

Cell-DEVS for Social Phenomena Modelling

Hoda Khalil and Gabriel Wainer

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 Specification to implement and visualize models of social systems. We present several 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—social systems, modeling and simulation, cellular automata, Cell-DEVS, DEVS.

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, in particular, natural observations of social changes require performing social experiments over long periods, which make 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, 26, 27, 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 (section II.A).

Some of these limitations have been solved by combining CA and Discrete Event Specification (DEVS) models [62] (section II.B). The Cellular Discrete Event Specification (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 (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 paper, 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 through our YouTube channel [2].

II. BACKGROUND

A. Cellular Automata

The most common method for cellular computing is cellular automata (CA). CA are 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. This update happens in a synchronous manner in all the cells of the CA. Hence, the CA evolves in discrete timestamps [66]. Figure 1 is a simple sketch of cellular automata; the shaded area represents the neighborhood that affects the value of the core cell (starred) in the next timestamp.

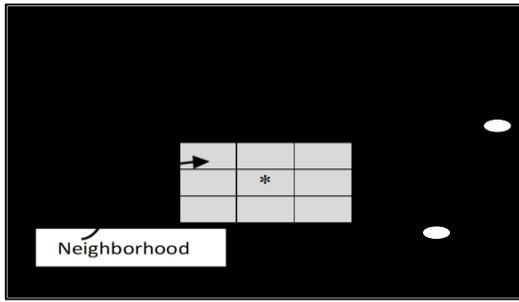


Figure 1 Cellular Automata

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

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

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 sufficiently complex 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, 15, 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, and (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, many tools were developed to address the different needs and uses of DEVS. A DEVS atomic model is formally described by:

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

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 can be 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.

Figure 2 is 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].

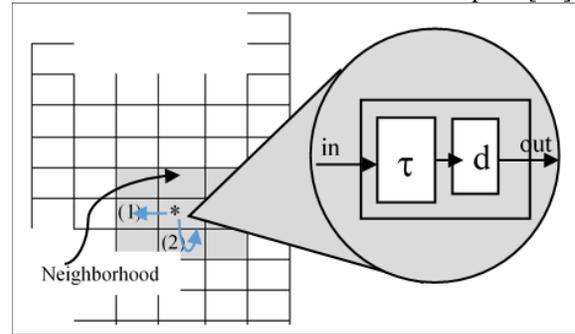


Figure 2 Two-Dimensional Cell-DEVS Informal Sketch.

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

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

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;
- 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 that 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]:

GCC= $\langle X_{\text{list}}, Y_{\text{list}}, I, X, Y, \eta, \{t_1, \dots, t_n\}, N, C, B, Z \rangle$
 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;
- $\eta \in N$ is the neighborhood size;
- $\{t_1, \dots, t_n\}$ 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 (e.g., borders can be wrapped). The translation function Z defines the internal and external coupling of cells in the model; it translates the outputs of the m^{th} output port in a cell C_{ij} into inputs to be fed to the m^{th} input port of another cell C_{kl} .

We can summarize the ways Cell-DEVS overcomes the drawbacks of classic CA (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, and (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 earlier (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, global warming and its effect on epidemics and social changes, etc.).

Different research has 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, 46]), regression models (e.g. [40]), CA (e.g. [54]), DEVS (e.g. [52]), etc. 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 the area of social modeling that explores the theoretical aspects in terms of finding mathematical models that are suitable for social phenomena. For example, R. Curiel introduces mathematical

models to consider different social challenges [13]. 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. 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 (like 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 our related work section into two subsections. The first subsection discusses how CA has been used in the literature to model social phenomena (section III.A). Then in the second subsection (section III.B), we review research work that uses Cell-DEVS and DEVS for social modeling.

A. CA for Social Modeling

CA are used extensively in the literature to model social behavior. For example, pedestrians' dynamics is one of many social aspects that have been modeled using CA [21, 29, 32, 42, 68]. Burstedde et al. represent pedestrians' behavior by using a two-dimensional 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 takes into account 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 some distance 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 cellular automata modeling concept to model pedestrians' behavior extensively [16, 21, 25, 38, 47]. For example, Zheng et al. propose a fire evacuation model based on cellular automata. 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. also use 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, Martín del Rey and Sánchez created a two-dimensional 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, Shukla, and Biradar propose a dynamic CA for both homogenous population 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. use 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 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 a few of many areas where CA is used to model different social phenomena. The research literature is rich with many other studies that utilize CA to model social behavior (e.g. [20, 27, 28, 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. uses DEVS to model human behavior for the sake of achieving a high level of realism for computer-generated forces (CGFs) [52]. Seck, Frydman, and Giambiasi propose 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 part of the poly-functional intelligent operational virtual reality agents (PIOVRA) project [12], Seck et al. add 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].

Another project that aims at simulating human social phenomena is the Simulation Constructive et Modélisation des effets des Opérations d’influence dans les réseaux Sociaux (SICOMPRES) [74]. The project’s objective is to model a social network of individuals, simulate the effect of information on them and then observe how the propagation of information affects this population of individuals [20]. As part of SICOMPRES, Bouanan et. al. present a Cell-DEVS model for how humans deal with information and how this information is propagated [10]. Similar to what we do in this work, the authors use CD++ to build their models. CD++ is a (M&S) tool for DEVS and Cell-DEVS [59]. A simple atomic DEVS model represents an individual with a set of attributes (e.g. gender, age, etc.). The individuals represented by the DEVS model combine to form the Cell-DEVS model representing the network of the population [10, 11].

Behl, Wainer, and Ruiz-Martin discuss 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, Zacharewicz, and Vallespir [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 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), 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 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 several advantages over CA. However, to the best of our knowledge, it has not been shown in the M&S research literature that DEVS and Cell-DEVS are adequate for performing different categories of social systems research. In addition, the tools needed to perform such M&S to facilitate social systems experimentations are not presented in this context.

The use of Cell-DEVS provides several advantages when compared to 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 can be seen as hierarchical schedulers of events that activate the corresponding submodels. The schedulers 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]. Several other existing formalisms can be expressed as DEVS models (including Petri Nets, FSM, State Charts, timed automata, etc.). Consequently, a modeler can express different properties in an adequate formalism and use DEVS hierarchical coupling as integration. In this way, one can take the advantage of current expertise of scientists in different domains. Modelers can describe individual components of cellular models using their own methods, which could result in enhanced model definition and would help to bridge the gap between traditional modeling techniques and cellular computing. Other advantages can be summarized as follows:

a) Our solution, compared to 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.

b) DEVS is a complete modeling and simulation 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.

c) 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 whole process for modeling and simulating social systems using a formal approach and automated means for the simulation process. The solution 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], we present in the following section several 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 (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 colonial organisms (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.B.4) we discuss how drug usage evolves in high-risk communities and can affect 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 from 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 we introduce in this section studies the different 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 (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 different types of neighbors, and add states to represent immunization and death as a possible consequence of the virus infection. We have to 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 km² and the other 1000 km². 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 that date. Outbreaks caused by the virus resulted in serious animal health crises worldwide and several economic crises. 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 serious 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, many 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 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 to model spatial epidemiology using CA. Models that are more sophisticated can be introduced by using and expanding the main concepts 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 1,225 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 some modifications. Using Cell-DEVS, we construct a two-dimensional 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 with direct contacts.

Each individual residing in a cell can be in one of the few states as described in the next section. The state of the individual (infected, susceptible, etc.) changes based on the states of the individuals in the neighborhood of cells 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 explained in section IV.A.3)b).

a) Basic Scenario

The first model shows the spreading of the virus at different probabilities. We use two states to model the spread of flu; susceptible (S) and infected (I). A susceptible cell is a cell that has not been infected but there is a certain probabilistic potential that this cell 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 two-dimensional Cell-DEVS space in the first out of three models, as defined in section II.B.

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

where

$$X_{list} = Y_{list} = \{\emptyset\}; I = \{0, 2\}; X = Y = \emptyset; \eta = 9;$$

$$t_1 = t_2 = 35; \text{ and}$$

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

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 of spreading the flu.

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 cell that is infected (having a value of 2), then the value of the core cell will change randomly to either 0 (susceptible) or 2 (infected). The second rule means that in all other cases the cell will keep its current value. Figure 3 shows the progress of a simulation scenario of the basic Avian flu model with a diffusion rate of 0.3, where the susceptible inhabitant cells are shown in light shade and the infected cells are dark. The simulation starts at

time 0 where there are two cells initially infected (Figure 3.a) and ends at time 4400 (Figure 3.d) when there are no more events to execute. Figure 3.b and Figure 3.c show the spread of the virus at times 100 and 1000 respectively.

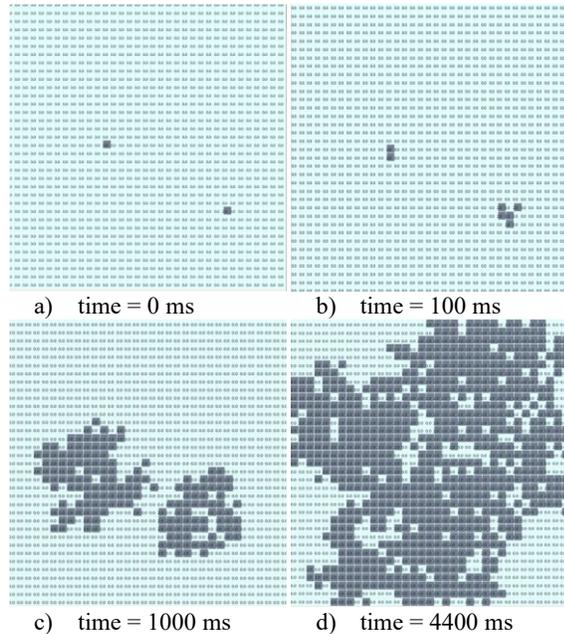


Figure 3. Avian Flu Basic Scenario: Snapshots of the Simulation at Different Times with Diffusion Rate 0.3.

a) *Introducing Boundaries*

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

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

$X_{list} = Y_{list} = \{\emptyset\}$; $I = \{0, 2, 5\}$; $X=Y = \emptyset$; $\eta = 9$; $t_1=t_2=35$; 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 are introduced to limit the spread of the disease. Figure 4 shows how the virus spreads with a 0.5 diffusion rate at different 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 (Figure 4.d).

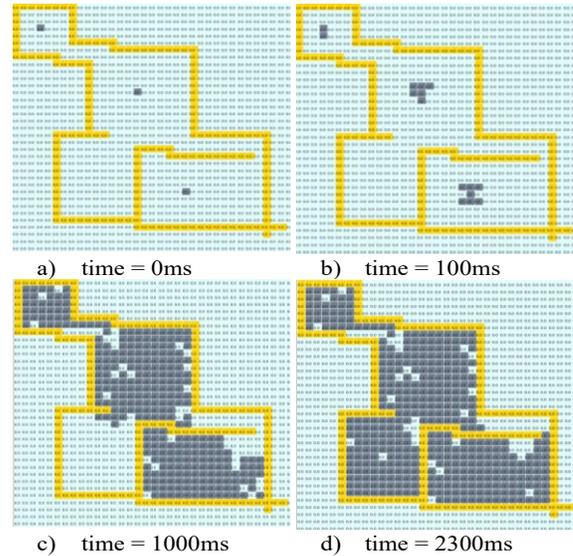


Figure 4 Avian Flu Boundary Scenario: Snapshots of the Simulation at Different Times with Diffusion Rate 0.5.

b) *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” which is next to the core cell, and the second type is the “remote neighbor” which is one cell farther (Figure 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 as compared to the remote ones.

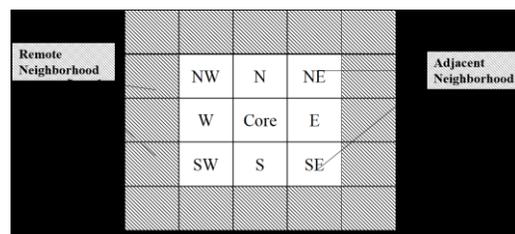


Figure 5 Avian Flu: 5x5 neighborhood

The following is the formal definition of the model. $Avian_Advanced = \langle X_{list}, Y_{list}, I, X, Y, \eta, \{t_1, t_2\}, N, C, B, Z \rangle$ where $X_{list} = Y_{list} = \{\emptyset\}$; $X = Y = \emptyset$; $I = \{-1, 0, 1, 2, 3\}$; $\eta = 25$; $t_1 = t_2 = 35$; and

$$N = \{(-1,0), (0,-1), (0,1), (1,0), (0,0), (-1,-1), (-1,1), (1,-1), (1,1), (-2,-2), (-2,-1), (-2,0), (-2,1), (-2,2), (-1,-2), (-1,2), (0,-2), (0,2), (1,-2), (1,2), (2,-2), (2,-1), (2,0), (2,1), (2,2)\}$$

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 rules above. The Appendix0 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 neighbor 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. Figure 6 shows the results when introducing the new neighborhood design.

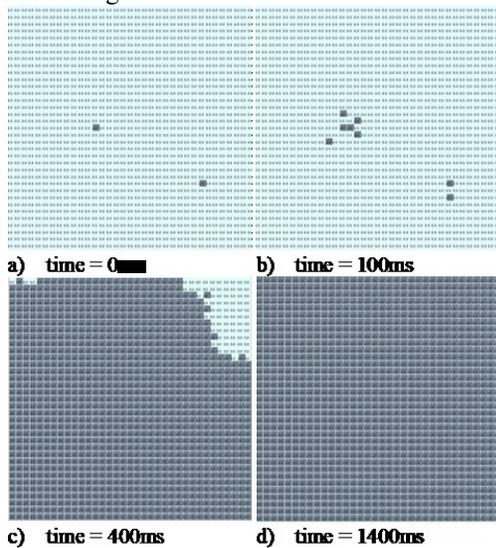


Figure 6. Advanced Avian Flu: Diffusion Rate Calculated Based on the Outer and Inner Neighbors.

As shown in the figure, when introducing remote neighbors, the disease spreads faster compared to the basic model of Figure 3 where only adjacent neighbors are considered.

In the next iteration of the model, we still use the neighborhood of Figure 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 based on probabilities calculated for immune cells (which is 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 they may spread no or less flu virus.

The rules for this model are defined below:

```
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 }
```

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

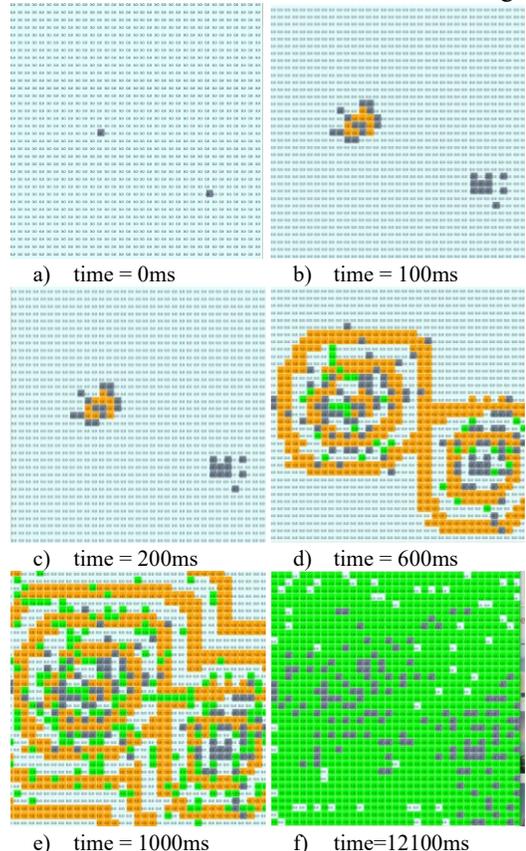


Figure 7 Avian Flu: 24 Cells Neighborhood: Immunization and Vigilance of the Inhabitants is considered after noticing symptoms.

In Figure 7.a, the simulation starts with two cells initially infected. Then, in Figure 7.b the virus spreads and some cells become questionable. In Figure 7.d, immunization starts to appear. This limits the spread of the disease as we can observe comparing Figure 7.f and Figure 6.d. While in Figure 6.d all the population is infected, Figure 7.f shows 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}
```

Figure 8 shows the simulation results when considering death due to a lack of immunization.

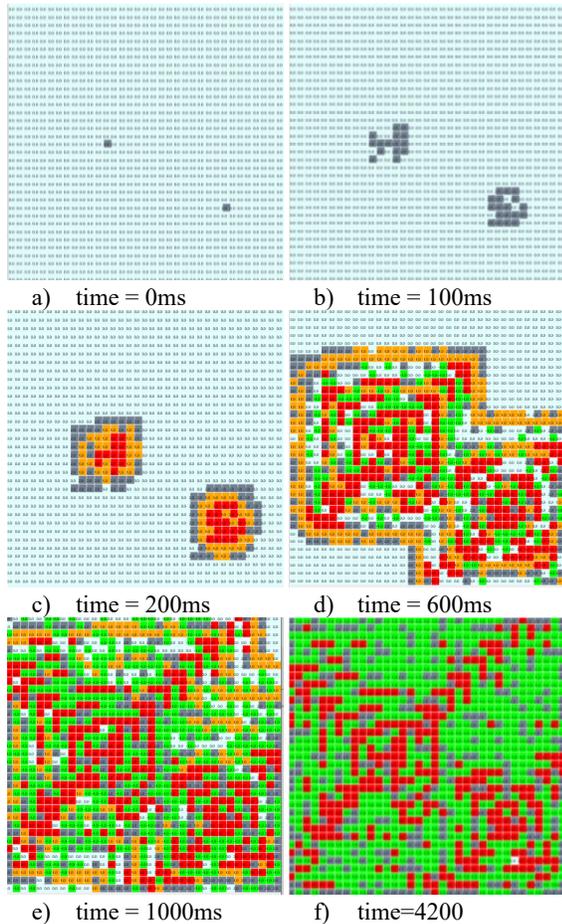


Figure 8 Avian Flu: Neighborhood of 24 cells: Inhabitants who are not Immunized Die Due to the Dangerous Flu

The simulation starts with two infected cells (Figure 8.a). Then, the disease starts to spread among the neighboring cells (Figure 8.b). The dead cells start to appear in Figure 8.c at the center of the infected population. The dead cells are surrounded by some questionable cells at this point in the simulation. Then in Figure 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 (Figure 8.e). At the end of the simulation, no more questionable cells are present. This is shown in Figure 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. Also, CD++ allowed for the implementation of zones (quarantine areas), outside of which the disease cannot spread. Furthermore, the possibility of using Cell-DEVS and CD++ to define and implement multiple states allowed for representing several stages of the disease (e.g. immune, susceptible and infected states). The proposed models are only a few of many other ways that can be used to simulate all 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 (10x10), then, a model with two types of ramets with a larger lattice (15x15), and finally, a complete model with a 60x60 lattice and 9 different ramets located at random.

1) Problem Definition

Clonal organisms are comprised of modules that are either semi 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 were introduced by Inghe in [31], where the authors presented a two-dimensional CA to analyze this problem. In this section, we build a two-dimensional 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 9 different genets that are spread randomly in a cell-space of 60 x 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 purpose of the model, survivorship will be compared in relation to the population. The model is represented by a cell-space that consists of 60 rows by 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.

The neighborhood of each cell includes 13 cells (including the core cell itself); the shape of the neighborhood is shown in Figure 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 Figure 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 where P'_i is given by:

$$P'_i = \left(\frac{P_i}{\sum_{i=1}^{12} P_i} \right) \left(1 - \prod_{i=1}^{12} (1 - P_i) \right)$$

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 9 genets, each consisting of one ramet, and then exposing them to the death rule. Subsequent iterations will then be carried out by applying the growth rule, followed by the death rule.

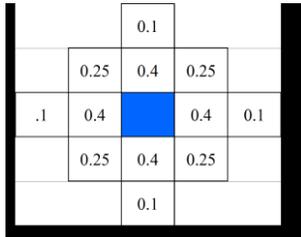


Figure 9. Genet and Ramet: 13 members' neighborhood

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:

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

where

$X_{list} = Y_{list} = \{\emptyset\}; I = \{0,1,2,3,4,5,6,7,8,9\}; X = Y = \emptyset;$

$\eta = 13 \quad t_1 = t_2 = 60;$

$N = \{(-2,0),(-1,-1),(-1,0),(-1,1),(0,-2),(0,-1),(0,0),(0,1),(0,2),$
 $(1,-1),(1,0),(1,1),(2,0)\};$

$C = \{C_{ij} / i \in [0,59] \wedge j \in [0,59]\};$

$B = \{\emptyset\}$, 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 by a neighbor. 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 below as an example:

rule : $\{(-2,0)\} 1000 \{(0,0)=0 \text{ and } (-2,0)>0 \text{ and}$

```
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 then the result is that the contents of cell (0, 0) will be replaced with the contents of cell (-2, 0). Rule thirteen 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)}
```

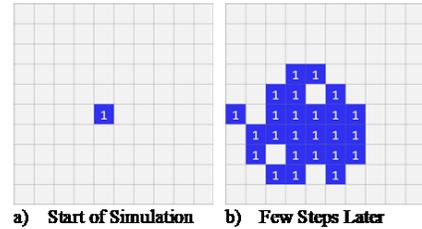


Figure 10. Single Ramet Simulation

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. Figure 10 shows the cell space at the start of the simulation (Figure 10.a) and a few steps afterward (Figure 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 15x15. Figure 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.



Figure 11. Two Ramet Colonies Simulation

Once these basic scenarios were completed, we executed various complex scenarios. Following, Figure 12 shows one of those scenarios. A simulation video showing how the colonies interact and evolve is available through the ARSLab YouTube

channel [4]. The figure shows the colonies near the start, middle, and end of the simulation. The first stage shows when the simulation starts with 9 ramets placed randomly in a 60×60 cell space (Figure 12.a). The second stage is the rapid growth and colonization of the whole cell space by the ramets until different colonies collide (Figure 12.b). The third stage occurs when 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 (Figure 12.c). Note how the colonies shift, as we can

see clearly by comparing the colonies from Figure 12.b and Figure 12.c. One of the colonies (close to the bottom right corner of Figure 12.b) can be seen dying out by comparing Figure 12.b and Figure 12.c. We can deduce 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.

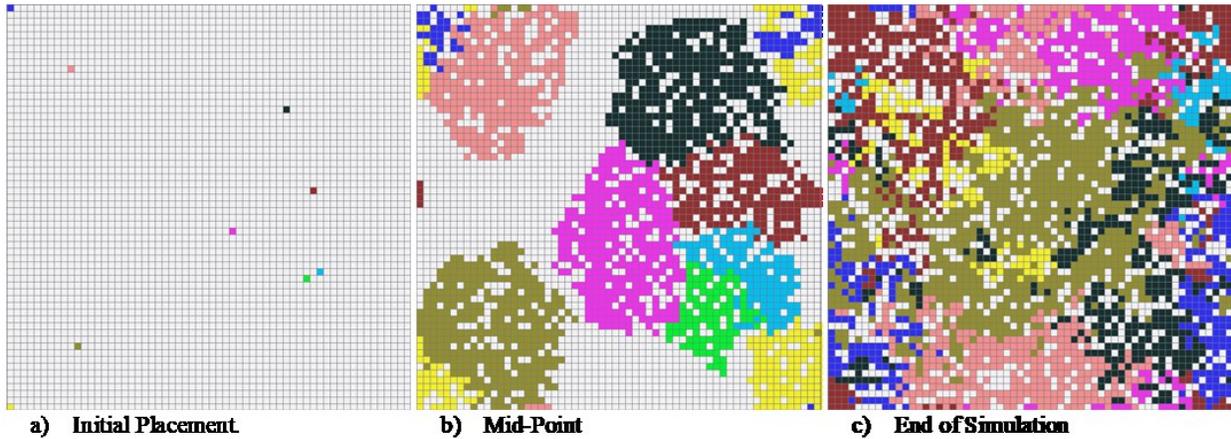


Figure 12 Ramet Colonies Simulation at Different Timestamps

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 connected boundaries of the represented area. Using CD++ allowed for defining different initial values of the cells to simulate the different types of ramets. In addition, we were able to implement the possibility of a cell colonizing its neighbor based on different probabilities using the different 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 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 we present here represents the criminal activities in an area and the hard drug (for example, cocaine and heroin) consumption using a three-dimensional 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 V. 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: it is 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 to them.
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 we introduce here analyze the influence of drug pattern/influence, peer associations, drug usage patterns, and criminality dependency.

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 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 the basic scenario, we only have two states: SP and LRP. When individuals start consuming drugs, they transition from SP to LRP. This transition is calculated 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 below:

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

$$X_{list} = Y_{list} = \{\emptyset\}; I = \{0, 1\}; X = Y = \emptyset; \eta = 5;$$

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

$$t_1 = t_2 = 20; C = \{C_{ij} / i \in [0, 19] \wedge j \in [0, 19]\}; \text{ and } B = \{\emptyset\}.$$

Where B is the border type, which is wrapped in our case, and the rules for this model are defined below

```
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 200ms if all the neighbors are in state 1. In other words, if the four neighbors with whom the individual is in social contact consume drugs, that 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 the fact that we are modeling 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. Figure 13 shows a simulation scenario for this basic model.

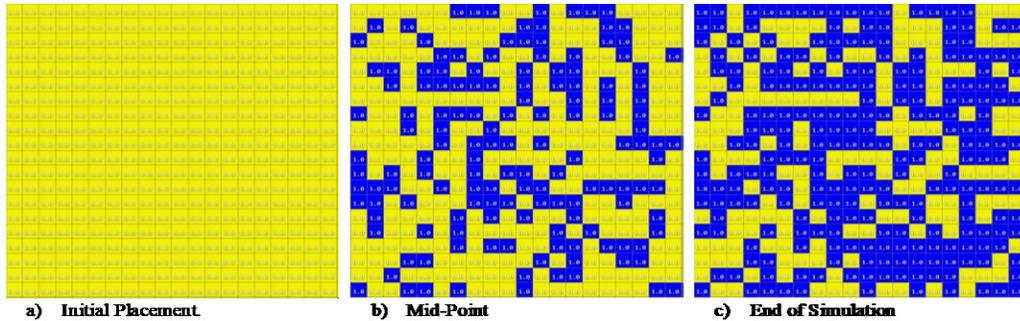


Figure 13. Drug Addiction Basic Model: Transitioning from Healthy Individual to Drug Consumer.

b) Addiction Scenario

The second model implements an addiction situation, where the healthy person in the cells (SP) starts consuming drugs and changes state (LRP). Finally, that person becomes a drug addict (HRP). This model is a 3-dimensional Cell-DEVS model. 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 Figure 14.

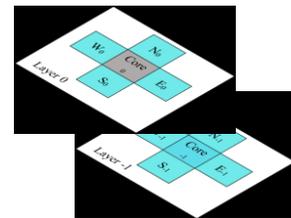


Figure 14 Drug Addiction 3D Model: Two Layers neighborhood

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:

$$Drug_Addiction = \langle X_{list}, Y_{list}, I, X, Y, \eta, N, \{t_1, t_2, t_3\}, C, B, Z \rangle$$

$$X_{list} = Y_{list} = \{\emptyset\}; I = \{0, 1, 2\}; X = Y = \emptyset; \eta = 10;$$

$N = \{(0, 0, 0), (-1, 0, 0), (0, -1, 0), (0, 1, 0), (1, 0, 0), (-1, 0, -1), (0, -1, -1), (0, 1, -1), (1, 0, -1), (0, 0, -1)\};$
 $t_1 = t_2 = 20; t_3 = 2;$
 $C = \{C_{ijk} / i \in [0, 19] \wedge j \in [0, 19] \wedge k \in [0, 1]\}; B = \{\emptyset\};$ and
 $Z = \{(0, 0, 0)..(19, 19, 0) \text{ use } \tau_1 / \tau_1 = \text{NonDrugPerson-transition}; (0, 0, 1)..(19, 19, 1) \text{ use } \tau_2 / \tau_2 = \text{DrugUser-transition}\}.$

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 defined below:

```

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

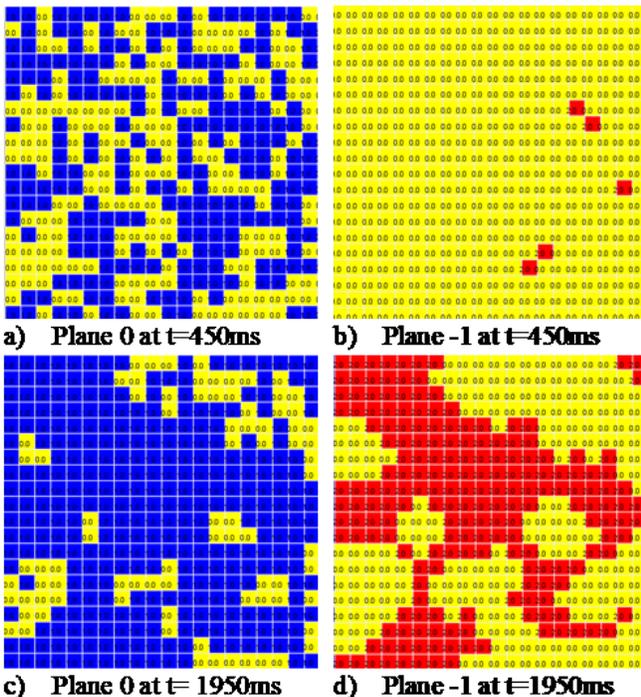


Figure 15 Drug Addiction Model: Transitioning from Healthy Individual to Drug Consumer, then to a drug addict.

Figure 15 shows the snapshots of two simulation scenarios of the drug addiction model. Figure 15.a shows the first layer of the model (SP and LRP individuals only). Figure 15.b shows the point in time when few HRP individuals start to appear in the second plane (Figure 15.b). Figure 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. Figure 15.d shows how this situation can progress due to social interaction with more drug consumers and many individuals can become drug addicts (HRPs).

c) Crime as a Result of Addiction Scenario

The final version of the model includes all states; the transition of an SP to an LRP, to an HRP, 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 (section IV.C.2).

We model the individual who is initially in an SP state in a two-dimensional 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 3D 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 some cells are changed from 0 to 1 following normal probability distribution. The cells which are surrounded by the 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: crimedrugs-transition{
            (0,0,2)..(19,19,2)}
zone: Incapacitation-transition{
            (0,0,3)..(19,19,3)}
    
```

The computation rules for each zone are defined below

```

[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] transition zone 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 ($N_{-1}, S_{-1}, E_{-1}, W_{-1}$) are 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.

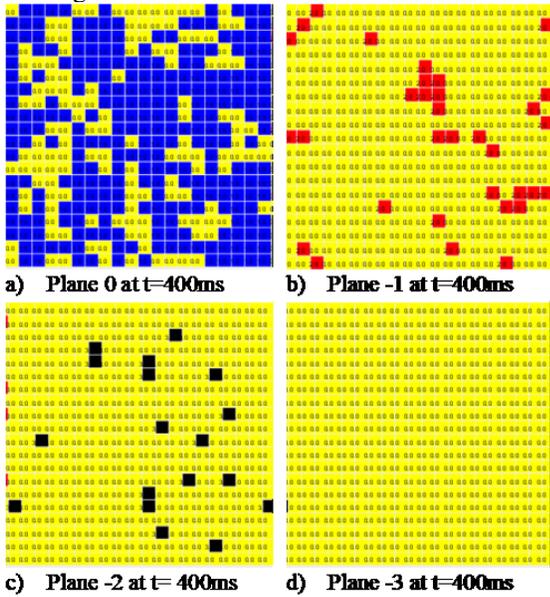


Figure 16 Crime Drug Model: Midpoint of Simulation.

Figure 16 shows a snapshot of the simulation at the middle of the execution of the model for the four different planes representing the four transition functions: Figure 16.a: τ_1 (SP to LRP), Figure 16.b: τ_2 (LRP to HRP), Figure 16.c: τ_3 (HRP to DCP), and Figure 16.d: τ_4 (HRP to IP). However, at this point in time of the simulation, the last plane does not have any IPs yet.

Figure 17 shows the end of the simulation where it is obvious from the comparison of Figure 16.b and Figure 17.b that more individuals are in the LR state (low-risk addicts) at the end of the simulation. In addition, comparing Figure 16.c and Figure 17.c illustrates that at the end of the simulation (Figure 17) many individuals transition into the HRP state (high risk). Finally, comparing the fourth plane of the simulation at a midpoint (Figure 16.d) and at the end of the simulation (Figure 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 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 non-consumers of drugs).

V. CONCLUSION

Modeling social changes has gained a lot of interest in the past and this interest has even increased recently due to changes in the social environments (e.g. social networks and their effect on societies and human behavior). Due to the need 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.

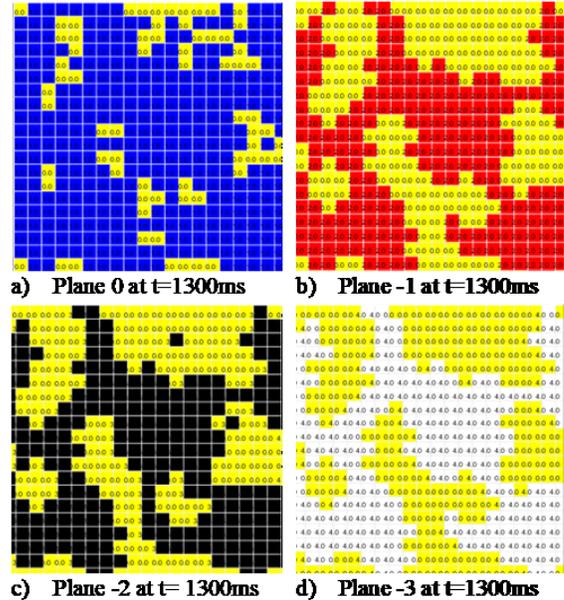


Figure 17 Crime Drug Model: End of Simulation.

In this paper, we proposed Cell-DEVS as a formal method for modeling social systems. We described three main conceptual models of different 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 in detail the advantages of Cell-DEVS over other simulation methods for modeling social systems in terms of performance, capabilities, and tool support (section III.B). Through the example models 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.

VI. ACKNOWLEDGMENTS

We thank S. Sahota, E. Sidhu, M. Singh, and A. Moughnic for developing various models presented in this paper.

VII. 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, 2016.
- [2] ARSLab, "ARSLab." [Online]. Available: <https://www.youtube.com/user/ARSLab>. Accessed on: 31-Jul-2019.
- [3] ARSLab, "Avian Flu Spread with Boundary Cells," 2019. [Online]. Available: <https://www.youtube.com/watch?v=BgrTehJAeVI>. Accessed on: 31-Jul-2019.
- [4] ARSLab, "Genet Colonies," 2019. [Online]. Available: <https://www.youtube.com/watch?v=Def5nj-ngcc>. Accessed on: 31-Jul-2019.
- [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.*, pp. 1–8, 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 *SummerSim*, San Diego, CA, USA, 2018.
- [7] D. Bombari, M. Schmid Mast, E. Canadas, and M. Bachmann, "Studying social interactions through immersive virtual environment technology: virtues, pitfalls, and future challenges," *Front. Psychol.*, vol. 6, 2015.
- [8] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proc. Natl. Acad. Sci.*, vol. 99, no. 3, pp. 7280–7287, 2002.
- [9] Y. Bouanan, G. Zacharewicz, and B. Vallespir, "Modelling of social influence with DEVS and Cell-DEVS formalism," in *4th International Defense and Homeland Security Simulation Workshop, DHSS*, 2014, pp. 43–48.
- [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 modeling and simulation of human social interaction and influence," *Eng. Appl. Artif. Intell.*, vol. 50, pp. 83–92, 2016.
- [12] A. G. Bruzzone and M. Massei, "Polyfunctional Intelligent Operational Virtual Reality Agent: PIOVRA Final Report," 2007.
- [13] R. Curiel, "Mathematical Modeling of Social Systems," Ph.D. dissertation, Dept. Math., Univ. College 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, no. 9–10, pp. 1752–1762, 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, 2011.
- [16] C. Dias and R. Lovreglio, "Calibrating cellular automaton models for pedestrians walking through corners," *Phys. Lett. Sect. A Gen. At. Solid State Phys.*, vol. 382, no. 19, pp. 1255–1261, 2018.
- [17] D. W. Dorsey and M. D. Coovert, "Mathematical Modeling of Decision Making: A Soft and Fuzzy Approach to Capturing Hard Decisions," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 45, no. 1, pp. 117–135, 2003.
- [18] W. Dzwiniel, "A cellular automata model of population infected by periodic plague," *ACRI*, Amsterdam, The Netherlands, 2004, pp. 464–473.
- [19] P. Eosina, T. Djatna, and H. Khusun, "A Cellular Automata modeling for visualizing and predicting spreading patterns of Dengue Fever," *Telkonnika*, vol. 14, no. 1, pp. 228–237, 2016.
- [20] M. Forestier et al., "Generating multidimensional social network to simulate the propagation of information," in *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. Sect. A Gen. At. Solid State Phys.*, vol. 383, no. 16, pp. 1897–1906, 2019.
- [22] Z. Fu et al., "A floor field cellular automaton for crowd evacuation considering different walking abilities," *Phys. A Stat. Mech. its Appl.*, vol. 420, pp. 294–303, 2015.
- [23] D. G. Green, "Emergent behaviour in biological systems," *Complex Systems -From Biology to Computation*, 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. Part B Methodol.*, vol. 46, no. 6, pp. 669–686, 2012.
- [25] T. M. Gwizdała, "Some properties of the floor field cellular automata evacuation model," *Phys. A Stat. Mech. its Appl.*, vol. 419, pp. 718–728, 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, 2005.
- [27] D. Helbing, J. Keltsch, and P. Molnár, "Modelling the evolution of human trail systems," *Nature*, vol. 388, no. 6637, pp. 47–50, 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," *Front. Vet. Sci.*, vol. 5, pp. 1–14, 2018.
- [29] A. Holko, M. Mędreń, Z. Pastuszek, and K. Phusavat, "Epidemiological modeling with a population density map-based cellular automata simulation system," *Expert Syst. Appl.*, vol. 48, pp. 1–8, 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, no. 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, 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. its Appl.*, vol. 312, no. 1–2, pp. 260–276, 2002.
- [33] R. Lawler and S. Jafer, "Egress modeling with cellular discrete event system," in *SoutheastCon*, Fort Lauderdale, FL, USA, 2015, pp. 1–8.
- [34] X. Li, X. Liu, L. Xu, and Z. Zhang, "Spatial transmission of avian influenza (type H5) in birds," *Integrative Zoology*, vol. 4, no. 4, pp. 418–425, 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. Infect. Dis.*, vol. 41, pp. 73–78, 2015.
- [36] A. López and G. Wainer, "Improved Cell-DEVS Model Definition in CD++," in *ACRI*, Amsterdam, The Netherlands, 2004, pp. 803–812.
- [37] J. Ludwig, "Models in software engineering - An introduction," *Inform. - Forsch. und Entwicklung*, vol. 18, no. 3–4, pp. 105–112, 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. its Appl.*, vol. 491, pp. 946–963, 2018.
- [39] N. Margolus and T. Toffoli, "Cellular Automata Machines," *Complex Syst.*, vol. 1, pp. 967–993, 1987.
- [40] P. Mavrodiev, C. J. Tessone, and F. Schweitzer, "Quantifying the effects of social influence," *Sci. Rep.*, vol. 3, 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. its Appl.*, vol. 267, no. 3, pp. 487–498, 1999.
- [43] T. Nagatani and K.-I. Tainaka, "Cellular automaton for migration in ecosystem: Application of traffic model to a predator-prey system," *Physica A: Statistical Mechanics and its Applications*, vol. 490, pp. 803–807, 2018.

[44] A. L. Osipian, "Corrupt organizations: Modeling educators' misconduct with cellular automata," *Comput. Math. Organ. Theory*, vol. 19, no. 1, pp. 1–24, 2013.

[45] M. C. Paul, T. Vergne, P. Mulatti, T. Tiensin, and I. Iglesias, "Editorial: Epidemiology of Avian Influenza Viruses," *Front. Vet. Sci.*, vol. 6, pp. 1–2, 2019.

[46] J. Pavon, C. Sansores, and J. J. G. Sanz, "Modelling and simulation of social systems with INGENIAS," *International Journal of Agent-Oriented Software Engineering*, 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, 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, 2017.

[49] A. W. Roddam, "Mathematical Epidemiology of Infectious Diseases: Model Building, Analysis and Interpretation," *Int. J. Epidemiol.*, vol. 30, no. 1, pp. 186–186, 2001.

[50] F. S., S. Herrando, V. Estreder, D. Ferreres, M.-A. Adell, and M. Ruiz-Soler, "Social Simulation Based on Cellular Automata: Modeling Language Shifts," *Cell. Autom. - Simplicity Behind Complex*, 2011.

[51] M. Seck, C. Frydman, and N. Giambiasi, "Using DEVS for Modeling and Simulation of Human Behaviour," in *AIS*, Jeju Island, Korea, pp. 692–698, 2005.

[52] M. Seck, N. Giambiasi, C. Frydman, and L. Baati, "DEVS for Human Behavior Modeling in CGFs," *J. Def. Model. Simul. Appl. Methodol. Technol.*, vol. 4, no. 3, pp. 196–228, 2007.

[53] H. Shang and G. Wainer, "A Model of Virus Spreading Using Cell-DEVS," in *Computational Science*, Atlanta, GA, USA, 2005, pp. 373–377.

[54] G. Sirakoulis, I. Karafyllidis, and A. Thanailakis, "A cellular automaton model for the effects of population movement and vaccination on epidemic propagation," *Ecological Modelling*, vol. 133, no. 3, pp. 209–223, 2000.

[55] H. Situngkir, "Epidemiology through cellular automata," Bandung Fe Institute, Tech. Rep., March 2004.

[56] K. L. Soothill, B. Francis, and R. Fligelstone, "Patterns of offending behaviour: a new approach," Home Off. Res. Find., Tech. Rep. pp. 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, 2001.

[58] T. Toffoli and N. Margolus, *Cellular automata machines : a new environment for modeling*. London, 1987.

[59] G. Wainer, "CD++: A toolkit to develop DEVS models," *Softw. - Pract. Exp.*, vol. 32, no. 13, pp. 1261–1306, 2002.

[60] G. Wainer and N. Giambiasi, "Cell-DEVS/GDEVs for complex continuous systems," *Simulation*, vol. 81, no. 2, pp. 137–151, 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. CRC Press, 2009.

[63] F. Wang, D. Zeng, C. Academy, and K. M. Carley, "Social Computing : From Social Informatics," *IEEE Intell. Syst.*, vol. 22, no. 2, pp. 79–83, 2007.

[64] WHO, "H5N1 influenza," Who, 2012. [Online]. Available: http://www.who.int/influenza/human_animal_interface/avian_influenza/h5n1_research/faqs/en/. [Accessed: 17-Jul-2019].

[65] S. Wolfram, "Cellular automata as models of complexity," *Nature*, vol. 311, no. 4, pp. 419–424, 1984.

[66] S. Wolfram, "Statistical mechanics of cellular automata," *Rev. Mod. Phys.*, vol. 55, no. 3, pp. 601–644, 1983.

[67] W. Wolff, "An individual-oriented model of a wading bird nesting colony," *Ecological Modelling*, vol. 72, no. 1-2, pp. 75–114, 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," *Discret. Dyn. Nat. Soc.*, vol. 2016, 2016.

[69] E. S. Yu, "Social modeling and i*," in *Conceptual Modeling: Foundations and Applications*, vol. 5600 LNCS, 2009, pp. 99–121.

[70] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. 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. its Appl.*, vol. 390, no. 18–19, pp. 3147–3156, 2011.

[72] "Definition of social system." [Online]. Available: <https://www.merriam-webster.com>. [Accessed: 30-Jul-2019].

[73] "CD++ Model Samples." [Online]. Available: http://www.sce.carleton.ca/faculty/wainer/wbgraf/doku.php?id=model_samples:start. [Accessed: 30-Jul-2019].

[74] "SICOMORES." [Online]. Available: <https://masasim.com/projects/sicomores/>.

APPENDIX - ADVANCED AVIAN FLU MACROS

Following, we list the macros referred to in section IV.A.3) of the document.

```
#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
```