 10.8 Simulation results: two pedestrians moving upward (with collision)

Since, each one is walking in a different lane; both reach the sidewalk on the other side of the street without colliding with the other.

In the fourth scenario (shown in Fig. 10.8), we simulated two pedestrians moving across the street. One of them moving upward and one is moving downward. Not like scenario 3, both pedestrians are walking in the same lane, and hence, they will collide at step 3. In step 4, the pedestrian walking upward steps to the west (as the cell to the east is unavailable). This resolves the collision, as the pedestrians are now walking in different lanes. Each pedestrian continue moving ahead until reaching the sidewalk on the other side of the street.

10.4 Cell-DEVS Fire-Evacuation Model

Shopping malls, office and school buildings are some examples of buildings that we come across in a day to day basis which include meeting of a large number of people within closed areas. These buildings are made in such ways to maximize the utilization of the limited space. However, safety should be taken into consideration when designing them, especially for emergency evacuation scenarios. One of the most common causes for such evacuations is the occurrence of fires. In such scenarios, a large number of people will have to evacuate the building in a limited time.

In this section, we present a more advanced pedestrian model using Cell-DEVS: a 3-dimensional fire-evacuation model. In the first subsection, we present our evacuation model. We also show some simulation results for the building evacuation model. In the second subsection, we discuss the fire spread model. Thereafter, present fire-evacuation scenarios and simulation results.

10.4.1 Building Evacuation

In our model, the state of each cell will be determined by two main factors. The first factor is the direction of the shortest path to the nearest exit, as illustrated in Fig. 10.9. The figure shows that each cell will have a certain direction the pedestrian (occupying the cell) should take to get to the exit. As we are working with a 3-dimensional model here, some cells will be used as stair-cases that lead from one floor to another.

The second factor that decides the cell sate is whether it is occupied or not. In the beginning of each simulation, the direction of each cell is determined. Then, pedestrians are distributed randomly throughout the building. Depending on the two factors above (direction and occupancy), the state of the cell will be set. In addition to the cells that model the rooms and pathways, other cells are used to represent walls and exits in the building. The cell states are shown in Table 10.2.

As can be seen in Table 10.2, numbers from 3 to 10 represent the pathways. Even numbers in that range represent occupied cells while odd numbers represent vacant cells. For example, a cell with a value of 10 means that the cell is occupied, and the person will move to the cell to the left when it is available. When the person moves, the state of the cell will change to 9. A three-dimensional neighborhood is also used here as pedestrian need to check the availability of the stairwell cells when moving up and down between building floors (in the third dimension).

When pedestrians enter a staircase cell, they "vanish" from the simulation, and they are considered as having left the building. Stairway cells are used to model stairways between the floors. A stairway is modeled with two cells, one to represent the top of the stairway, and another to represent the bottom. In the event that both the top and base of the stairway are occupied, no one can enter until either is vacant.

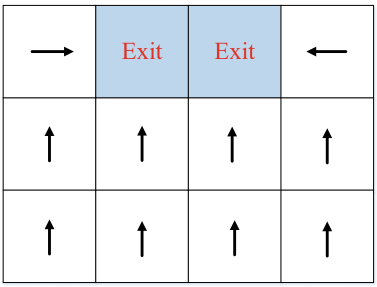

Fig. 10.9 Movement direction of the cells

Table 10.2 Cell states of the fire evacuation model

State	State color	State name	State	State color	State name
1	Black	Wall	8	Green	Up occupied
2	Red	Exit	9	White	Left
3	White	Down	10	Green	Left occupied
4	Green	Down occupied	11	Yellow	Top of stairs
5	White	Right	12	Green	Top occupied
6	Green	Right occupied	13	Blue	Bottom of stairs
7	White	Up	14	Green	Bottom occupied
15	Red	Fire		Green	

A pedestrian enters the top of stairwell when the pathway instructs it to and when the top of the stairway has the value of 11 (unoccupied). A pedestrian at the top of the stairway moves down to the bottom of the stairway whenever it is vacant. Afterwards, the pedestrians leave the bottom of the stairway and follow the path to the next stairway or to the exit of the building.

The rules used in the model can be categorized into three sets: initialization, movement, and fire-spread rules. As their name indicates, initialization rules are used to distribute pedestrians in the model, and set the initial states of the pathway cells. A sample of the initialization rules, which are used to initialize the first floor, are shown below:

```
rule : { if (uniform(0,1) < 0.2), 10, 9) } 0 { (0,0,0) = 0 and (0,-1,0) > 1
      and (0,-1,0) < 11}
rule: { if (uniform(0,1) < 0.2), 4, 3) } 0 { (0,0,0) = 0 and (1,0,0) > 1 and
      (1,0,0) < 11}
rule : { if (uniform(0,1) < 0.2), 6, 5) }0 { (0,0,0) = 0 and (0,1,0) > 1 and
      (0,1,0) < 11}
rule : { if (uniform(0,1) < 0.2), 8, 7) } 0 { (0,0,0) = 0 and (-1,0,0) > 1
      and (-1,0,0) < 11}
```

All the pathway cells first have a value of zero. The cells next to the exit will first be initialized to point to the exit. Then, the neighboring cells of the ones that were just initialized will be also initialized to point to the former ones, and so on, until all the pathway cells are initialized. In this way, the direction of the cells will always form the shortest path to the exit. In addition to initializing a cell with direction, each cell will be populated with a pedestrian with a certain probability. The probability determine the density of pedestrians in the building. For instance, the rules above generate pedestrians with a probability of 0.2.

The second set of rules are used to control the movement of pedestrians in the various area of the model, including the pathways, stairwells, and exits. Following, we show as an example a set of rules used to move people into an empty cell within the same floor:

```
rule : 4 400 { (0,0,0) = 3 and ( (0,1,0) = 10 or (-1,0,0) = 4 or
      (0,-1,0) = 6 or (-1,0,0) = 14 or (1,0,0) = 14 or (0,1,0) = 14
      or (0,-1,0) = 14 )}
rule : 6 400 { (0,0,0) = 5 and ( (1,0,0) = 8 or (-1,0,0) = 4 or
      (0,-1,0) = 6 or (-1,0,0) = 14 or (1,0,0) = 14  or (0,1,0) = 14
      or (0,-1,0) = 14 )}
rule : 8 400 { (0,0,0) = 7 and ( (1,0,0) = 8 or (0,1,0) = 10 or
      (0,-1,0) = 6 or (-1,0,0) = 14 or (1,0,0) = 14  or (0,1,0) = 14
      or (0,-1,0) = 14 )}
rule : 10 400 { (0,0,0) = 9 and ( (1,0,0) = 8 or (0,1,0) = 10 or
      (-1,0,0) = 4 or (-1,0,0) = 14 or (1,0,0) = 14  or (0,1,0) = 14
      or (0,-1,0) = 14 )}
```

The first rule states that if the core cell value is 3 (vacant cell with direction of movement pointing down), and any of the cells around it is occupied with direction of movement toward the core cell, the next state of the core cell will be 4 (occupied cell with down direction of movement). The other 3 rules are the same but for cells with different direction of movement, for example, the second rule assumes the core cell state is 5. As another example, we list another set of rules below that are used to move people into an exit or a stairwell:

```
rule : 3 400 { (0,0,0) = 4 and ((1,0,0) = 2 or (1,0,0) = 11) }
rule : 9 400 { (0,0,0) = 10 and ((0,-1,0) = 2 or (0,-1,0) = 11) }
rule : 7 400 { (0,0,0) = 8 and ((-1,0,0) = 2 or (-1,0,0) = 11) }
rule : 5 400 { (0,0,0) = 6 and ((0,1,0) = 2 or (0,1,0) = 11) }
```

The first rule, for example, states that if the current value of the cell is 4, and the cell below (in the same level) is either 2 (an exit) or 11 (empty top of stairs), the next value of the core cell will be 3.

In the following, we show some simulation results for building evacuation with the rules above (without fire). Fire-evacuation is discussed in the next subsection.

We consider a 3-dimensional building model of the dimensions (30, 35, 4). We consider different building designs, and evaluate the evacuation time of each design.

10.4.1.1 Model with One Exit

In first design (shown in Fig. 10.10), there is only one exit/stairwell on every floor on the right-side of the building.

Figure 10.10 shows the 4 floors of the building, with the top floor on right, and the bottom level at the left. The figure shows the building at time $t = 0$, i.e., at the time the building is just initialized with pedestrians. The building was populated with pedestrians with 20% density, i.e., each empty cell is occupied with a probability of 0.2. Figure 10.11 shows a screenshot from the simulation at time $t = 1.14$ min. As

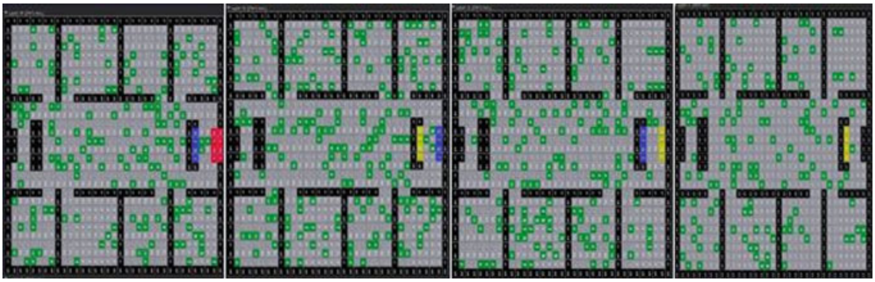

Fig. 10.10 The model of the first 4-floor building at time t = 0

can be seen, pedestrians at each floor are moving towards the entrances of the stairwells to get to the bottom floor and eventually exit the building. With this design, It takes 5:59 m for all the pedestrians to evacuate the building.

10.4.1.2 Model with Two Exits

To improve the above design and reduce the evacuation time, another stairwell has been added to the opposite side of the upper levels, and an exit has been added to the other side of the first level (Fig. 10.12).

From the simulation results we can see that people are evacuating from two exits available on both sides of the building. This should reduce congestion at the stairwells/exits, and hence, reduce evacuation time. With this design, building is evacuated at time t = 3:18 min, which is almost half of the time needed to evacuate the first design. A screenshot of the evacuation process in this design at time, $t = 0.5$ minute, is shown in Fig. 10.13.

10.4.1.3 Model with Obstacles

In previous building designs, there were no structures in each floor. Realistically, buildings usually have structures such as columns. Such structures sometimes

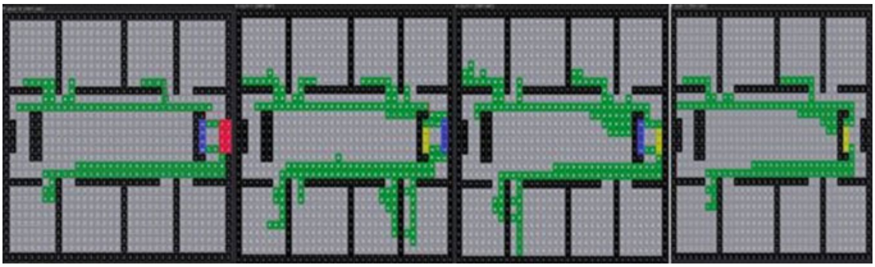

Fig. 10.11 The model of the first 4-floor building at time t = 1.14 min

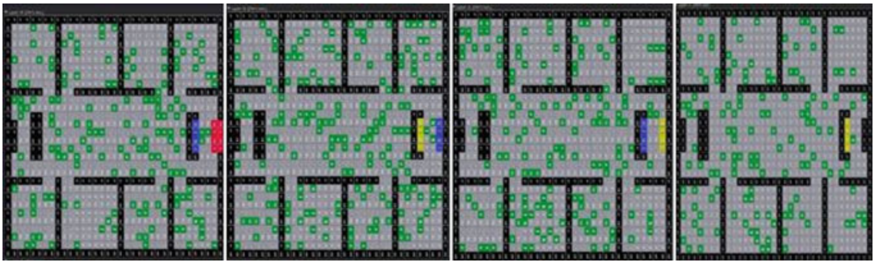

Fig. 10.12 The model of the second 4-floor building at time t = 0 min

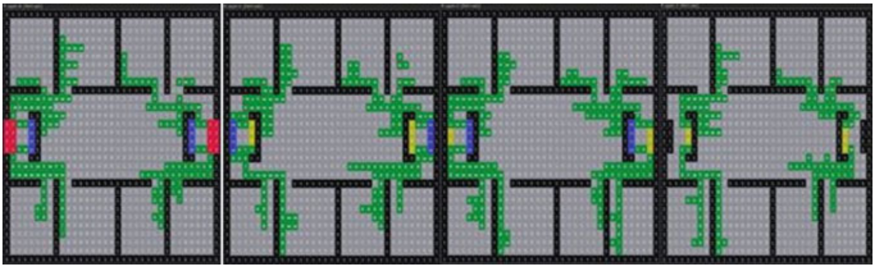

Fig. 10.13 The model of the second 4-floor building at time t = 0.5 min

present obstructions to pedestrians while evacuating the building. To study the effect of such structures, we added obstacles in the previous building (with two exits).

Figure 10.14 shows the initial state of the simulation. Simulations have shown that when evacuating the rooms of each floor, pedestrians were able to maneuver and avoid the obstacles in the rooms. Hence, such obstacles did not have much impact on the evacuation process. However, the obstacles at the stairwells and exits did have an impact on the evacuation process. These obstacles blocked parts of the exits and stairwell entrances and increases congestion at these locations. Results

Fig. 10.14 The model of the third 4-floor building at time t = 0 min

show that evacuation of this building took 4:50 m, which is a considerable increase in the evacuation time over the last design (without obstacles).

10.4.2 Fire-Evacuation Model

In this model, the initial values of the cells is set so that certain cells can be designated as the origin of fire. Furthermore, a set of rules are defined to govern the spread of fire through the building. The fire spread in all directions and the speed of the fire spread is controlled by time delay of the fire cells. Rules are defined such that everything coming in contact of fire will turn into fire in the simulation except cells with a value of 1, i.e., wall cells. The rules for fire spread are as follows,

```
rule : 15 5000{ (0,0,0) = 3 and ((-1,-1,0)=15 or (-1,0,0)=15 or
      (-1,1,0)=15  or  (0,-1,0)=15  or  (0,1,0)=15  or  (1,-1,0)=15  or
      (1,0,0)=15 or (1,1,0)=15)}
rule : 15 5000{ (0,0,0) = 4 and ((-1,-1,0)=15 or (-1,0,0)=15 or
      (-1,1,0)=15  or  (0,-1,0)=15  or  (0,1,0)=15  or  (1,-1,0)=15  or
      (1,0,0)=15 or (1,1,0)=15)}
...
rule : 15 5000{ (0,0,0) = 13 and ((-1,-1,0)=15 or (-1,0,0)=15 or
      (-1,1,0)=15  or  (0,-1,0)=15  or  (0,1,0)=15  or  (1,-1,0)=15  or
      (1,0,0)=15 or (1,1,0)=15)}
rule : 15 5000{ (0,0,0) = 14 and ((-1,-1,0)=15 or (-1,0,0)=15 or
      (-1,1,0)=15  or  (0,-1,0)=15  or  (0,1,0)=15  or  (1,-1,0)=15  or
      (1,0,0)=15 or (1,1,0)=15)}
```

Here, we adopt the building model with two exits and obstructions. Initially, a fire begins on the top floor in one room, and it spreads across the floor. It destroys everything including people and stairs. Fire starts form the red cell at the room at the bottom-right corner of the top floor, as we can see in Fig. 10.15. Fire spreads across the top floor as per the rules above, until it takes over the whole floor, as depicted in Fig. 10.16. People follow the evacuation rules, and eventually flee the building. Nevertheless, when a person is surrounded by fire, that person dies and gets eliminated from the simulation.

With this model, it takes 2:57 m to evacuate the building, which is less than the evacuation time of model without fire. However, not everyone makes it out of the building safely, which explain the reduction in evacuation time, as there were fewer people leaving the building. The speed of fire spread was 1/5 of the speed of pedestrians in the simulations above. Obviously, the speed of the fire spread has an impact on the number of causalities.

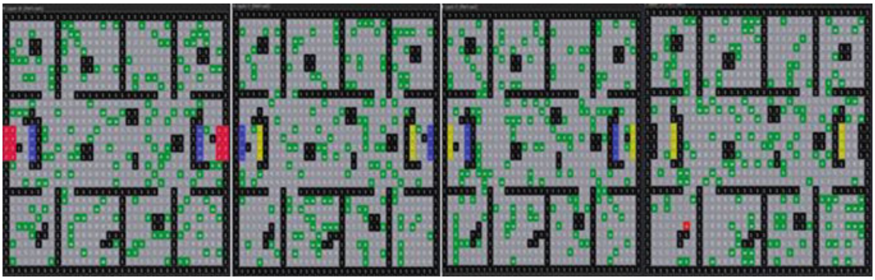

Fig. 10.15 The model of the third 4-floor building at time t = 0.5 min

We ran simulations for another scenario where the fire starts at the third floor instead. In this case, the fire spreads across the third floor and reaches and destroys the stairs from the fourth floor. As such, pedestrians on the fourth floor will not be able to evacuate. Simulations of this scenarios are shown in Figs. 10.17, 10.18, and 10.19. Only the third and fourth floors are shown in these figures.

Figure 10.17 shows the beginning of the simulations when the fire starts at the bottom-left corner of the third floor. Figure 10.19 shows the simulations at t = 32 s. Once can see that the fire took over the whole room and spread outside the room. At t = 3.05 min, one can see that the fire took over the whole third floor and destroyed both stairwells.

One can see from the simulations above, that after the fire spread across the whole third floor and destroyed the stairwells, many people are trapped at the fourth floor and are not able to evacuate the building. Similarly, if the fire starts at the second floor, eventually people at both the third and fourth floors will be trapped inside the building and will not be able to evacuate.

This model has many parameters such as time delays which determine the speed of crowd evacuation, fire spread, etc. These parameters can be calibrated to simulate real-life scenarios and build real-life applications. The model can be used to evaluate different building designs in terms of evacuation speed and causalities in fire evacuation under different scenarios (people density, fire-spread speed, etc.).

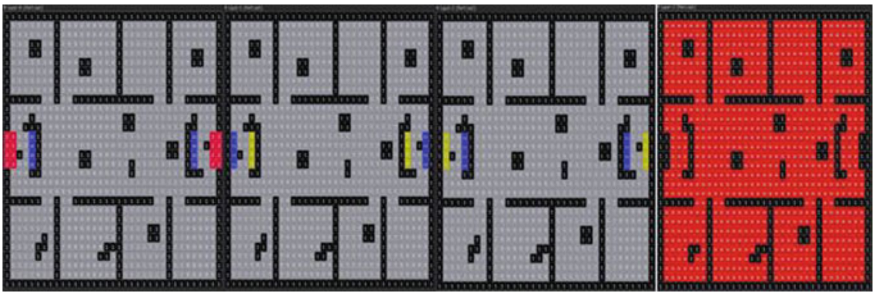

Fig. 10.16 The model of the third 4-floor building at the end of the simulations

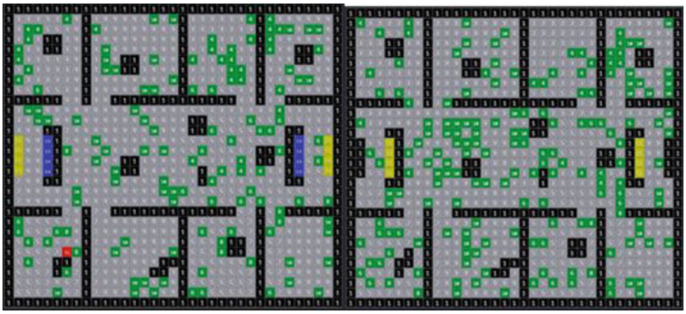

Fig. 10.17 The model of the third-floor fire at time t = 0.0 min

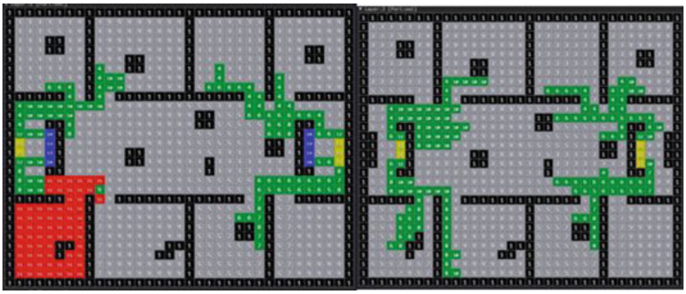

Fig. 10.18 The model of the third-floor fire at time t = 32 s

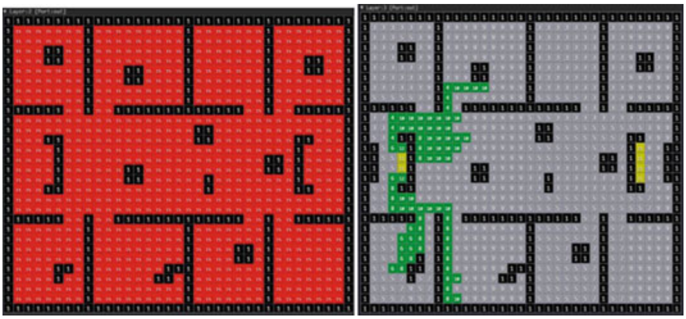

Fig. 10.19 The model of the third-floor fire at time t = 3.05 min

10.5 Conclusion

We provided a review of the existing methods for pedestrian modeling in the literature. Furthermore, an approach for spatial discrete-event Modeling and Simulation (M&S) of crowds using the Cell-DEVS formalisms is presented and discussed in detail. Since Cell-DEVS implement entity-based modeling, it provides a higher level of details than fluid dynamics models, and needs less computation demands than agent-based models. Furthermore, Cell-DEVS has multiple advantages over CA-based methods that makes it easier to develop larger models.

We explain through simple one- and two-dimensional models how Cell-DEVS is used to build pedestrian models. Furthermore, the usability of the approach is verified by employing it in real-life case studies. We discussed different case studies, employing the Cell-DEVS approach to build a model for a general building evacuation, and a fire evacuation model.

Acknowledgements This research has been partially funded by NSERC. The authors thank S. Wang, R. Farrell, M. Moallemi, W. Xiang, M. Van Schyndel, S. Jafer, H. Sohanpal and R. Chunara who built some of the models presented here.

References

Al-Habashna, A. A., & Wainer, G. (2015). Modeling pedestrian behavior with Cell-DEVS: Theory and applications. *Simulation: Transactions of the Society for Modeling and Simulation International, 92,* 117–139.

Bandini, S., Manzoni, S., & Vizzari, G. (2006). Crowd modeling and simulation: The role of multi-agent simulation in design support systems. In *Innovations in design and decision support systems in architecture and urban planning* (pp. 105–120).

Bradley, G E. (1993). A proposed mathematical model for computer prediction of crowd movements and their associated risks. In *Proceedings of the International Conference on Engineering for Crowd Safety.* London, UK.

Burks, W. A. (1971). *Essays on cellular automata*. Urbana: University of Illinois Press.

Burstedde, C., Klauck, K., Schadschneider, A., & Zittartz, J. (2001). Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications, 295,* 507–525.

Campanella, C, Hoogendoorn, S. P., & Daamen, W. (2009). Improving the nomad microscopic walker model. In *12th IFAC Symposium on Control in Transportation Systems.* Fairfield, NJ.

Dogbe, C. (2010). On the cauchy problem for macroscopic model of pedestrian flows. *Journal of Mathematical Analysis and Applications, 372,* 77–85.

Helbing, D., & Molnár, P. (1995). Social force model for pedestrian dynamics. *Physical Review E, 51,* 4282–4286.

Helbing, D., Farkas, I., & Vicsek, T. (2000). Simulating dynamical features of escape panic. *Letters to Nature.*

Hesham, O., Aburime, W., Rabeh, Z., Bhushan, S., & Wainer, G. (2018). Observed behaviours in simulated close-range pedestrian dynamics. In *Proceedings of the Symposium on Simulation for Architecture & Urban Design (SimAUD).* Delft, Netherlands.

Hoogendoorn, S., & Bovy, P. (2004). Pedestrian route-choice and activity scheduling theory and models. *Transportation Research Part B: Methodological, 38,* 169–190.

Hughes, R. L. (2003). The flow of human crowds. *Annual Review on Fluid Mechanics, 35,* 169–182.

Ji, X., Zhou, X., & Ran, B. (2013). A cell-based study on pedestrian acceleration and overtaking in a transfer station corridor. *Physica A: Statistical Mechanics and its Applications, 392,* 1828–1839.

Kari, J. (2005). Theory of cellular automata: A survey. *Theoretical Computer Science, 334,* 3–33.

Klügl, F., & Rindsfüser, G. (2007). Large-scale agent-based pedestrian simulation. *Multiagent System Technologies, 4687,* 145–156.

Liu, S., Lo, S., Ma, J., & Wang, W. (2014). An agent-based microscopic pedestrian flow simulation model for pedestrian traffic problems. *IEEE Transactions on Intelligent Transportation Systems, 15,* 992–1001.

Masuda, T., Nishinari, K., & Schadschneider, A. (2014). Cellular automaton approach to arching in two-dimensional granular media. *Lecture Notes in Computer Science, 8751,* 310–319.

McPhail, C. (1991). *The myth of the madding crowd*. New York: A. de Gruyter.

Pluchino, A., Garofalo, C., Inturri, G., Rapisarda, A., & Ignaccolo, M. (2014). Agent-based simulation of pedestrian behaviour in closed spaces: A museum case study. *Journal of Artificial Societies and Social Simulation, 17,* 1–14.

Ronald, N., Sterling, L., & Kirley, M. (2007). An agent-based approach to modelling pedestrian behaviour. *International Journal of Simulation Systems, 8,* 25–38.

Steffen, B., & Chraibi, M. (2014). Multiscale simulation of pedestrians for faster than real time modeling in large events. *Cellular Automata Lecture Notes in Computer Science, 8751,* 492–500.

Tao, W., & Jun, C. (2009). An improved cellular automaton model for urban walkway bi-directional pedestrian flow. In *Measuring Technology and Mechatronics Automation International Conference.* Zhangjiajie, Hunan.

Treuille, A., Cooper, S., & Popović, Z. (2006). Continuum crowds. In *Proceedings of the 33rd International Conference on Computer Graphics and Interactive Techniques.* Boston, Massachusetts.

Vihas, C., Georgoudas, I., & Sirakoulis, G. (2013). Cellular automata incorporating follow-the-leader principles to model crowd dynamics. *Journal of Cellular Automata, 8,* 333–346.

Von Neumann, J., & Burks, A. (1966). *Theory of self-reproducing automata*. Urbana: University of Illinois Press.

Wainer, G. (2000). Improved cellular models with parallel cell-DEVS. *International Transactions of the Society for Computer Simulation, 17,* 73–88.

Wainer, G. (2009). *Discrete-event modeling and simulation: A practitioner's approach*. Boca Raton: CRC/Taylor & Francis Group.

Was, J., Lubas, R., & Mysliwiec, W. (2012). Proxemics in discrete simulation of evacuation. *Lecture Notes in Computer Science, 7495,* 768–775.

Wolfram, S. (1986). *Theory and applications of cellular automata*. Singapore: World Scientific.

Xia, Y., Wong, S. C., & Shu, C. W. (2009). Dynamic continuum pedestrian flow model with memory effect. *Physical Review E, 79,* 1–20.

Zeigler, B., Praehofer, H., & Kim, T. (2000). *Theory of modeling and simulation*. San Diego: Academic Press.

Zhan, B., Monekosso, D. N., Remagnino, P., Velastin, S. A., & Xu, L. (2008). Crowd analysis: A survey. *Machine Vision and Applications, 19,* 345–357.

Zhou, S., Chen, D., Cai, W., Luo, L., Low, M., Tian, F., Tay V. S., Ong, D. W., & Hamilton B. D. (2010). Crowd modeling and simulation technologies. *ACM Transactions on Modeling and Computer Simulation* 20, 1–35.

Gabriel A. Wainer FSCS, SMIEEE, received the M.Sc. (1993) at the University of Buenos Aires, Argentina, and the Ph.D. (1998, with highest honors) at UBA/Université d'Aix-Marseille III, France. In July 2000, he joined the Department of Systems and Computer Engineering at Carleton University (Ottawa, ON, Canada), where he is now Full Professor and Associate Chair for Graduate Studies. He has held visiting positions at the University of Arizona; LSIS (CNRS), Université Paul Cézanne, University of Nice, INRIA Sophia-Antipolis, Université de Bordeaux (France); UCM, UPC (Spain), University of Buenos Aires, National University of Rosario (Argentina) and others. He is one of the founders of the Symposium on Theory of Modeling and

Simulation, SIMUTools and SimAUD. Prof. Wainer was Vice-President Conferences and Vice-President Publications, and is a member of the Board of Directors of the SCS. Prof. Wainer is the Special Issues Editor of SIMULATION, member of the Editorial Board of IEEE Computing in Science and Engineering, Wireless Networks (Elsevier), Journal of Defense Modeling and Simulation (SCS). He is the head of the Advanced Real-Time Simulation lab, located at Carleton University's Centre for advanced Simulation and Visualization (V-Sim). He has been the recipient of various awards, including the IBM Eclipse Innovation Award, SCS Leadership Award, and various Best Paper awards. He has been awarded Carleton University's Research Achievement Award (2005, 2014), the First Bernard P. Zeigler DEVS Modeling and Simulation Award, the SCS Outstanding Professional Award (2011), Carleton University's Mentorship Award (2013), the SCS Distinguished Professional Award (2013), and the SCS Distinguished Service Award (2015). He is a Fellow of SCS.

Ala'a Al-Habashna received his Master of Engineering degree from Memorial University of Newfoundland in 2010, and his Ph.D. degree from Carleton University in 2018, both in Electrical and Computer Engineering. Currently, Dr. Al-Habashna is a Postdoctoral Fellow and an Instructor at Carleton University, Ottawa, Canada. He received the fellowship of the School of Graduate Studies at Memorial University of Newfoundland in 2010. Furthermore, Dr. Al-Habashna has won multiple awards including excellence and best-paper awards. He worked as a reviewer for many conferences and journals including the IEEE ICC and Multimedia Tools and Applications journal. Dr. Al-Habashna is also a Technical Program Committee in the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS). His current research interests include discrete-event modeling and simulations, signal detection and classification, cognitive radio systems, 5G wireless networks, communication networks architecture and protocols, IoT applications, and multimedia communication over wireless networks.