Cell-DEVS for Social Phenomena Modeling

Hoda Khalil, *Member, IEEE*, and Gabriel Wainer, *Senior Member, IEEE*

***Abstract*— Motivated by the need for formal methods as well as supporting tools to model and simulate social systems, we propose cellular discrete-event system specification as a formalism for modeling social systems. We also propose the use of a toolkit that implements the formalism of cellular discrete-event system specifications to implement and visualize models of social systems. We present examples of social system models that are different in sizes, nature, and rules controlling the interactions within those systems. We show that cellular discrete-event system specification with its unique features can successfully deal with the shortcoming of other modeling techniques. In addition, we show that together with its supporting toolkit, cellular discrete-event system specification is suitable for modeling, simulating, implementing, and visualizing social systems.**

***Index Terms*— Cellular discrete-event specification (Cell-DEVS), cellular automata (CA), DEVS, modeling and simulation (M&S), social systems.**

I. Introduction

MODELING was perceived in the past as a helping tool for software development by providing information about the consequences of building certain artifacts before they are actually made [37]. In other words, software systems that are subject to being implemented were the focus of modeling. Computer modeling then progressed into a way of understanding the environment and for predicting the evolution of different phenomena [69].

Different methods of computer modeling have been used to model, simulate, and, accordingly, understand environmental, biological, social, and other types of systems. For social systems, natural observations of social changes require performing social experiments overextended periods, which makes them sometimes infeasible and even not useful. In addition, natural observations and social experiments allow us to study social phenomena that already occurred [7], while, in reality, there is a clear need for predicting the future of these phenomena. Modeling and simulation (M&S) and available data collected from previous social events can be used for this purpose. Hence, computer modeling became the solution that allowed researchers and scientists to model and simulate complex social systems changes while overcoming the disadvantages of traditional analytical tools [41].

Many computer modeling theories have been used to model social systems (e.g., [6], [15], [20], and [26]–[28]), and among them is the cellular automata (CA), sometimes referred to as the computer scientist's counterpart of a physicist's field [58]. A CA can be defined as a grid of specified shape that evolves through a number of discrete-time steps according to a set of rules based on the states of neighboring cells [65]. Although CAs are very useful for modeling complex systems, they have limitations that restrict their applicability (see Section II-A).

Some of these limitations have been solved by combining CA and discrete-event specification (DEVS) models [62] (see Section II-B). The cellular DEVS (Cell-DEVS) formalism has its own supporting tools that facilitate its use, for instance, the CD++ framework [59].

Motivated by the continuous need to model social systems and inspired by the advantages that Cell-DEVS can bring to this field of study, we will show how Cell-DEVS and CD++ can be successfully used to formally model, implement, simulate, and visualize social systems. Note that the definition of a social system is "the patterned series of interrelationships existing between individuals, groups, and institutions and forming a coherent whole" [72], which means that it is not necessarily limited to systems representing interrelationships between people but can also include animals, for example, as we will show in our experiments.

We will start by introducing some background information about CA and Cell-DEVS (see Section II). In Section III, we review related work that uses CA and Cell-DEVS for modeling social systems. In Section IV, we show some examples of how to use Cell-DEVS to formalize social systems. In all the models presented in this article, we use CD++ as an implementation tool [59]. We model three main social systems: the spread of avian flu, survivorship of clonal organisms, and drug consumption in high-risk communities. For each model, we describe the conceptual model, formalize the definition of the model, show the rules implemented in CD++, and show the simulation results. All the simulations are reproducible, and the models are available through the advanced real-time and simulation (ARS) lab models' repository [73]. In addition, videos showing the simulation are available in our YouTube channel [2].

II. Background

A. Cellular Automata

The most common method for cellular computing is CA. CA is a regular uniform n-dimensional lattices with a discrete variable at each site (cell). The state of the CA is determined by the values of the variables in each cell [65]. The values of the variables in one cell at a certain timestamp are determined by the values of the variables in a finite set of cells comprising the neighborhood of that cell in the previous timestamp.

Manuscript received September 26, 2019; revised January 30, 2020; accepted March 11, 2020. Date of publication April 10, 2020; date of current version June 10, 2020. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC). *(Corresponding author: Hoda Khalil.)*

The authors are with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: hodakhalil@cmail.carleton.ca).

Digital Object Identifier 10.1109/TCSS.2020.2982885

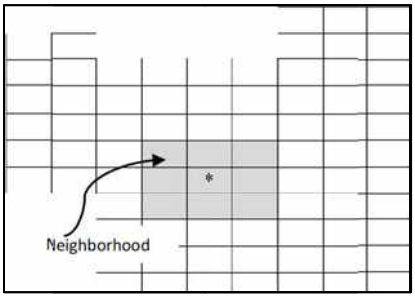

Fig. 1. CA.

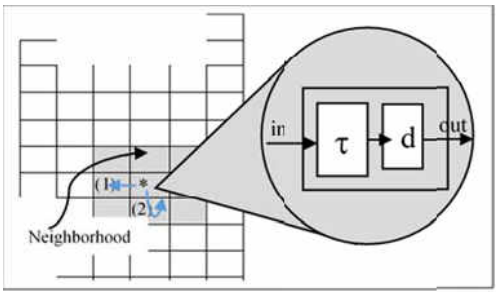

Fig. 2. 2-D Cell-DEVS informal sketch.

This update happens in a synchronous manner in all the cells of the CA. Hence, the CA evolves in discrete timestamps [66]. Fig. 1 shows a simple sketch of CA; the shaded area represents the neighborhood that affects the value of the core cell (starred) in the next timestamp.

The neighborhood can be defined differently depending on the model. The following is the definition of a conceptual CA:

$$\text{CCA} = \langle S, n, C, N, T, \tau, c^* Z_0^+ \rangle$$

where S is the set of states, n is the dimension of cell space, N is neighborhood set, T is the global transition function, τ is the local computing function, and $c^* Z_0^+$ is the discrete-time base for the CA.

On one hand, CA is simple enough to allow for detailed mathematical analysis. On the other hand, combining simple computations in individual cells makes it sufficient to model complex systems [65]. Being simple, homogenous, and yet capable of handling complex systems makes CA a suitable and popular technique for modeling and simulating complex phenomena in various domains [1], [5], [14]–[16], [18], [19], [29], [30].

CA is a common cellular method for cellular computing [39]. However, it has its drawbacks in terms of power, usability, and feasibility.

1) The fact that CA is based on discrete-time cell updates affects the performance and precision in modeling complex systems.
2) While CA updates are asynchronous in nature, it has to be implemented using synchronous digital computers.
3) It is hard to handle time-triggered activities in each cell using a CA [60].

B. Cell-DEVS

Cell-DEVS overcomes most of the limitations of CA by combining CA and DEVS formalism [70]. DEVS mathematically formalizes the definition of systems using a hierarchical composition of behavioral and structural models, referred to as atomic and coupled, respectively [62].

DEVS as a mathematical formalization is independent of any tool or language used for M&S. Consequently, different tools were developed to address the unique needs and uses of DEVS. A DEVS atomic model is formally described by

$$M = \langle X, S, Y, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, \text{ta} \rangle$$

where X is the input events set, S is the set of states, Y is the output events set, δ_{int} is the internal transition function, δ_{ext} is the external transition function, λ is the output function, and ta is the time advance function.

A DEVS atomic model is seen as having input and output ports to communicate with other models. δ_{int} changes the internal state of the model after a period defined by ta, while δ_{ext} specifies how the model reacts to inputs received from other models and collected through the input ports. The output ports transfer the output generated by the output function λ to the outer world. The output function is fired before changing the internal state of the model.

Cell-DEVS is an extension of the DEVS formalism that implements the general concept of CA while handling the shortcomings of the classic CA.

Fig. 2 shows an informal sketch that illustrates how Cell-DEVS is a combination of both DEVS and CA. A Cell-DEVS model is an n-dimensional lattice of cells where every cell is an atomic model, and the whole-cell space is a coupled DEVS model [62]. Each DEVS model in the lattice can interact with other cells through the model's interface (ports) and can interact with models outside the defined cell space [60].

The first step to define a Cell-DEVS model is to define the atomic model of each cell, as shown follows:

$$\text{TDC} = \langle X, Y, S, N, \text{delay}, d, \delta_{\text{int}}, \delta_{\text{ext}}, \tau, \lambda, D \rangle$$

where X is the input events set, S is the set of states, Y is the output events set, N is the set of input values, the delay is the type of delay, d is the delay duration, δ_{int} is the internal transition function, δ_{ext} is the external transition function, τ is the local computing function, λ is the output function, and D is the state's duration function [36].

The local computing function τ calculates the future state of the cell ($s \in S$). After a transport delay d elapses, the output values are transmitted. The inertial delay is used as a preemptive mechanism; it prevents any scheduled change from taking place upon receiving an external event from a neighbor cell before the scheduled time. This can result in the present state acquiring a different value [59]. Each state has a lifetime defined by the duration function D. When the duration is consumed, δ_{int} is fired to change the internal state of the cell. Before activating the internal transition function, the output function λ generates the model's output Y. When an external event arrives through the input ports, δ_{ext} is triggered.

Now, as the atomic model is defined for the cells, the next step is to define the complete cell space. A Cell-DEVS space is formally defined as a coupled model as follows [36]:

$$\mathrm{GCC} = \langle X_{\mathrm{list}}, Y_{\mathrm{list}}, I, X, Y, \eta, \{t_1, \ldots, t_n\}, N, C, B, Z \rangle$$

where X_{list} is the list of external input coupling, Y_{list} is the list of external output coupling, I is the set of states, X is the set of external input events set, Y is the set of external output events set, $\eta \in \mathrm{N}$ is the neighborhood size, $\{t_1, \ldots, t_n\}$ is the number of cells in each dimension, N is the neighborhood set, C is the cell space, B is the set of border cells, and Z is the translation function.

The border cells B can have different behavior than the rest of the cell space (or borders can be wrapped). The translation function Z defines the internal/external coupling of cells in the model; it translates the outputs of the m^{eth} output port in a cell C_{ij} into inputs to be fed to the m^{eth} input port of another cell C_{kl}.

We can summarize the ways Cell-DEVS overcomes the drawbacks of classic CA (see Section II-A) as follows.

1) Cell-DEVS provides asynchronous execution to model the asynchronous nature of a system.
2) This results in better execution times [36].
3) One can simply define complex timing conditions in cells using the available timing constructions.
4) Cell-DEVS allows for certain areas in the lattice to have different defined behavior, which is a more accurate representation of many natural phenomena.
5) Cell-DEVS inherits the property of being closed under coupling from DEVS, which makes it easier to integrate with other modeling formalisms [36], [60].

C. Modeling Social Systems

As mentioned in Section I, modeling is an interdisciplinary approach where mathematics and software integrate with other disciplines and sciences to understand, simulate, and predict different phenomena. Applications of social computing are expanding and expected to multiply [63]. Social changes and their impacts are among the disciplines where modeling is essential and can complement analytical methods to facilitate social studies. Social modeling has become even more relevant with the increasing risks, concerns, and unpredictable social phenomena [63] (e.g., privacy concerns, social media effects, the impact of technologies on cultural norms, and global warming and its effect on epidemics and social changes).

Different studies have been conducted to use computer models for social phenomena research. Examples include modeling residential migration [14], crowd behavior [26], [33], epidemiology [35], and social influence [9], [63]. Many modeling techniques aim at modeling and simulating social systems. This includes but not limited to agent-based modeling (e.g., [8] and [46]), regression models (e.g., [40]), CA (e.g., [54]), and DEVS (e.g., [52]). Regression modeling requires specifying explicitly the type of relation expected between variable; this cannot be specified properly in an area, such as social systems [17]. There is important research in social modeling that explores the theoretical aspects in terms of finding mathematical models that are suitable for social phenomena. For example, Curiel [13] introduces mathematical models to consider different social challenges. Other research projects on social modeling are concerned with the use of certain approaches for modeling specific social phenomena, for example, on the use of CA to model pedestrian behavior [16], [26], [32]. Other research focuses on how to narrow the gap between social scientists (who do not necessarily have the technical background to develop models as expected by simulation tools) and toolkits developers. For example, Pavon *et al.* [46] use a graphical agent-based language for specifying social models. The models can then be automatically transferred into a format that suitable to be fed to INGENIAS Development Kit [46]. There are numerous approaches for simulation in social sciences, but this research focuses on the use of spatial models, in particular those that can be formalized and subsequently executed by automated means derived from the formal specifications (such as CA, Cell-DEVS, or DEVS) to model social behavior. The rest of the literature discussed in Section III focuses on the body of related work that has been done in this area, including a discussion of the advantages of our approach as a complete modeling solution.

III. Related Work

We divide this section into two subsections. The first subsection discusses how CA has been used in the literature to model social phenomena (see Section III-A). Then, in Section III-B, we review the research work that uses Cell-DEVS and DEVS for social modeling.

A. CA for Social Modeling

CA is used extensively in the literature to model social behavior. For example, pedestrians' dynamics is one of the aspects that have been modeled using CA [21], [29], [32], [42], [68]. Burstedde *et al.* [75] represened pedestrians' behavior by using a 2-D grid of cells where each cell can be occupied by at most one pedestrian. The authors introduce the concept of floor field to determine the transition probabilities of pedestrians. The floor field concept considers the interactions between pedestrians as well as the geometry of the building. The floor field modifies the transition probabilities in a way that gives preference to moving in the direction of a larger field. The idea is to substitute the need for long-ranged interactions (interacting with people walking ahead) into a local interaction based on the trace that a pedestrian left a while back. This makes it more efficient and easier to calculate [26]. This concept later became a standard in pedestrians modeling [67], [68]. Researchers used the floor field CA modeling concept to model pedestrians' behavior extensively [16], [21], [25], [38], [47]. For example, Zheng *et al.* [71] proposed a fire evacuation model based on CA. Their work is based on the floor field concept proposed by Burstedde *et al.* earlier. The model shows that the number of pedestrians who manage to evacuate the room is highly related to the design of the room and the original fire location [71]. Fu *et al.* [22] also used the floor field concept to simulate the evacuation of 1000 pedestrians randomly distributed with different velocities and capabilities

to move. The simulation shows that the total evacuation time is mostly determined by the slower evacuees even if they compose the smaller portion of the population [22].

Another important social study area where CA modeling has been applicable is epidemiology. Researchers use CA to model the spread of different diseases among the population. White *et al.* [30] created a 2-D CA model to simulate epidemic spreading. Each cell represents a geographical area, and it is affected by only four neighbors. The authors report that their results mimic real-life scenarios for epidemic spread. The authors assume homogeneous distribution of the population; all the cells have the same number of residents. They also assume that the total population is constant [30]. Athithan *et al.* [5] proposed a dynamic CA for both homogenous populations and heterogeneous populations confined in small geographical locations (patches). The authors also consider the movement of the population across different locations and the effect of this on the epidemic behavior [5]. For more information on the effect of social interactions on the epidemic spread, the reader is referred to the relevant literature [19], [28], [29], [48], [49], [55].

Another area of social modeling that benefit from CA is peer influence on the social habits of individuals. Jackson *et al.* have developed the "Binge Drinking Cellular Automata" program to model the effect of peer pressure on the drinking behavior of undergraduate students. The authors model the good and bad influence of individuals on other members of the group, and they discuss how this affects the drinking habits of the whole group. Dabbaghian *et al.* [15] used CA to model the transformation of members of a community from being healthy individuals to using drugs due to peer influence. The authors also examine the transition of individuals into being heavy drug addicts and committing crimes to afford their expensive addiction habit. They also examine positive peer influence and discuss how it can help the rehabilitation of an individual so they can become healthy [15]. Another interesting social study is conducted by Osipian [44] where the author models the corruption in educational organizations. The author offers a framework based on CA to model corruption in large education institutes and to predict the possible scale of corruption within the organizations. The author studies the effect of the economic benefits, the cost of corruption (consequences), and the effect of peer pressure on the moral constraints of individuals [44].

The examples mentioned in this section are only one of the areas where CA is used to model different social phenomena. The research literature is rich with other studies that utilize CA to model social behavior (e.g., [20], [27], [28], and [50]). In the next section, we will focus on social studies that use DEVS and Cell-DEVS as a mean of M&S.

B. DEVS and Cell-DEVS for Social Modeling

The work of Seck *et al.* [52] uses DEVS to model human behavior for the sake of achieving a high level of realism for computer-generated forces (CGFs). Seck *et al.* [52] proposed a framework for modeling human behavior under constraints such as stress and tiredness. The behavioral atomic model introduced by the authors represents a set of tasks (states) and the set of transitions between the tasks belonging to the same mission. They introduce a stress state model consisting of five states and four categories of external events that are especially applicable for the military. Those events transfer the model from one state to the other. The duration of a task and the difficulty level of a task that an individual performs within the system are the factors considered when calculating tiredness. The proposed framework paves the way for adding new performance moderators and complexities [51].

As a part of the poly-functional intelligent operational virtual reality agents (PIOVRA) project [12], Seck *et al.* [51] added to their previously proposed framework openness as an example of what they call personality filters that affect human performance and behavior. This is done by introducing a cognitive complexity variable to the tasks integrated into the previously explained framework. Different individuals represented in the model will perform the tasks differently in terms of time and decision-making based on the filters incorporated in the complexity variable (openness in this example) [51].

The objective of Simulation Constructive et Modélisation des effets des Opérations d'influence dans les réseaux Sociaux (SICOMPRE) [74] is to model a social network, simulate the effect of information on them, and observe how the propagation of information affects this population [20]. As a part of SICOMOPRE, Bouanan *et al.* [10] presented a Cell-DEVS model for how humans deal with information, and it is propagated [10]. Like us, the authors use CD++, an M&S tool for DEVS and Cell-DEVS [59], to build their models. Each cell represents an individual with a set of attributes (e.g., gender and age), which are combined to form a cell space representing the network of the population [10], [11] modeled with Cell-DEVS.

Behl *et al.* [6] discussed the DEVS and Cell-DEVS models and present a schema to study how social interaction affects the formulation of human opinions using Cell-DEVS and agent-based modeling. They extend the work of Bouanan *et al.* [11] by testing the effect of varying some parameters in the model; they perform their experiments using different population sizes, different neighborhood shapes, and variable thresholds when an individual adopts a new opinion [6].

DEVS and Cell-DEVS have also been used to model other social phenomena, such as epidemics and the spread of viruses through social interactions between animals or humans. For example, based on a CA model of a population affected by the plague [18], Shang and Wainer [53] present a Cell-DEVS model to describe the interactions between individuals and viruses using CD++. Individuals and viruses reside in the different cells of the model, and the rules applied to implement interactions between the cells consider age, people reproduction, change in the state of the virus (active and inactive), and competition between viruses and individuals (which one beats the other) [53].

Crowd behavior is another area where Cell-DEVS has been used as an approach to model social behavior. Al-Habashna and Wainer [1] propose Cell-DEVS models to model and simulate movements of crowds. The authors present models for handling evacuation and movement in multiple floor

buildings. The aim of the model is to use the results of the simulation to provide recommendations for architects in the design process of multiple floor buildings. The authors conclude that Cell-DEVS can be adequately used to model pedestrians' behavior [1].

As mentioned earlier, Cell-DEVS is suitable for modeling and simulating complex systems with advantages over CA. However, up until now, the M&S research literature has not shown how DEVS and Cell-DEVS can be applied for performing various categories of social systems research. In addition, the tools needed to perform such M&S to facilitate social systems experimentations have not been explored in this context.

The use of Cell-DEVS provides advantages compared with other modeling approaches. Cell-DEVS is an extension of DEVS, which provides the advantages of a discrete-event approach in terms of execution performance. Discrete-event models evolve in continuous time. Events are instantaneous and can occur asynchronously at unpredictable times. DEVS simulators use hierarchical schedulers of events that activate the corresponding submodels. The schedules allow skipping periods of inactivity in the simulation. This is a clear departure from time-based approaches (including CA) where all components are updated at the same time, even when they do not need to. Another advantage is that expressing a timing delay is done in a natural fashion, allowing the modeler to reduce the development time related to timing control programming. Another advantage is that the complexity of this physical phenomenon is such that the inclusion of other external influences is difficult to be considered. As Cell-DEVS models are DEVS models, we can combine the social models with other external models defined in a different formalism, using DEVS as a mechanism to assemble models in a seamless fashion [62]. Other existing formalisms can be expressed as DEVS models (including Petri Nets, FSM, State Charts, and timed automata). Consequently, a modeler can express different properties in an adequate formalism and use DEVS hierarchical coupling as integration. In this way, one can take advantage of the current expertise of scientists in different domains. Modelers can describe individual components of cellular models using their own methods, which could result in the enhanced model definition and would help to bridge the gap between traditional modeling techniques and cellular computing. Other advantages can be summarized as follows.

1) Our solution, compared with the use of tools and informal methods, provides the advantages introduced by formal mathematical specifications. A formal model is simpler to verify and then can be validated, improving the error detection process and reducing testing time. DEVS models are closed under coupling; therefore, a coupled model is equivalent to an atomic one, improving reuse. DEVS supplies facilities to translate the formal specifications into executable models. In this way, the behavior of a conceptual model can be validated against the real system, and the response of the executable model can be verified against the conceptual specification.
2) DEVS is a complete M&S technique. It provides a way to specify models that can be coupled into higher level ones, which are later simulated by independent abstract entities (in single-processor or parallel architectures). Each model can be associated with an experimental framework, allowing the individual testing of components and making integration testing easier.
3) DEVS can be applied using predictive quantization of arbitrary ordinary differential equation models. Quantized models improve substantially performance with bounded error.

This research, thus, focuses on how to provide a complete solution that covers the entire process for modeling and simulating social systems using a formal approach and automated means for the simulation process. The solution that we propose covers the process starting by formalizing the model using Cell-DEVS theoretical concepts and using CD++ to implement the formal models, execute the simulation, and, finally, visualize the results.

Based on previously created CA models [15], [19], [38], in Section IV, we present DEVS and Cell-DEVS models in three different fields that belong to the social category of models. We use the CD++ toolkit for M&S [59]. We prove through our models that DEVS and Cell-DEVS formalisms from one side and CD++ as an implementation toolkit from the other are suitable and effective in modeling complex phenomena in different social contexts that involve both the behaviors of humans as well as other living creatures.

IV. Social Applications Examples

In this section, we discuss three different examples representing social changes and their consequences in different communities with variable characteristics. First, we present a group of models of the spread of avian flu (H5N1) in poultry and animals (see Section IV-A). The spread of the virus happens through interactions between birds, or birds and animals. This model shows how social interactions among animals affect their wellbeing as well as the health of individuals in direct contact with the infected animals. In addition, it has been shown that there is a resemblance between the spread of viruses and information propagation in social networks, and therefore, models similar to the virus-spreading models can be used to simulate the spread of information in communities [9]. Second, we propose models of the phenomena of the survivorship of colonel organisms (see Section IV-B). Through the proposed models, we explore how the interaction between different members of the colony affects the survivor of members of the colony. Finally, in Section IV-C, we discuss how drug usage evolves in high-risk communities and can affect the crime rate using n-dimensional Cell-DEVS.

It is evident in our experiments presented in this section that in all the models, Cell-DEVS allows us to separate the formalism from the implementation, which facilitates describing the problem clearly and independently of the implementation mechanism.

After the formal specification is defined, we implement our models using CD++ [58], [59], a toolkit that implements DEVS and Cell-DEVS theoretical concepts. Using the toolkit, one can define an atomic model using C++ or graphical

notation, while Cell-DEVS models are specified using a specification language provided by the tool. Reference [59] has more details about using CD++ for M&S.

A. Avian Influenza

The first group of models that we introduce in this section studies diverse aspects of the spread of avian influenza (H5N1). We first propose the basic scenario of the spread of the virus with different probabilities using simple neighborhood definition (see Section II-B). Then, we introduce boundaries to the model to prevent the spread of the virus outside a certain community. Finally, we expand the neighborhood, introduce the concept of distinct types of neighbors, and add states to represent immunization and death as a possible consequence of the virus infection. We must clarify that the size of the cells is not representative of spatial size, but of topology and location. That is, one cell in the model might represent 1 km^2 and the other 1000 km^2. The neighborhood relationship represents the movement of birds between areas of influence, which has been used to describe the migratory behavior of the birds. This type of migratory specification has been presented in various CA models of birds [34], [54], [67] and other animals [43]. In fact, CAs have been used on a wider scale to model population movement and disease transmission in general [23].

1) Problem Definition: H5N1 is a highly contagious viral disease that spreads among poultry and may attack other species of animals as well. The virus first emerged in 1996 in Eastern Asia and has been spreading among birds' population since then. Outbreaks caused by the virus resulted in animal health and economic crises worldwide. The virus also attacks humans through direct contact with living infected poultry or animals and, in unusual cases, through human-to-human interaction [45]. However, when it attacks people, the mortality rate becomes 60% [64]. The preemptive actions in terms of control and immunization can help to avoid the grave consequences of the virus spread.

H5N1 has received attention from scholars as a potential pandemic threat. Due to the threats, consequences of the virus spread, and the unique nature of the virus, different attempts have been made to model the spread of the virus [32], [49]. In our example, we are not trying to discuss the biological aspects of the spread of avian influenza. Instead, we focus on modeling the social aspect of epidemiology, in terms of the spread of the disease in a social system and possible ways to control it.

2) Conceptual Model: The model that we propose, based on the one in [55], is presented to show how DEVS, Cell-DEVS, and CD++ are adequate for modeling social systems. The model we use here is based on the work done by Situngkir [55] to model spatial epidemiology using CA. More sophisticated models can be introduced by using and expanding the main concepts that we use here. The avian flu model involves the transmission of the virus among animals living in the same area. We model a population of 1225 agents (individuals) represented as a grid. We limit the discussion to birds for simplicity; however, including humans and other animals in the model is possible with modifications. Using Cell-DEVS, we construct a 2-D 35×35 square grid of cells where each cell represents an individual acting as part of the population. We assume that the individuals are residing in the same area as direct contacts.

Everyone residing in a cell can be in one of the few states as will be described in the next section. The state of the individual (infected, susceptible, and so on) changes based on the individuals in the cell's neighborhood surrounding that individual.

3) Formal Models Specifications and Simulations: We study three different main models in this section. First, we model a basic case where we have only two states for infected and susceptible cells. We assume that each cell is affected by a neighborhood of nine near neighbors. Second, using the same neighborhood of the basic case, we model the population when boundary cells are introduced to control the spread of the disease. Finally, we expand our neighborhood to consist of 24 neighbors classified in two different proximities from the core cell. We also introduce three more states in this last model as will be explained in Section IV-A3.b.

a) Basic scenario: The first model shows the spreading of the virus with different probabilities. We use two states to model the spread of flu: susceptible (S) and infected (I). A susceptible cell has not been infected, but there is a probabilistic potential that it can be infected in the future, while an infected cell is a cell that caught the virus. The neighborhood for the core cell is the Moore neighborhood, which includes the cell's nine near neighbors.

The following is the formal definition of the 2-D Cell-DEVS space in the first out of three models, as defined in Section II-B:

$$\text{Avian\_Basic} = \langle X_{\text{list}}, Y_{\text{list}}, I, X, Y, \eta, \{t_1, t_2\}, N, C, B, Z \rangle$$

where $X_{\text{list}} = Y_{\text{list}} = \{\emptyset\}, I = \{0, 2\}, X = Y = \emptyset, \eta = 9$, $t_1 = t_2 = 35$, and $N = \{(-1, 0), (0, -1), (0, 1), (1, 0), (0, 0), (-1, -1), (-1, 1), (1, -1), (1, 1)\}$.

For the basic scenario model, we use two states only ($I = \{0, 2\}$), where 0 is the state of the susceptible inhabitant and 2 represents an infected cell. Infected cells are responsible for spreading the flu. In this model, we consider an area containing 1225 inhabitants, and one of them is the root cause.

The transition (S to I) indicates that the inhabitant was infected by the avian flu and that it will spread the flu to others. We initially set one inhabitant as infected. The rule in CD++ can be written as follows:

```
rule: 2 100 {(0,0) = 0 and stateCount (2) > 0
                      and random < 0.3}
rule: {(0,0)} 100 {t}.
```

The first rule indicates that if the core cell is not infected (0), and there is one or more neighboring cells infected (value of 2), then the core cell will change randomly to become either susceptible (0) or infected (2). The second rule means that in all other cases, the cell will keep its current value. Fig. 3 shows the progress of a simulation scenario of the basic model with a diffusion rate of 0.3, where the susceptible inhabitant cells are shown in light shade and the infected cells are dark.

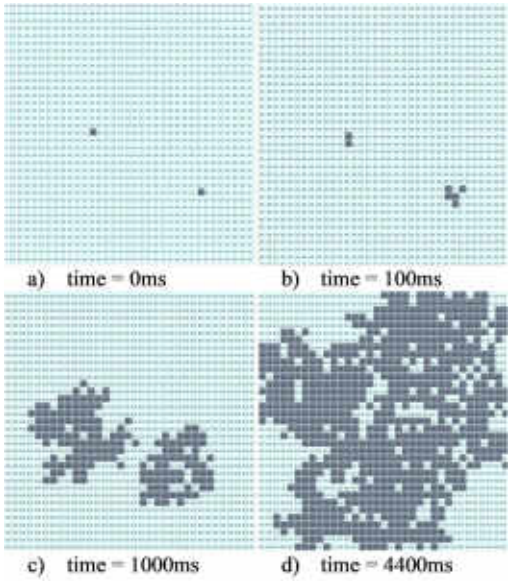

Fig. 3. Avian flu basic scenario: snapshots of the simulation at different times with diffusion rate 0.3. (a) time = 0 ms. (b) time = 100 ms. (c) time = 1000 ms. (d) time = 4400 ms.

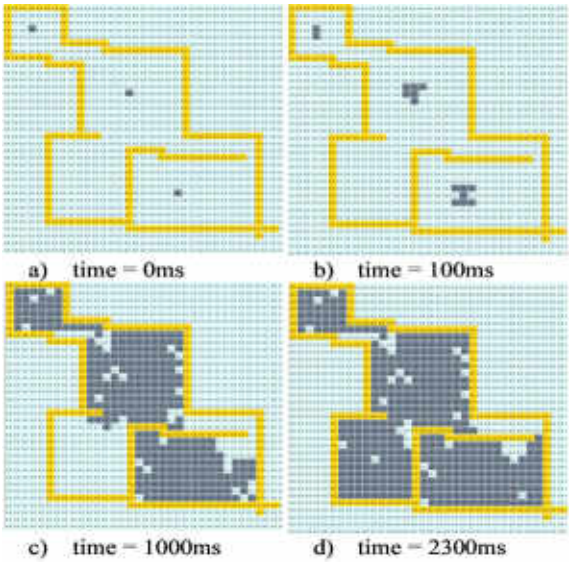

Fig. 4. Avian flu boundary scenario: snapshots of the simulation at different times with diffusion rate 0.5. (a) time = 0 ms. (b) time = 100 ms. (c) time = 1000 ms. (d) time = 2300 ms.

The simulation starts at time 0 where there are two cells initially infected [see Fig. 3(a)] and ends at time 4400 [see Fig. 3(d)] when there are no more events to execute. Fig. 3(b) and (c) shows the spread of the virus at times 100 and 1000, respectively.

b) Introducing boundaries: The second model shows the spread of the virus with different probabilities inside a predefined boundary and is formalized as follows:

$$\text{Avian\_Boundaries} = \langle X_{\text{list}}, Y_{\text{list}}, I, X, Y, \eta, \{t_1, t_2\}, N, C, B, Z \rangle$$

where $X_{\text{list}} = Y_{\text{list}} = \{\emptyset\}, I = \{0, 2, 5\}, X = Y = \emptyset, \eta = 9, t_1 = t_2 = 35$, and $N = \{(-1, 0), (0, -1), (0, 1), (1, 0), (0, 0), (-1, -1), (-1, 1), (1, -1), (1, 1)\}$, where the state 0 represents the susceptible inhabitant, 2 represents the infected ones who are responsible for spreading the flu, and 5 represents the boundary. This model uses zones in CD++, which helps modeling behavior. The boundary model defines a zone where the flue spreading rules are applicable (and different rules outside that zone). The following shows how zones are defined. This means that rules labeled "flu" are only applicable inside the defined zones

```
zone : flu {(0,0)..(0,7)}
zone : flu {(1,0)..(7,0)}
zone : flu {(1,7)..(5,7)}
zone : flu {(7,1)..(7,10)}
zone : flu {(5,8)..(5,12)}
zone : flu {(8,9)..(17,9)}
...
zone : flu {(7,22)..(17,22)}
zone : flu {(18,22)..(18,31)}
zone : flu {(19,31)..(32,31)}
zone : flu {(30,5)..(30,15)}
zone : flu {(31,15)..(31,32)}.
```

The boundary cells can be mapped to boundaries around the infected avian community that is introduced to limit the spread of the disease. Fig. 4 shows how the virus spreads with a 0.5 diffusion rate at various times when boundary areas are introduced. The simulation shows how introducing a boundary area can stop the disease from spreading to the rest of the population after 2300 time units when the simulation stops.

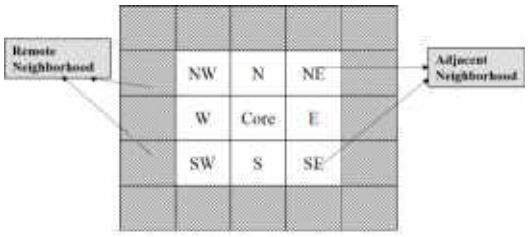

Fig. 5. Avian flu: 5 × 5 neighborhood.

A video of the simulation for this model can be found in the ARSLab YouTube channel [3].

c) Larger neighborhood and more states: In the next iteration, we introduce a new neighborhood definition with two kinds of neighbors. The first type we have is the "adjacent neighbor" that is next to the core cell, and the second type is the "remote neighbor" that is one cell farther (see Fig. 5). Inhabitants have 24 contacts that can be used to spread the flu: 16 of them are remote, while 8 are adjacent. Adjacent neighbors have more chances of infection compared with the remote ones.

The following is the formal definition of the model:

$$\text{Avian\_Advanced} = \langle X_{\text{list}}, Y_{\text{list}}, I, X, Y, \eta, \{t_1, t_2\}, N, C, B, Z \rangle$$

where $X_{\text{list}} = Y_{\text{list}} = \{\emptyset\}, X = Y = \emptyset, I = \{-1, 0, 1, 2, 3\}, \eta = 25, t_1 = t_2 = 35$, and $N = \{(-1, 0), (0, -1), (0, 1), (1, 0), (0, 0), (-1, -1), (-1, 1), (1, -1), (1, 1), (-2, -2), (-2, -1), (-2, 0), (-2, 1), (-2, 2), (-1, -2), (-1, 2), (0, -2), (0, 2), (1, -2), (1, 2), (2, -2), (2, -1), (2, 0), (2, 1), (2, 2)\}$.

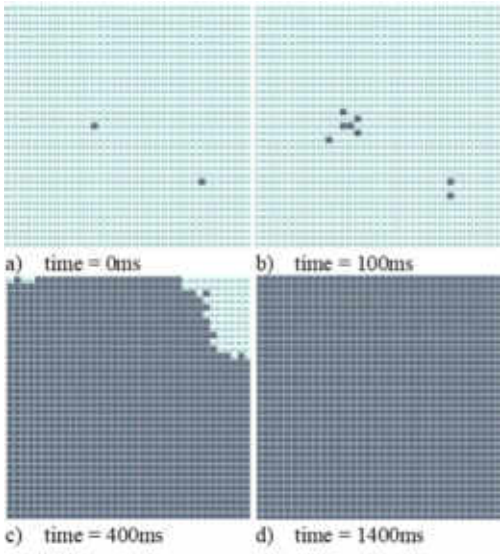

Fig. 6. Advanced avian flu: diffusion rate calculated based on the outer and inner neighbors. (a) time = 0 ms. (b) time = 100 ms. (c) time = 400 ms. (d) time = 1400 ms.

Here, the states are defined with a value of 0 for the susceptible inhabitants, 2 for the infected ones who are responsible for spreading the flu, 1 is a questionable cell, −1 is an immune cell, and 3 is a dead cell. Questionable, immune, and dead states are explained later in this section.

The following is the rule that defines the spread of the flu based on the new neighborhood for this model and before adding the new states:

```
rule: 2 100 {(0,0)=0 and stateCount(2) > 0 and
   random < #macro(Spreading) * (
     #macro(Inner) * #macro(inner_2)/8 +
        #macro(Outer) * #macro(outer_2)/16)}.
```

Inhabitants are infected by the flu, which is spread by animals. The possibility of transition is calculated based on the number of infected inhabitants in the neighborhood along with the diffusion rate with a higher probability of spreading when the infected cells are closer to the core cell. The diffusion rate is calculated as shown in the abovementioned rules. The Appendix shows the different macros and how the diffusion is calculated. Inner and outer indicate the possibility of infection by the influence of the adjacent and remote neighbors, respectively. The variables *inner_2* and *outer_2* indicate the number of inhabitants infected in the set of "adjacent neighbors" and the set of "remote neighbors," respectively. Fig. 6 shows the results when introducing the new neighborhood design.

As shown in Fig. 6, when introducing remote neighbors, the disease spreads faster compared with the basic model of Fig. 3 where only adjacent neighbors are considered.

In the next iteration of the model, we still use the neighborhood of Fig. 5. However, we notice the risk of flu and possible steps that may be taken to immunize the inhabitants. Inhabitants will try to get immune after flu symptoms have been noticed. To do so, we add new states defining questionable and immune individuals; this is done simply by introducing new state values. A cell becomes questionable

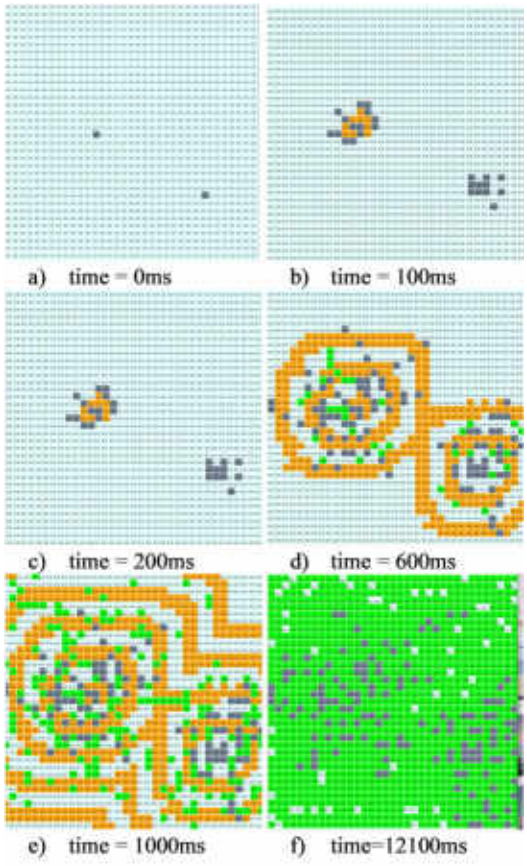

Fig. 7. Avian flu: 24 cells neighborhood: immunization and vigilance of the inhabitants is considered after noticing symptoms. (a) time = 0 ms. (b) time = 100 ms. (c) time = 200 ms. (d) time = 600 ms. (e) time = 1000 ms. (f) time = 12 100 ms.

based on probabilities calculated for immune cells (0.1, as shown in the Appendix). The vigilance of the inhabitants is also considered while considering its 24 neighbors' states and their proximity to the core cell. The immune state is a virus-free state. This means that they may spread no or less flu virus.

The rules for this model are defined as follows:

```
rule: 1 100 {(0,0)=0 and (#macro(Inner) *
   #macro(inner_2) + #macro(Outer) *
    #macro(outer_2) >3) and random < (
     #macro(Vigilance) * (#macro(Inner) *
      #macro(inner_2)/8 + #macro(Outer) *
       #macro(outer_2)/16))}
rule: 1 100 {(0,0)=0 and (#macro(Inner) *
   #macro(inner_1) + #macro(Outer) *
    #macro(outer_1) > 0) and random < (
     #macro(Vigilance) * (#macro(Inner) *
      #macro(inner_1)/8 + #macro(Outer) *
       #macro(outer_1)/16))}
rule: {if (random < #macro(Immune), -1,0)}
            100 {(0,0)=1}
rule: 2 100 {(0,0)=0 and stateCount(2) > 0
   and random < #macro(Spreading) * (
    #macro(Inner) * #macro(inner_2)/8 +
     #macro(Outer) * #macro(outer_2)/16)}
rule: {(0,0)} 100 {t}}.
```

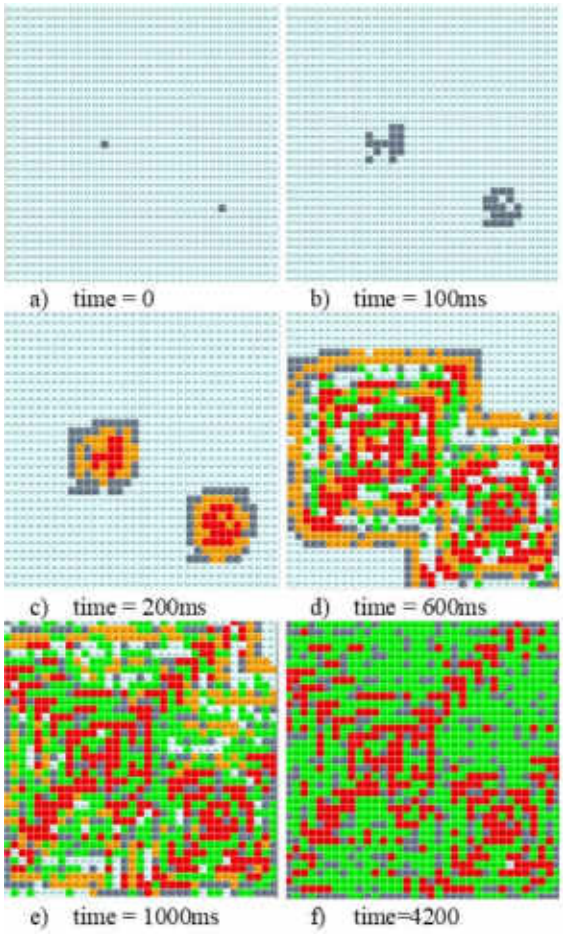

Fig. 8. Avian flu: neighborhood of 24 cells: inhabitants who are not immunized die due to the dangerous flu. (a) time = 0 ms. (b) time = 100 ms. (c) time = 200 ms. (d) time = 600 ms. (e) time = 1000 ms. (f) time = 4200 ms.

The simulation results of this model are shown in Fig. 7.

In Fig. 7(a), the simulation starts with two cells initially infected. Then, in Fig. 7(b), the virus spreads and some cells become questionable. In Fig. 7(d), immunization starts to appear. This limits the spread of the disease as we can observe comparing Figs. 6(d) and 7(f). While in Fig. 6(d), the whole population is infected, Fig. 7(f) shows a few infected individuals surrounded by the immune population.

In the last model, we add a dead state and one new rule to the previous model. We assume that inhabitants who do not get the treatment die because of the dangerous flu. The following is the rule added for this model:

```
rule: {if (random < #macro(dead), 3, 0)} 100
          {(0,0)=2}.
```

Fig. 8 shows the simulation results when considering death due to a lack of immunization. The simulation starts with two infected cells [see Fig. 8(a)]. Then, the disease starts to spread among the neighboring cells [see Fig. 8(b)]. The dead cells start to appear in Fig. 8(c) at the center of the infected population. The dead cells are surrounded by questionable cells at this point in the simulation. Then, in Fig. 8(d), some immune cells start to appear, but also the number of dead individuals increases. Then, gradually, the number of immune cells increases at a faster rate [see Fig. 8(e)]. At the end of the simulation, no more questionable cells are present. This is shown in Fig. 8(d), where there are only three types of cells: infected, immune, and dead cells.

4) Discussion: We presented Cell-DEVS to model the spread of avian flu using a different number of states, different rules, and different neighborhoods. Our experiments show how different parameters that affect the spread of the disease can be modeled and how the effects of such parameters can be simulated using Cell-DEVS as formalism and CD++ as an implementation tool.

For example, we were able to represent different probabilities of the spread of disease based on the spatial proximity of the individuals from the infected person using neighborhoods modeling in Cell-DEVS. Similarly, CD++ allowed for the implementation of zones (quarantine areas), outside of which the disease cannot spread. Furthermore, using Cell-DEVS and CD++ to define and implement multiple states allowed for representing stages of the disease (e.g., immune, suspectable, and infected). The models can be used to simulate aspects of social interaction that can affect the spread of a viral infection.

B. Genet and Ramet Survivorship

Genet is a single genetic individual that can be comprised of separate colonies (ramets) [57]. The models we present in this section try to model the survival behavior of genet nodes. We start by defining the problem. Then, we introduce a general conceptual model. Then, we present the simulation results of a simple model with only one ramet and a small cell space (10×10), then, a model with two types of ramets with a larger lattice (15×15), and finally, a complete model with a 60×60 lattice and nine different ramets located at random.

1) Problem Definition: Clonal organisms are comprised of modules that are either semi-independent or partially independent to the level that they can survive the death of part of the colony. The study of the traits and behavior of such colonial organisms was introduced by Inghe [31], where the authors presented a 2-D CA to analyze this problem. In this section, we build a 2-D Cell-DEVS model based on [31] to explore the interaction and survival behavior among different genet nodes.

2) Conceptual Model: The main model we present includes nine different genets that are spread randomly in a cell-space of 60 × 60. The model shows how each genet colony interacts with the other, showing whether the colony expands or decreases in size. The original purpose is to show the relationship between clonal growth (i.e., producing laterally spreading shoots that become at a later stage physically independent ramets [31]) and genet survival under spatially localized disturbances where the size of the disturbance can be varied.

3) Formal Model Specifications and Simulations: For the model, survivorship will be compared in relation to the population. The model is represented by a cell-space that consists of 60 rows × 60 columns. The cells have two states, either empty or occupied by one ramet. The cell-space is wrapped. The top row is wrapped with the bottom, and the most-right column is wrapped with the most-left column.

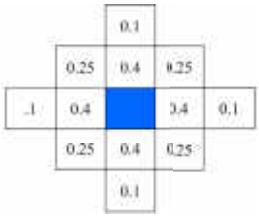

Fig. 9. Genet and ramet: 13 members' neighborhood.

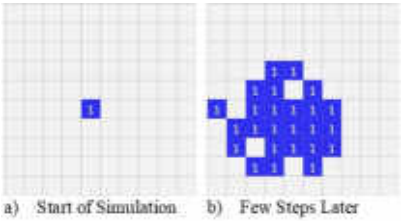

Fig. 10. Single ramet simulation. (a) Start of simulation. (b) Few steps later.

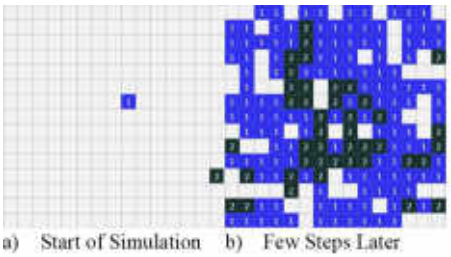

Fig. 11. Two ramet colonies simulation. (a) Start of simulation. (b) Few steps later.

The neighborhood of each cell includes 13 cells (including the core cell itself); the shape of the neighborhood is shown in Fig. 9. If the cell is empty and that there is only one growing ramet in the neighborhood, then the ramet will colonize (growth rule) the empty cell with a probability P_{I}. The probability P_{I} for each cell in the neighborhood is shown in Fig. 9.

If more than one growing ramet exists in the neighborhood, then they would compete in colonizing the empty space. In this case, the probability for each competing ramet will be P'_{I}, which is given by

$$P_{I'} = \left(\frac{P_I}{\sum_{i=1}^{12} Pi}\right)\left(1 - \prod_{i=1}^{12}(1 - P_i)\right).$$

The death rules assumed in this model is that the death of an individual is caused by an overcrowded population. The simulation will start by randomly sowing nine genets, each consisting of one ramet and then exposing them to the death rule. Subsequent iterations will then be conducted by applying the growth rule, followed by the death rule.

We use, for this example, three Cell-DEVS coupled models, but they are all similar with few differences in the cell space dimensions and the initial values. The main Cell-DEVS coupled model is defined as follows:

$$\text{Genet\_Ramet} = \langle X_{\text{list}}, Y_{\text{list}}, I, X, Y, \eta, \{t_1, t_2\}, N, C, B, Z\rangle$$

where $X_{\text{list}} = Y_{\text{list}} = \{\emptyset\}$, $I = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $X = Y = \emptyset$, $\eta = 13$, $t_1 = t_2 = 60$, $N = \{(-2, 0), (-1, -1), (-1, 0), (-1, 1), (0, -2), (0, -1), (0, 0), (0, 1), (0, 2), (1, -1), (1, 0), (1, 1), (2, 0)\}$, $C = \{C_{ij}/i \in [0, 59] \wedge j \in [0, 59]\}$, $B = \{\emptyset\}$ is a wrapped border in our case.

There is only one set of 14 rules in total. The first 12 rules are similar in that they calculate the probability of cell (0, 0) being colonized. Each rule corresponds to one of the neighbors.

Rule 1, which calculates the probability of the neighboring cell (−2, 0) colonizing cell (0, 0), is shown in the following as an example:

```
rule : {(-2,0)} 1000 {(0,0)=0 and (-2,0)>0 and
 random < ((0.1/(if(((-2,0)>0),0.1,0) +
  if(((-1,-1)>0),0.25,0)+ if(((-1,0)>0),0.4,0)
   + if(((-1,1)>0),0.25,0) +
   if(((0,-2)>0),0.1,0)+ if(((0,-1)>0),0.4,0)+
    if(((0,1)>0),0.4,0)+ if(((0,2)>0),0.1,0)+
if(((1,-1)>0),0.25,0)+ if(((1,0)>0),0.4,0)+
 if(((1,1)>0),0.25,0)+ if(((2,0)>0),0.1,0)))
  *(1- (if(((-2,0)>0),0.9,1) *
    if(((-1,-1)>0),0.75,1) *
 if(((-1,0)>0),0.6,1)* if(((-1,1)>0),0.75,1)*
  if(((0,-2)>0),0.9,1)* if(((0,-1)>0),0.6,1)
  * if(((0,1)>0),0.6,1)* if(((0,2)>0),0.9,1)*
 if(((1,-1)>0),0.75,1)* if(((1,0)>0),0.6,1)*
if(((1,1)>0),0.75,1)* if(((2,0)>0),0.9,1)))))}.
```

If the conditions of rule one are met, the contents of cell (0, 0) will be replaced with the contents of cell (−2, 0). Rule 13 is shown below where there is a 40% chance of killing the ramet in cell (0, 0) if there are less than 3 empty cells around it

```
rule : {0} 1000 {((0,0) > 0) and
      (falseCount < 3) and (random < 0.4)}.
```

We first simulate a single ramet test that uses only one type of ramet and a small cell space with the dimensions (10, 10). This test is to visualize the spread of a ramet. Fig. 10 shows the cell space at the start of the simulation [see Fig. 10(a)] and a few steps afterward [see Fig. 10(b)].

The two colonies' simulation uses two types of ramets, each producing a colony. This visualizes the interaction between more than one colony. The dimension of the cell space here is 15 × 15. Fig. 11 shows the cell space at the start of the simulation and few steps afterward. The results show a colony moving into the center of another colony and dispersing, while the surrounding colony has spread throughout the cell space and has the chance to take over the contained colony.

Once these basic scenarios were completed, we executed various complex scenarios. Fig. 12 shows one of those scenarios. A simulation video showing how the colonies interact and evolve is available in the ARSLab YouTube channel [4]. Fig. 12 shows the colonies near the start, middle, and end of the simulation. The first stage shows when the simulation starts with nine ramets placed randomly in a 60 × 60 cell space [see Fig. 12(a)]. The second stage is the rapid growth and colonization of the whole cell space by the ramets until different colonies collide [see Fig. 12(b)]. The third stage occurs when

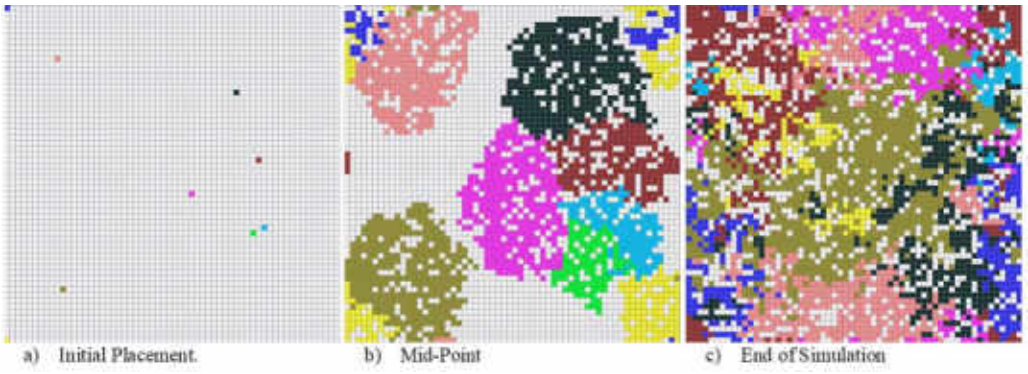

Fig. 12. Ramet colonies simulation at different timestamps. (a) Initial placement. (b) Midpoint. (c) End of simulation.

the colonies start competing for empty cells near each of the colonies' borders; colonies here may increase or decrease in size, shift places, diffuse into one another, and/or die out [see Fig. 12(c)]. Note how the colonies shift, as we can see clearly by comparing the colonies from Fig. 12(b) and (c). One of the colonies [close to the bottom right corner of Fig. 12(b)] can be seen dying out by comparing Fig. 12(b) and (c). We can deduct from this simulation that colonies that start with no competition tend to spread at a faster rate than colonies with the competition. In addition, colonies that start by being placed close to another colony tend to be smaller and are more prone to dying out than colonies that start by being placed further apart.

4) Discussion: In this section, we presented a Cell-DEVS model based on the work presented by Inche [31] to explore the relationship between clonal growth and the genet survival through interactions between the different colonies. The simulation shows that colonies placed apart are more likely to survive. Our model can be improved by adding new death rules for example. However, we show in our experiments that Cell-DEVS formalism and CD++ tool kit are suitable for modeling, simulating, and implementing the survivorship of the ramet as a species. Using Cell-DEVS, we could model a different kind of neighborhood that did not resemble the rectangular shape represented in the previous example. We could also define, simulate, and implement a wrapped cell-space that modeled the boundaries of the represented area. Using CD++ allowed for defining different initial values of the cells to simulate the diverse types of ramets. In addition, we were able to implement the possibility of a cell colonizing its neighbor based on different probabilities using the unique features CD++.

C. Influence of Drug Use for Crime in High-Risk Communities

In this section, we present three different models to study the spread of drug usage, addiction, and crime in a high-risk community. We first introduce a simple scenario where a person starts consuming drugs, while the second model shows the evolution of a drug user into an addict. The third model focuses on the transition of an addict into criminal activities and considers the possibility of treating a drug addict. As described in previous sections, we start by defining the problem and introducing the conceptual model. Then, we formally define the models and show the simulation results.

1) Problem Definition: Drug addiction has been identified as a major cause of different criminal acts. In this case, the addicts conduct crimes for the sake of covering the cost of the expensive drugs that they are dependent on [56]. In this section, we model the spread of drug addiction in a high-risk community, as well as the influence of introducing treatment and rehabilitation of the addict, and how this can affect the whole community in terms of criminal acts and addiction patterns.

2) Conceptual Model: The model that we present here represents the criminal activities in an area and the hard drug (for example, cocaine and heroin) consumption using a 3-D Cell-DEVS and the influence that the addictions could provoke in increased criminal activities in an area. The model is based on a previous model proposed by Dabbaghian *et al.* [15]. We also consider incapacitation (treatment of drug users in hospitals), which tends to reduce drug usage and criminal activities, thus influencing the scenario in a positive way.

We incorporate, in our work, five basic types of actors/characters: susceptible person (SP), low-risk person (LRP), high-risk person (HRP), drug addict criminal person (DCP), and incapacitated person (IP).

1) *SP:* An individual who currently does not consume drugs but is vulnerable or likely to start taking drugs in the future.
2) *LRP:* An individual who consumes drugs occasionally but is not an addict.
3) *HRP:* An individual who consumes drugs and is addicted.
4) *DCP:* An individual who is addicted to drugs and commits a crime to support the drug habit.
5) *IP:* An individual who has been treated by doctors and has successfully completed treatment.

The Cell-DEVS models that we introduce here analyze the influence of drug pattern/influence, peer associations, drug usage patterns, and criminality dependence.

3) Formal Model Specifications and Simulation: As mentioned earlier, we will discuss three: 1) a basic scenario when someone starts consuming drugs; 2) an addiction scenario

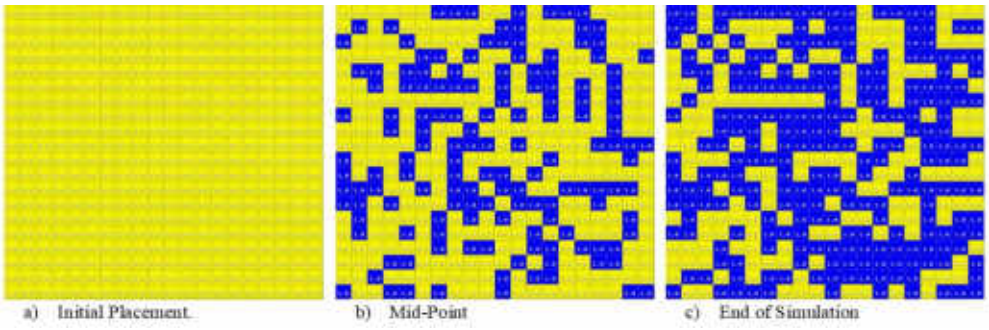

Fig. 13. Drug addiction basic model: transitioning from a healthy individual to drug consumer. (a) Initial placement. (b) Midpoint. (c) End of simulation.

when the drug consumer becomes addicted; and 3) a criminal scenario when an addict commits crimes to support their addiction. We also add the possibility of treating and rehabilitating the addict. Each cell contains an individual, and the cell value is the state of that individual.

a) Basic drug use scenario: In this scenario, we have two states: SP and LRP. When individuals start consuming drugs, they transition from SP to LRP based on the states of the neighbors. In this model, we use the Von Neumann neighborhood; we consider only the North (N), East (E), West (W), and South (S) neighbors. We assume here that a person does not get impacted by all individuals surrounding them. Instead, each individual is influenced by those whom they are in social contact with [15]. The formal description of the basic model is given as follows:

$$\begin{aligned}\text{Basic} &= \langle X_{\text{list}}, Y_{\text{list}}, I, X, Y, \eta, N, \{t_1, t_2\}, C, B, Z\rangle \\ X_{\text{list}} &= Y_{\text{list}} = \{\emptyset\}; \quad I = \{0, 1\}; \quad X = Y = \emptyset; \quad \eta = 5 \\ N &= \{(0, 0), (-1, 0), (0, -1), (0, 1), (1, 0)\} \\ t_1 &= t_2 = 20; \quad C = \{C_{\text{ij}}/i \in [0, 19] \wedge j \in [0, 19]\}\end{aligned}$$

and

$$B = \{\emptyset\}$$

where B is the border type, which is wrapped in our case, and the rules for this model are defined as follows:

```
rule: 1 200 {(0,0)=0 AND(0,-1)=1 AND (-1,0)=1
        AND(0,1)=1 AND(1,0)=1}
rule: 1 200 {((0,0)=0 OR(0,0)=1) AND
         normal(0.4,0.3) > 0.7}
rule: {(0,0)} 200 {t}.
```

The first rule forces the cell to switch to 1 after 200 ms if all the neighbors are 1. In other words, if the four neighbors with whom the individual is in social contact consume drugs, then the individual starts consuming. In the second rule, the individual can randomly start consuming drugs based on a normal distribution randomization function. In this rule, we consider a high-risk community where individuals could start consuming drugs even if not all their social peers are drug users. In any other case, the state of the individual remains unchanged. Fig. 13 shows a simulation scenario for this basic model.

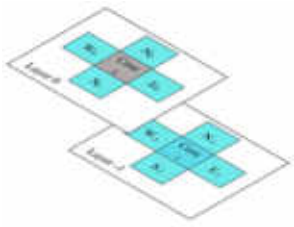

Fig. 14. Drug addiction 3-D model: two-layer neighborhood.

b) Addiction scenario: The second model implements an addiction situation, where a healthy person (SP) starts consuming drugs and changes state (LRP). Finally, that person becomes a drug addict (HRP). This model uses a 3-D Cell-DEVS. The first dimension of the neighborhood of the model is the same as the neighborhood of the previous model (Von Neumann). However, there is a second layer of the neighborhood where we consider five more neighbors in addition to the neighbors defined in the base layer. The neighborhood for this model is illustrated in Fig. 14.

The model defines two different zones with two different behavioral patterns. Each zone is represented in one plane in the model. Each zone represents one state variable.

The formal specifications of this model are

$$\begin{aligned}&\text{Drug\_Addiction} \\ &\quad = \langle X_{\text{list}}, Y_{\text{list}}, I, X, Y, \eta, N, \{t_1, t_{2,}t_3\}, C, B, Z\rangle \\ &X_{\text{list}} = Y_{\text{list}} = \{\emptyset\}; \quad I = \{0, 1, 2\}; \quad X = Y = \emptyset; \quad \eta = 10 \\ &N = \{(0, 0, 0), (-1, 0, 0), (0, -1, 0), (0, 1, 0), (1, 0, 0), \\ &\qquad (-1, 0, -1), (0, -1, -1), (0, 1, -1), (1, 0, -1), \\ &\qquad (0, 0, -1)\} \\ &t_1 = t_2 = 20; \quad t_3 = 2 \\ &C = \{C_{ijk}/i \in [0, 19] \wedge j \in [0, 19] \wedge k \in [0, 1]\}; \quad B = \{\emptyset\}\end{aligned}$$

and

$$\begin{aligned}Z = \{&(0, 0, 0)..(19, 19, 0) \\ &\text{use } \tau 1/\tau 1 = \text{NonDrugPerson-transition}; \\ &(0, 0, 1)..(19, 19, 1) \text{ use } \tau 2/\tau 2 = \text{DrugUser-transition}\}.\end{aligned}$$

Note that the border here is wrapped, but cells in different planes (zones) of the third dimension of the cellular model use different transition functions. The rules for this model are

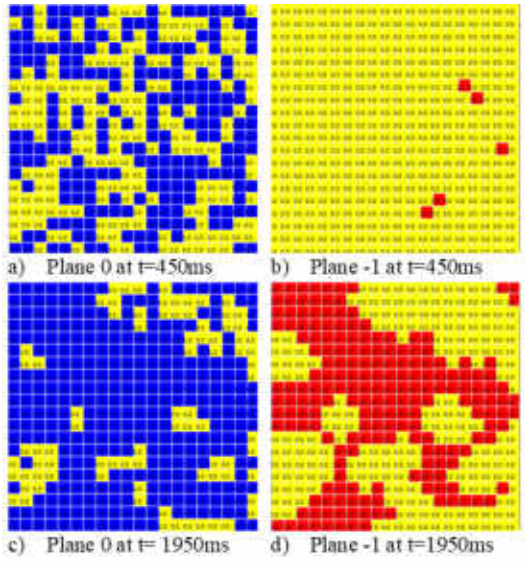

Fig. 15. Drug addiction model: transitioning from a healthy individual to drug consumer and then to a drug addict. (a) Plane 0 at $t = 450$ ms. (b) Plane -1 at $t = 450$ ms. (c) Plane 0 at $t = 1950$ ms. (d) Plane -1 at $t = 1950$ ms.

```
[NonDrugPerson-transition]
rule: 1 150 {(0,0,0)=0 AND (0,-1,0)=1 AND
      (-1,0,0)=1 AND (0,1,0)=1 AND (1,0,0)=1}
rule: 1 150 {(0,0,0)=0 AND normal(.4,.3) > .6}
rule: {(0,0,0)} 150 {t}
[DrugUser-transition]
rule: 2 150 {(0,0,0)=0 AND (0,0,-1)=1 AND
    (0,-1,-1)=1 AND (-1,0,-1)=1 AND (0,1,-1)=1
       AND (1,0,-1)=1}
rule: 2 150 {(0,0,0)=2}
rule: 0 150 {t}.
```

The first group of rules is the same as in the previous model. It describes the transition of a person from state 0 (SP) to 1 (LRP). The second group of rules (DrugUser-transition) represents the behavior of a cell (individual) whose neighbors (social influencers) are drug users. If the neighbors in the other layer belong to the LRP group, this individual's state becomes state 2 (HRP). In other words, this individual becomes a drug addict.

Fig. 15 shows the snapshots of two simulation scenarios of the drug addiction model. Fig. 15(a) shows the first layer of the model (SP and LRP individuals only). Fig. 15(b) shows the point in time when few HRP individuals start to appear in the second plane [see Fig. 15(b)]. Fig. 15(c) shows the state of the individuals in the first plane of the mode at the end of simulation where most of the population is in the LRP state, while a small percentage of the population is still in the SP state. Fig. 15(d) shows how this situation can progress due to social interaction with more drug consumers and individuals can become drug addicts (HRPs).

c) Crime as a result of addiction scenario: The last version of the model includes all states; the transition of an SP to an LRP, to an HRP, and then finally to a DCP. It also considers the IP individuals who are in the stage of being treated by doctors in the hospital to return to society as healthy members (see Section IV-C2).

We model the individual who is initially in an SP state in a 2-D plane, while the transitions to the four other possible states (LRP, HRP, DCP, and IP) are modeled by three more planes in the third dimension. There are four transition functions for calculating the state of each variable in the cell based on the variables of other cells within its neighborhood. The neighborhood is a 3-D Von Neumann neighborhood with four layers. The variables of LRP, HRP, and IP stages are supplied to the model to simulate complicated drug patterns and crime rates in an area. To generate random drug behavior, in the beginning, the variables in different cells are changed from 0 to 1 following normal probability distribution. The cells that are surrounded by persons who consume drugs are also susceptible and will likely start to consume drugs.

The crime drug model defines four zones for using different local computations as follows:

```
zone: LRP-transition{(0,0,0)..(19,19,0)}
zone: HRP-transition{(0,0,1)..(19,19,1)}
zone: crimedrug-transition{
            (0,0,2)..(19,19,2)}
zone: Incapacitation-transition{
            (0,0,3)..(19,19,3)}.
```

The computation rules for each zone are defined as follows:

```
[LRP-transition]
rule: 1 100 {(0,0,0)=0 AND (0,-1,0)=1 AND
     (-1,0,0)=1 AND (0,1,0)=1 AND (1,0,0)=1}
rule: 1 100 {(0,0,0)=0 AND normal(.4,.3) > .6}
rule: {(0,0,0)} 100 {t}

[HRP-transition]
rule: 2 100 {(0,0,0)=0 AND (0,0,-1)=1 AND
        (0,-1,-1)=1 AND (-1,0,-1)=1 AND
           (0,1,-1)=1 AND (1,0,-1)=1}
rule: 2 100 {(0,0,0)=2}
rule: 0 100 {t}

[crimedrug-transition]
rule: 3 100{(0,0,0)=0 AND normal(.4,.1) > .6}
rule: 3 100 {(0,0,0)=0 AND (0,0,-1)=2 AND
        ((0,-1,-1)=2 OR (-1,0,-1)=2 OR
            (0,1,-1)=2 OR (1,0,-1)=2)}
rule: 3 100 {(0,0,0)=3}
rule: 0 100 {t}

[Incapacitation-transition]
rule: 4 100 {(0,0,0)=0 AND (0,0,-1)=3 AND
        (0,0,-2)=2 AND normal(0.4,0.3) > 0.1}
rule: 4 100 {(0,0,0)=4}
rule: 0 100 {t}.
```

The set of rules listed under the [crime-drug] are responsible for switching the state of the cell to an HRP state if the core cell in layer -1 (relative to the neighborhood) is 2 (HRP), and at least, one of its four direct neighbors in the same plane (N_{-1}, S_{-1}, E_{-1}, and W_{-1}) is in the HRP state. The group of rules listed under [Incapacitation-Transition] handles transitioning the cell to the IP state. This happens when the value of the core cell in layer -1 is HRP, and the value of the core cell in layer -2 is IP using a random normal distribution generator.

Fig. 16 shows a snapshot of the simulation in the middle of the execution of the model for the four different planes

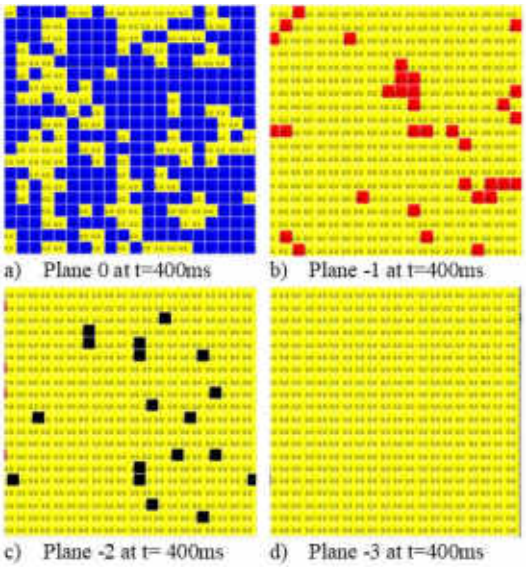

Fig. 16. Crime drug model: midpoint of simulation. (a) Plane 0 at $t = 400$ ms. (b) Plane -1 at $t = 400$ ms. (c) Plane -2 at $t = 400$ ms. (d) Plane -3 at $t = 400$ ms.

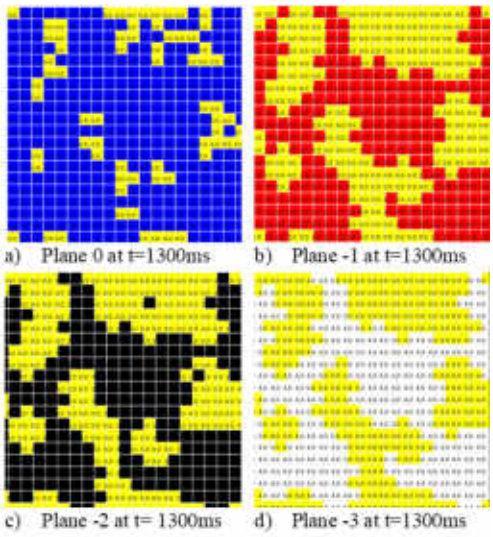

Fig. 17. Crime drug model: end of simulation. (a) Plane 0 at $t = 1300$ ms. (b) Plane -1 at $t = 1300$ ms. (c) Plane -2 at $t = 1300$ ms. (d) Plane -3 at $t = 1300$ ms.

representing the four transition functions: Fig. 16(a): $\tau 1$ (SP to LRP); Fig. 16(b): $\tau 2$ (LRP to HRP); Fig. 16(c): $\tau 3$ (HRP to DCP); and Fig. 16(d): $\tau 4$ (HRP to IP). However, at this time of the simulation, the last plane does not have any IPs yet.

Fig. 17 shows the end of the simulation where it is obvious from the comparison of Figs. 16(b) and 17(b) that more individuals are in the LR state (low-risk addicts) at the end of the simulation. In addition, comparing Figs. 16(c) and 17(c) illustrates that at the end of the simulation (see Fig. 17), many individuals transition into the HRP state (high risk). Finally, comparing the fourth plane of the simulation at a midpoint [see Fig. 16(d)] and at the end of the simulation (see Fig. 17) shows the transition of more than half of the population to the IP state (individuals that have been incapacitated).

4) Discussion: In this section, we modeled how in a high-risk community an individual state changes from being a normal society member, into being a drug consumer, addict, or crime committer for the sake of supporting a drug usage habit. In addition, we simulated the possibility of being rehabilitated to quit the drug addiction habit. We showed how Cell-DEVS formalism and the CD++ tool could be used to model and simulate the progress of such behavior or a similar phenomenon in a community. In this example, we made use of an additional feature of Cell-DEVS and CD++. We used a multidimensional Cell-DEVS model with the n-layered neighborhood. Cells in different layers use different transition functions, which allowed us to differentiate between the behavior of different individuals in the community (e.g., consumers and nonconsumers of drugs).

V. Conclusion

Modeling social changes has gained interest in the past, and this interest has even increased due to changes in the social environments (e.g., social networks and their effect on societies and human behavior). To model social systems efficiently, simulate changes in societies, and predict future evolutions without conducting long-term inaccurate observations, rigorous improved formal modeling methods are required. Here, we proposed Cell-DEVS as a formal method for modeling social systems. We described three main conceptual models of various kinds of social systems, formalized the models using Cell-DEVS, implemented the models using CD++, and showed the simulation results of the implemented models.

We explained the advantages of Cell-DEVS for modeling social systems in terms of performance, capabilities, and tool support. Through the example models that we presented, we showed how the unique features of Cell-DEVS (e.g., complex timing and zoning) overcome the shortcoming of other modeling formalism. In addition, we showed the availability of solid tools (CD++ toolkit), which allows researchers to implement the theoretical models formalized using Cell-DEVS successfully. Therefore, we propose Cell-DEVS and CD++ as a complete solution for modeling, implementing, simulating, and visualizing complex social systems.

Appendix
Advanced Avian Flu Macros

In following, we list the macros referred to in Section IV-A3.b

```
#BeginMacro(outer_2)
if ((-2,-2)=2,1,0) + if ((-2,-1)=2,1,0)
 + if ((-2,0)=2,1,0) + if ((-2,1)=2,1,0)
  + if ((-2,2)=2,1,0) + if ((-1,-2)=2,1,0)
   + if ((0,-2)=2,1,0) + if ((1,-2)=2,1,0)
    + if ((2,-2)=2,1,0) + if ((2,-1)=2,1,0)
     + if ((2,0)=2,1,0) + if ((2,1)=2,1,0)
       + if ((2,2)=2,1,0) + if ((1,2)=2,1,0)
      + if ((0,2)=2,1,0) + if ((-1,2)=2,1,0)
#EndMacro

#BeginMacro(inner_2)
if ((-1,-1)=2,1,0) + if ((0,-1)=2,1,0)
  + if ((1,-1)=2,1,0) + if ((1,0)=2,1,0)
   + if ((1,1)=2,1,0) + if ((0,1)=2,1,0)
    + if ((-1,1)=2,1,0) + if ((-1,0)=2,1,0)
#EndMacro
```

```
#BeginMacro(outer_1)
if ((-2,-2)=1,1,0) + if ((-2,-1)=1,1,0)
 + if ((-2,0)=1,1,0) + if ((-2,1)=1,1,0)
  + if ((-2,2)=1,1,0) + if ((-1,-2)=1,1,0)
   + if ((0,-2)=1,1,0) + if ((1,-2)=1,1,0)
    + if ((2,-2)=1,1,0) + if ((2,-1)=1,1,0)
     + if ((2,0)=1,1,0) + if ((2,1)=1,1,0)
      + if ((2,2)=1,1,0) + if ((1,2)=1,1,0)
       + if ((0,2)=1,1,0) + if ((-1,2)=1,1,0)
#EndMacro

#BeginMacro(inner_1)
if ((-1,-1)=1,1,0) + if ((0,-1)=1,1,0) +
  if ((1,-1)=1,1,0) + if ((1,0)=1,1,0) +
   if ((1,1)=1,1,0) + if ((0,1)=1,1,0) +
    if ((-1,1)=1,1,0) + if ((-1,0)=1,1,0)
#EndMacro

#BeginMacro(Immune) 0.1   #EndMacro
#BeginMacro(dead) 0.9     #EndMacro
#BeginMacro(Outer) uniform(0,0.05) #EndMacro
#BeginMacro(Inner) uniform(0,0.75) #EndMacro
#BeginMacro(Spreading) 0.2 #EndMacro
#BeginMacro(Vigilance) 0.6 #EndMacro.
```

ACKNOWLEDGMENT

The authors would like to thank S. Sahota, E. Sidhu, M. Singh, and A. Moughnie for developing various models presented in this article.

REFERENCES

[1] A. Al-Habashna and G. Wainer, "Modeling pedestrian behavior with cell-DEVS: Theory and applications," *SIMULATION*, vol. 92, no. 2, pp. 117–139, Feb. 2016.

[2] ARSLab. *ARSLab*. Accessed: Jul. 31, 2019. [Online]. Available: https://www.youtube.com/user/ARSLab

[3] ARSLab. (2019). *Avian Flu Spread with Boundary Cells*. Accessed: Jul. 31, 2019. [Online]. Available: https://www.youtube.com/watch?v=BgrTehJAeVI

[4] ARSLab. (2019). *Genet Colonies*. Accessed: Jul. 31, 2019. [Online]. Available: https://www.youtube.com/watch?v=DEf5nj-ngcc

[5] S. Athithan, V. P. Shukla, and S. R. Biradar, "Dynamic cellular automata based epidemic spread model for population in patches with movement," *J. Comput. Environ. Sci.*, vol. 2014, pp. 1–8, Feb. 2014.

[6] A. Behl, G. Wainer, and C. Ruiz-Martin, "Cell-DEVS: An approach to model the influence of social interactions in human behavior," in *Proc. SummerSim*, San Diego, CA, USA, 2018, Art. no. 18.

[7] D. Bombari, M. Schmid Mast, E. Canadas, and M. Bachmann, "Studying social interactions through immersive virtual environment technology: Virtues, pitfalls, and future challenges," *Frontiers Psychol.*, vol. 6, p. 869, Jun. 2015.

[8] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 3, pp. 7280–7287, May 2002.

[9] Y. Bouanan, G. Zacharewicz, and B. Vallespir, "Modelling of social influence with DEVS and cell-DEVS formalism," in *Proc. 4th Int. Defense Homeland Secur. Simulation Workshop (DHSS)*, 2014, p. 400.

[10] Y. Bouanan, M. B. El Alaoui, G. Zacharewicz, and B. Vallespir, "Using DEVS and CELL-DEVS for modelling of information impact on individuals in social network," *IFIP Adv. Inf. Commun. Technol.*, vol. 440, no. 3, pp. 409–416, 2014.

[11] Y. Bouanan, G. Zacharewicz, and B. Vallespir, "DEVS modelling and simulation of human social interaction and influence," *Eng. Appl. Artif. Intell.*, vol. 50, pp. 83–92, Apr. 2016.

[12] A. G. Bruzzone and M. Massei, "Polyfunctional intelligent operational virtual reality agent: PIOVRA final report," EDA, Genoa, Italy, Tech. Rep., 2007.

[13] R. Curiel, "Mathematical modeling of social systems," Ph.D. dissertation, Dept. Math., Univ. College London, London, U.K., 2018.

[14] V. Dabbaghian, P. Jackson, V. Spicer, and K. Wuschke, "A cellular automata model on residential migration in response to neighborhood social dynamics," *Math. Comput. Model.*, vol. 52, nos. 9–10, pp. 1752–1762, Nov. 2010.

[15] V. Dabbaghian, V. Spicer, S. K. Singh, P. Borwein, and P. Brantingham, "The social impact in a high-risk community: A cellular automata model," *J. Comput. Sci.*, vol. 2, no. 3, pp. 238–246, Aug. 2011.

[16] C. Dias and R. Lovreglio, "Calibrating cellular automaton models for pedestrians walking through corners," *Phys. Lett. A*, vol. 382, no. 19, pp. 1255–1261, May 2018.

[17] D. W. Dorsey and M. D. Coovert, "Mathematical modeling of decision making: A soft and fuzzy approach to capturing hard decisions," *Hum. Factors, J. Hum. Factors Ergonom. Soc.*, vol. 45, no. 1, pp. 117–135, Mar. 2003.

[18] W. Dzwinel, "A cellular automata model of population infected by periodic plague," in *Proc. Int. Conf. Cellular Automata (ACRI)*, Amsterdam, The Netherlands, 2004, pp. 464–473.

[19] P. E. Hosen, T. Djatna, and H. Khusun, "A cellular automata modeling for visualizing and predicting spreading patterns of dengue fever," *TELKOMNIKA (Telecommun. Comput. Electron. Control)*, vol. 14, no. 1, pp. 228–237, 2016.

[20] M. Forestier *et al.*, "Generating multidimensional social network to simulate the propagation of information," in *Proc. ASONAM*, Paris, France, 2015, pp. 1324–1331.

[21] Z. Fu, X. Zhan, L. Luo, A. Schadschneider, and J. Chen, "Modeling fatigue of ascending stair evacuation with modified fine discrete floor field cellular automata," *Phys. Lett. A*, vol. 383, no. 16, pp. 1897–1906, Jun. 2019.

[22] Z. Fu *et al.*, "A floor field cellular automaton for crowd evacuation considering different walking abilities," *Phys. A, Stat. Mech. Appl.*, vol. 420, pp. 294–303, Feb. 2015.

[23] D. G. Green, "Emergent behaviour in biological systems," *Complex Syst.-From Biol. Comput.*, vol. 1, pp. 24–35 1993.

[24] R.-Y. Guo, H.-J. Huang, and S. C. Wong, "Route choice in pedestrian evacuation under conditions of good and zero visibility: Experimental and simulation results," *Transp. Res. B, Methodol.*, vol. 46, no. 6, pp. 669–686, Jul. 2012.

[25] T. M. Gwizdałła, "Some properties of the floor field cellular automata evacuation model," *Phys. A, Stat. Mech. Appl.*, vol. 419, pp. 718–728, Feb. 2015.

[26] D. Helbing, L. Buzna, A. Johansson, and T. Werner, "Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions," *Transp. Sci.*, vol. 39, no. 1, pp. 1–24, Feb. 2005.

[27] D. Helbing, J. Keltsch, and P. Molnár, "Modelling the evolution of human trail systems," *Nature*, vol. 388, no. 6637, pp. 47–50, Jul. 1997.

[28] A. Hidano, G. Enticott, R. M. Christley, and M. C. Gates, "Modeling dynamic human behavioral changes in animal disease models: Challenges and opportunities for addressing bias," *Frontiers Vet. Sci.*, vol. 5, pp. 1–14, Jun. 2018.

[29] A. Holko, M. Mędrek, Z. Pastuszak, and K. Phusavat, "Epidemiological modeling with a population density map-based cellular automata simulation system," *Expert Syst. Appl.*, vol. 48, pp. 1–8, Apr. 2016.

[30] S. Hoya White, A. Martín del Rey, and G. Rodríguez Sánchez, "Using cellular automata to simulate epidemic diseases," *Appl. Math. Sci.*, vol. 3, nos. 17–20, pp. 959–968, 2009.

[31] O. Inghe, "Genet and ramet survivorship under different mortality regimes—A cellular automata model," *J. Theor. Biol.*, vol. 138, no. 2, pp. 257–270, May 1989.

[32] A. Kirchner and A. Schadschneider, "Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics," *Phys. A, Stat. Mech. Appl.*, vol. 312, nos. 1–2, pp. 260–276, Sep. 2002.

[33] R. Lawler and S. Jafer, "Egress modeling with cellular discrete event system," in *Proc. SoutheastCon*, Fort Lauderdale, FL, USA, Apr. 2015, pp. 1–8.

[34] X. Li, X. Liu, L. Xu, and Z. Zhang, "Spatial transmission of avian influenza (type H5) in birds," *Integrative Zool.*, vol. 4, no. 4, pp. 418–425, Dec. 2009.

[35] Z. Liu and C.-T. Fang, "A modeling study of human infections with avian influenza a H7N9 virus in mainland China," *Int. J. Infectious Diseases*, vol. 41, pp. 73–78, Dec. 2015.

[36] A. López and G. Wainer, "Improved cell-DEVS model definition in CD++," in *Proc. Int. Conf. Cellular Automata (ACRI)*, Amsterdam, The Netherlands, 2004, pp. 803–812.

[37] J. Ludewig, "Models in software engineering–an introduction," *Informatik-Forschung und Entwicklung*, vol. 18, nos. 3–4, pp. 105–112, Apr. 2004.

[38] L. Luo, Z. Fu, H. Cheng, and L. Yang, "Update schemes of multi-velocity floor field cellular automaton for pedestrian dynamics," *Phys. A, Stat. Mech. Appl.*, vol. 491, pp. 946–963, Feb. 2018.
[39] N. Margolus and T. Toffoli, "Cellular automata machines," *Complex Syst.*, vol. 1, pp. 967–993, Oct. 1987.
[40] P. Mavrodiev, C. J. Tessone, and F. Schweitzer, "Quantifying the effects of social influence," *Sci. Rep.*, vol. 3, no. 1, p. 1360, Dec. 2013.
[41] E. Mollona, "Computer simulation in social sciences," *J. Manag. Gov.*, vol. 12, no. 2, pp. 205–211, 2008.
[42] M. Muramatsu, T. Irie, and T. Nagatani, "Jamming transition in pedestrian counter flow," *Phys. A, Stat. Mech. Appl.*, vol. 267, nos. 3–4, pp. 487–498, May 1999.
[43] T. Nagatani and K.-I. Tainaka, "Cellular automaton for migration in ecosystem: Application of traffic model to a predator–prey system," *Phys. A, Stat. Mech. Appl.*, vol. 490, pp. 803–807, Jan. 2018.
[44] A. L. Osipian, "Corrupt organizations: Modeling educators' misconduct with cellular automata," *Comput. Math. Org. Theory*, vol. 19, no. 1, pp. 1–24, Mar. 2013.
[45] M. C. Paul, T. Vergne, P. Mulatti, T. Tiensin, and I. Iglesias, "Editorial: Epidemiology of avian influenza viruses," *Frontiers Veterinary Sci.*, vol. 6, pp. 1–2, May 2019.
[46] J. Pavon, C. Sansores, and J. J. G. Sanz, "Modelling and simulation of social systems with INGENIAS," *Int. J. Agent-Oriented Softw. Eng.*, vol. 2, no. 2, p. 196, 2008.
[47] Y.-C. Peng and C.-I. Chou, "Simulation of pedestrian flow through a 'T' intersection: A multi-floor field cellular automata approach," *Comput. Phys. Commun.*, vol. 182, no. 1, pp. 205–208, Jan. 2011.
[48] E. Pérez-Molina, R. Sliuzas, J. Flacke, and V. Jetten, "Developing a cellular automata model of urban growth to inform spatial policy for flood mitigation: A case study in Kampala, Uganda," *Comput., Environ. Urban Syst.*, vol. 65, pp. 53–65, Sep. 2017.
[49] A. W. Roddam, "Mathematical epidemiology of infectious diseases: Model building, analysis and interpretation," *Int. J. Epidemiol.*, vol. 30, no. 1, p. 186, Feb. 2001.
[50] F. S. Beltran, S. Herrando, V. Estreder, D. Ferreres, M.-A. Adell, and M. Ruiz-Soler, "Social simulation based on cellular automata: Modeling language shifts," in *Cellular Automata—Simplicity Behind Complexity*. London, U.K.: InTech, 2011, ch. 15, p. 323.
[51] M. Seck, C. Frydman, and N. Giambiasi, "Using DEVS for modeling and simulation of human behaviour," in *Proc. Int. Conf. AI, Simulation, Planning High Autonomy Syst. (AIS)*, Jeju-do, South Korea, 2005, pp. 692–698.
[52] M. Seck, N. Giambiasi, C. Frydman, and L. Baati, "DEVS for human behavior modeling in CGFs," *J. Defense Model. Simulation, Appl., Methodol., Technol.*, vol. 4, no. 3, pp. 196–228, Jul. 2007.
[53] H. Shang and G. Wainer, "A model of virus spreading using cell-DEVS," in *Proc. Int. Conf. Comput. Sci.*, Atlanta, GA, USA, 2005, pp. 373–377.
[54] G. C. Sirakoulis, I. Karafyllidis, and A. Thanailakis, "A cellular automaton model for the effects of population movement and vaccination on epidemic propagation," *Ecol. Model.*, vol. 133, no. 3, pp. 209–223, Sep. 2000.
[55] H. Situngkir, "Epidemiology through cellular automata," Bandung Fe Inst., Bandung, Indonesia, Tech. Rep. WPG2004, Mar. 2004.
[56] K. L. Soothill, B. Francis, and R. Fligelstone, "Patterns of offending behaviour: A new approach," Home Off. Res. Find., London, U.K., Tech. Rep. 1-6, 2002.
[57] J. E. Tanner, "The influence of clonality on demography: Patterns in expected longevity and survivorship," *Ecology*, vol. 82, no. 7, pp. 1971–1981, Jul. 2001.
[58] T. Toffoli and N. Margolus, *Cellular Automata Machines: A New Environment for Modeling*. London, U.K.: MIT Press, 1987.
[59] G. Wainer, "CD++: A toolkit to develop DEVS models," *Softw., Pract. Exper.*, vol. 32, no. 13, pp. 1261–1306, Nov. 2002.
[60] G. A. Wainer and N. Giambiasi, "Cell-DEVS/GDEVS for complex continuous systems," *SIMULATION*, vol. 81, no. 2, pp. 137–151, Feb. 2005.
[61] G. Wainer and R. Castro, "A survey on the application of the cell-DEVS formalism," *J. Cell. Autom.*, vol. 5, no. 6, pp. 509–524, 2010.
[62] G. Wainer, *Discrete-Event Modeling and Simulation: A Practitioner's Approach*, 1st ed. Boca Raton, FL, USA: CRC Press, 2009.
[63] F. Wang, D. Zeng, C. Academy, and K. M. Carley, "Social computing: From social informatics to social intelligence," *IEEE Intell. Syst.*, vol. 22, p. 2, pp. 79–83, Mar./Apr. 2007.
[64] WHO. (2012). *H5N1 Influenza.* Accessed: Jul. 17, 2019. [Online]. Available: http://www.who.int/influenza/human_animal_interface/avian_influenza/h5n1_research/faqs/en/
[65] S. Wolfram, "Cellular automata as models of complexity," *Nature*, vol. 311, no. 5985, pp. 419–424, Oct. 1984.
[66] S. Wolfram, "Statistical mechanics of cellular automata," *Rev. Mod. Phys.*, vol. 55, no. 3, pp. 601–644, 1983.
[67] W. F. Wolff, "An individual-oriented model of a wading bird nesting colony," *Ecol. Model.*, vol. 72, nos. 1–2, pp. 75–114, Mar. 1994.
[68] H. Wu, Z. Yuan, H. Li, and J. Tian, "Research on the effects of heterogeneity on pedestrian dynamics in walkway of subway station," *Discrete Dyn. Nature Soc.*, vol. 2016, pp. 1–10, Jul. 2016.
[69] E. S. Yu, "Social modeling and *i**," in *Conceptual Modeling: Foundations and Application* (Lecture Notes in Computer Science), vol. 5600. 2009, pp. 99–121.
[70] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of Modeling and Simulation*. New York, NY, USA: Academic. 2000.
[71] Y. Zheng, B. Jia, X.-G. Li, and N. Zhu, "Evacuation dynamics with fire spreading based on cellular automaton," *Phys. A, Stat. Mech. Appl.*, vol. 390, nos. 18–19, pp. 3147–3156, Sep. 2011.
[72] *Definition of Social System*. Accessed: Jul. 30, 2019. [Online]. Available: https://www.merriam-webster.com
[73] *CD++ Model Samples*. Accessed: Jul. 30, 2019. [Online]. Available: http://www.sce.carleton.ca/faculty/wainer/wbgraf/doku.php?id=model_samples:start
[74] *SICOMORES*. Accessed: Jul. 30, 2019. [Online]. Available: https://masasim.com/projects/sicomores/
[75] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz, "Simulation of pedestrian dynamics using a two-dimensional cellular automaton," *Phys. A, Stat. Mech. Appl.*, vol. 295, nos. 3–4, pp. 507–525, Jun. 2001.

Hoda Khalil (Member, IEEE) received the Ph.D. degree in electrical and computer engineering with a major in software engineering from Carleton University, Ottawa, ON, Canada, in 2018. Her Ph.D. thesis was focused on model-based testing.

From 2015 to 2018, she was a Research Assistant with the Software Quality Engineering Laboratory, Carleton University, where she is currently a Post-Doctoral Fellow with the Advanced Real-time Simulation Laboratory, researching modeling for sustainability, and an Instructor teaching software engineering (requirement elicitation and model-driven engineering). She has considerable experience in the software industry. Her work has been published in top peer-reviewed journals and conferences. Her current research is on new methods and tools for integrating real-time sensor data with occupancy information and integrating parametric data from buildings to construct formal models that can be used for both simulation and control.

Dr. Khalil received multiple scholarships and awards, including the W.B. McDermid Holbein Memorial Award for outstanding academic achievement.

Gabriel Wainer (Senior Member, IEEE) received the Ph.D. degree (Hons.) from UBA/Université d'Aix-Marseille III, Marseille, France, in 1998.

He held visiting positions at The University of Arizona, Tucson, AZ, USA, LSIS (CNRS), Marseille, France, Université Paul Cézanne, Marseille, the University of Nice, Nice, France, INRIA Sophia Antipolis, Biot, France, the Université de Bordeaux, Bordeaux, France, the Universidad Complutense de Madrid (UCM), Madrid, Spain, the Polytechnic University of Catalonia (UPC), Barcelona, Spain, the University of Buenos Aires, Buenos Aires, Argentina, the National University of Rosario, Rosario, Argentina, and so on. He is currently a Full Professor and an Associate Chair for Graduate Studies with Carleton University, Ottawa, ON, Canada, where he is also the Head of the Advanced Real-Time Simulation Laboratory, Centre for advanced Simulation and Visualization (V-Sim).

Dr. Wainer is also a Fellow of The Society for Modeling and Simulation International (SCS). He is also a member of the Board of Directors of the SCS. He is also a member of the Editorial Board of the IEEE COMPUTING IN SCIENCE & ENGINEERING, *Wireless Networks* (Elsevier), and *The Journal of Defense Modeling and Simulation* (SCS). He was a recipient of various awards, including the IBM Eclipse Innovation Award, the SCS Leadership Award, and various best paper awards. He also received the Carleton University's Research Achievement Awards in 2005 and 2014, the First Bernard P. Zeigler DEVS Modeling and Simulation Award, the SCS Outstanding Professional Award in 2011, the Carleton University's Mentorship Award in 2013, the SCS Distinguished Professional Award in 2013, the SCS Distinguished Service Award in 2015, the Nepean's Canada 150th Anniversary Medal in 2017, the ACM Recognition of Service Award in 2018, and the IEEE Outstanding Engineering Award (Ottawa Section) in 2019. He was the vice-president of several conferences and the vice-president of several publications. He is also one of the founders of the Symposium on Theory of Modeling and Simulation, SIMUTools, and the Symposium on Simulation of Architecture and Urban Design (SimAUD). He is also the Special Issues Editor of *Simulation*.