

Asymmetric Cell-DEVS Models with the Cadmium Simulator

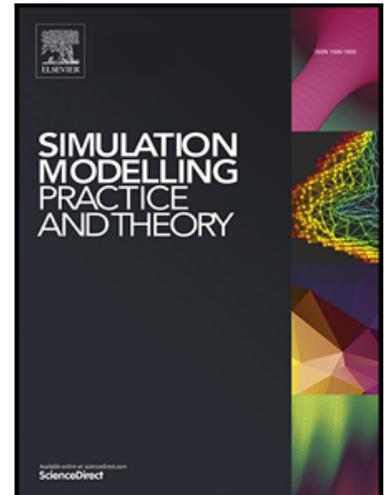
Román Cárdenas , Gabriel Wainer

PII: S1569-190X(22)00119-8
DOI: <https://doi.org/10.1016/j.simpat.2022.102649>
Reference: SIMPAT 102649

To appear in: *Simulation Modelling Practice and Theory*

Received date: 6 April 2022
Revised date: 18 August 2022
Accepted date: 22 August 2022

Please cite this article as: Román Cárdenas , Gabriel Wainer , Asymmetric Cell-DEVS Models with the Cadmium Simulator, *Simulation Modelling Practice and Theory* (2022), doi: <https://doi.org/10.1016/j.simpat.2022.102649>



This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Asymmetric Cell-DEVS Models with the Cadmium Simulator

Román Cárdenas^{ab} and Gabriel Wainer^b

^aLaboratorio de Sistemas Integrados, Universidad Politécnica de Madrid, Madrid, Spain; ^bSystems and Computer Engineering, Carleton University, Ottawa, Canada.

CORRESPONDING AUTHOR: Román Cárdenas. Email: r.cardenas@upm.es

ABSTRACT

Cellular models provide a natural approach for describing the behavior, dynamics, and structure of natural systems with spatial features. The classic Cell-DEVS formalism allows us to define discrete-event cellular models as a lattice of cells in which cell states are updated asynchronously over a continuous time base with explicit timing delays. However, it cannot define scenarios with irregular cell shapes. Here we present asymmetric Cell-DEVS, a formalism that supports the definition of cellular models with irregular topologies and complex relationships between cells, allowing us to model more realistic scenarios. We also include a prototype implementation of this formalism using the Cadmium simulator and illustrate the use of this formalism and tools by presenting different cellular models combined with geographical data to provide realistic results.

KEYWORDS: cellular models; discrete-event modeling and simulation; DEVS

1. Introduction

Numerous natural systems show great complexity in their behavior, dynamics, and structure, making it difficult to describe them mathematically (Wolfram, 1984). Cellular models provide simple abstractions of these systems, enabling the study of geographical and temporal changes on them (Janelle, 2005). Among the different approaches for implementing cellular models, Cellular Automata (CA) (Toffoli and Margolus, 1987) is one of the most popular methods. CA is a discrete-time formalism that is typically conceived as a lattice of cells. Each cell behaves as a Finite State Machine (FSM) and computes its next state according to its previous state and the previous state of nearby cells (i.e., the cell neighborhood). CA has been successfully used in multiple disciplines, such as chemistry (Gerhardt and Schuster, 1989), biology (Hatzikirou et al., 2012), or generative arts (Ashlock and Kreitzer, 2020).

In CA, cells state transitions are computed synchronously in a discrete-time base, constraining the precision of the simulations. Furthermore, multiple cells do not transition to new states in every step, compromising the computation efficiency of the simulated models. Additionally, the CA formalism is not easy to combine with other approaches (Wainer and Giambiasi, 2001). The Cell-DEVS formalism (Wainer, 2009) combines CA with the Parallel Discrete Event System Specification (PDEVS) (Zeigler et al., 2000) to overcome these issues. PDEVS is a revision of the original Discrete Event System Specification (DEVS). PDEVS is currently the prevailing DEVS variant. In the following, the use of the DEVS word implies PDEVS formal specifications.

Cell-DEVS allows us to define discrete-event cellular models in which cell states are updated asynchronously over a continuous time base, removing unnecessary computations and increasing the accuracy of the results. Furthermore, as Cell-DEVS is defined on top of the DEVS formalism, it is easy to integrate with models described with DEVS or other formalisms (Vangheluwe, 2000). The CD++ toolkit (Wainer, 2002) is a simple framework for implementing Cell-DEVS models. These models are executed asynchronously and provide a continuous time base for improving the accuracy of the simulations. CD++ has been successfully used in different research areas (e.g., building evacuation protocols (Wang et al., 2012), CO₂ diffusion models (Khalil et al., 2020), or epidemics (Cárdenas et al., 2020)).

Nonetheless, current tools for simulating Cell-DEVS models present some limitations. For example, they only support typical CA models that divide the scenario in a regular grid of cells. This approach hinders the integration with spatial systems with irregular topologies, for instance Geographic Information Systems (GISs), which follow a vectorial approach to describe spaces with irregular topologies (Chang, 2019). In these scenarios, alternative modeling approaches show better results. For instance, metapopulation models (Hanski, 1994) divide the system into multiple populations with different characteristics that interact with each other. The relationship between every pair of groups is defined separately. Furthermore, current Cell-DEVS tools integrate a low-level interface for describing aspects of the Cell-DEVS models (e.g., cells state transition functions or neighborhoods). This interface makes defining different scenarios for exploring the search space of the system under study cumbersome. Finally, current Cell-

DEVS tools cannot easily use third-party software for more advanced models (e.g., TensorFlow (Abadi et al., 2016) for representing the cells state transitions using complex machine learning models).

In this paper, we present the asymmetric Cell-DEVS formalism, a new flavor of Cell-DEVS. While the classic Cell-DEVS formalism has been used for scenarios with regular topologies, it relies heavily on isotropic geometries. Asymmetric Cell-DEVS is a generalization of classic Cell-DEVS that integrates concepts of CA and metapopulation models to define cellular models with complex topologies and dynamics. It uses a continuous time base and avoids unnecessary state transition computations. Moreover, as it relies on the DEVS formalism, we can easily combine asymmetric Cell-DEVS models with other DEVS models.

In addition, we present a new version of the Cadmium simulator (Belloli et al., 2019) that supports the implementation of asymmetric and classic Cell-DEVS scenarios. This new implementation is a flexible, header-only C++ (Stroustrup, 2013) library that allows users to implement custom cellular models with complex state transitions and timing behaviors. It also integrates a high-level interface for configuring Cell-DEVS scenarios using JavaScript Object Notation (JSON) files (Bray, 2017). This interface allows reusing a base Cell-DEVS model to execute different setups effortlessly. Furthermore, using JSON instead of any other notation format eases the integration with web-based tools and the readability of the scenario setup. It also simplifies the integration with GIS models that use the GeoJSON format (Butler et al., 2016). Cadmium eases the development and simulation of computational models based on the asymmetric Cell-DEVS formalism, as modelers do not have to worry about simulation details, and they can focus on implementing the logic of their conceptual models only.

The remainder of this paper is organized as follows. Section 2 presents related work. In Section 3, we present the asymmetric Cell-DEVS formalism, an extension of the classic Cell-DEVS formalism to support scenarios with irregular shapes. Section 4 describes the new version of the Cadmium simulator. We focus on its new Application Programming Interface (API) and the semantics supported by JSON configuration files for defining asymmetric and classic Cell-DEVS models. We illustrate how the asymmetric Cell-DEVS formalism and Cadmium can assist modelers in developing cellular models in Section 5. We also enumerate a set of successful use cases that follow this approach. Section 6 draws the main conclusions of this work.

2. Related Work

The modeling of natural systems aims to capture the behavior of biological processes to understand the relationships and dynamics of these systems (Wolfram, 1984). For example, models for pandemic spread may assist us with the definition of effective contention policies to reduce the impact of the disease. The COVID-19 outbreak has shown that modeling and simulation methodologies are valuable for understanding the disease and assessing the effect of different approaches to diminish its propagation speed (Adiga et al., 2020). Numerous proposals follow the approach presented by Kermack & McKendrick (1927), which classifies the population into three compartments: susceptible

(S), infected (I), and can recovered (R). This model is known as the SIR compartmental model. Figure 1 shows a schematic of the SIR model.

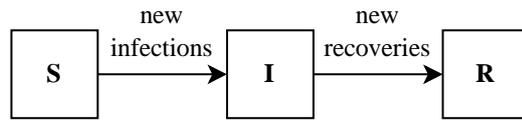


Figure 1. Schematic of the SIR compartmental model.

People that have never been infected remain in the S compartment. Infected individuals are assigned to the I group. People in this compartment can infect the susceptible population. Once infected individuals recover from the disease, they transition to the R compartment. The SIR model describes the dynamics between compartments with a set of Ordinary Differential Equations (ODEs). Different works extended this model with new compartments (e.g., deceased (D)) and new dynamics between compartments (e.g., immunity loss) (Tang et al., 2020). However, these models rely on the law of large numbers and cannot capture interactions between groups across both time and space. Several state-of-the-art works combine SIR-based compartmental models with cellular models to integrate the spatial aspects of the spread of the disease (Bin et al., 2019).

CA is one of the most popular approaches to define cellular models. CA is a discrete space-time formalism built as a lattice network of cells (Wolfram, 2002). In classic cellular models, cells are arranged uniformly in an n-dimensional lattice. A grid cellular scenario is characterized by the following parameters:

- Shape: n-tuple that defines the dimension of the scenario. Usually, $n \in \{1, 2, 3\}$.
- Origin cell: it indicates which cell is considered the first cell of the scenario. The other cells are numbered according to their position with respect to the origin cell.
- Wrap: wrapped scenarios connect the scenario boundaries with their opposite boundary. For instance, a wrapped 2-dimensional scenario connects cells in the left boundary with the cells in the right boundary, and uppermost cells are connected to lowermost cells. Wrapped scenarios can be thought of as n-cubes.

Figure 2 shows an example of a classic cellular model. In this example, the shape of the scenario is $3,4$ (i.e., a 3 by 4 2-dimensional cell grid). The origin cell is (1,1). Therefore, cell (2,1) is below the origin cell, and the cell (1,2) is to the right of cell (1,1). The shown scenario is wrapped. Thus, even though cell (1,4) corresponds to the upper-right corner of the scenario, cell (1,1) is to its right, and cell (3,4) is above it.

(3,4)	(3,1)	(3,2)	(3,3)	(3,4)	(3,1)
(1,4)	(1,1)	(1,2)	(1,3)	(1,4)	(1,1)
(2,4)	(2,1)	(2,2)	(2,3)	(2,4)	(2,1)
(3,4)	(3,1)	(3,2)	(3,3)	(3,4)	(3,1)
(1,4)	(1,1)	(1,2)	(1,3)	(1,4)	(1,1)

Figure 2. Example of classic cellular model.

CA are formally defined with the following quadruple:

$$CA = (S^n, N, f) \quad (1)$$

where:

- S^n is the n-dimensional working space. Usually, $n \in \{1, 2, 3\}$.
- N is the set of states. At time step t , the state of any cell corresponds to one of these states:

$$s_{i,j}^{(t)} \in S^n \quad (2)$$

- N is the neighborhood set. It contains relative positions of the neighboring cells concerning a given cell.
- f is the local computation function. It comprises a set of rules to obtain the new state of all the cells at every time step:

$$s_{i,j}^{(t+1)} = f(\{s_{i+j}^{(t)} \mid j \in N\}) \quad (3)$$

The neighborhood set in CA models is defined as a set of distance vectors. Every cell of the scenario can compute its neighboring cells by summing its location and each distance vector in the neighborhood set. For example, in a 2-dimensional scenario, if the neighborhood set is $N = \{(1,0), (0,0), (0,1), (-1,0), (0,-1)\}$, the neighboring cells of cell (i,j) are $(i+1,j)$, (i,j) , $(i,j+1)$, $(i-1,j)$, and $(i,j-1)$. This neighborhood set is common in the literature and is known as the von Neumann neighborhood of range 1, or simply the von Neumann neighborhood. For a given cell, the von Neumann neighborhood of range r is the set of cells in the scenario which Manhattan distance is less than or equal to r . Another popular cellular neighborhood is the Moore neighborhood. For a given cell, the Moore neighborhood of range r is formally described as the cells in the scenario which Chebyshev distance is less than or equal to r . Figure 3 represents these popular neighborhoods for 2-dimensional scenarios.

(i-1,j-1)	(i-1,j)	(i-1,j+1)
(i,j-1)	(i,j)	(i,j+1)
(i+1,j-1)	(i+1,j)	(i+1,j+1)

(a) von Neumann neighborhood ($r = 1$).

(i-1,j-1)	(i-1,j)	(i-1,j+1)
(i,j-1)	(i,j)	(i,j+1)
(i+1,j-1)	(i+1,j)	(i+1,j+1)

(b) Moore neighborhood ($r = 1$).

(i-2,j-2)	(i-2,j-1)	(i-2,j)	(i-2,j+1)	(i-2,j+2)
(i-1,j-2)	(i-1,j-1)	(i-1,j)	(i-1,j+1)	(i-1,j+2)
(i,j-2)	(i,j-1)	(i,j)	(i,j+1)	(i,j+2)
(i+1,j-2)	(i+1,j-1)	(i+1,j)	(i+1,j+1)	(i+1,j+2)
(i+2,j-2)	(i+2,j-1)	(i+2,j)	(i+2,j+1)	(i+2,j+2)

(c) von Neumann neighborhood ($r = 2$).

(i-2,j-2)	(i-2,j-1)	(i-2,j)	(i-2,j+1)	(i-2,j+2)
(i-1,j-2)	(i-1,j-1)	(i-1,j)	(i-1,j+1)	(i-1,j+2)
(i,j-2)	(i,j-1)	(i,j)	(i,j+1)	(i,j+2)
(i+1,j-2)	(i+1,j-1)	(i+1,j)	(i+1,j+1)	(i+1,j+2)
(i+2,j-2)	(i+2,j-1)	(i+2,j)	(i+2,j+1)	(i+2,j+2)

(d) Moore neighborhood ($r = 2$).

Figure 3. Schematic of von Neumann and Moore cellular neighborhoods.

Usually, the cell space uses a rectangular topology, in which each cell is represented as a rectangle. However, previous works proposed alternative lattice patterns like hexagons (Tariq and Kumaravel, 2016). Furthermore, Baetens & De Baets (2012) extend the CA formalism to support irregular tessellations. CA has been successfully used to describe the dynamics in natural systems, such as the course of braided rivers (Murray and Paola, 1994) or urban development (Ward et al., 2000).

In CA, the state of all the cells in the cell space are recomputed at every time step in synchronous fashion. However, usually, only a few cells update their state at a given time step. Thus, cells whose state remains unchanged add an unnecessary computation overhead. This overhead is greater for simulations that need shorter time steps for better time precision.

Different research have defined asynchronous CA (Ingerson & Buvel, 1984, Fatès, 2013, Dennunzio et al., 2017) to avoid these problems. However, these approaches still present some limitations inherent to the CA formalism (e.g., limited model complexity and closure to external events (Muzy et al., 2005)). Combining CA with the DEVS formalism has shown success in addressing these issues. DEVS (Zeigler et al., 2000) is a hierarchical and modular approach to describe discrete-event systems. The DEVS formalism describes systems at two levels: atomic models, which define the behavior of a system as transitions between states and response to external events; and

coupled models, which specify how subcomponents of the system interconnect. The formal definition of an atomic model is described as the following:

$$A = \langle X, Y, S, \tau_a, \tau_{int}, \tau_{ext}, \tau_{con}, \gamma \rangle, \quad (4)$$

where:

- X is the set of input events. Each element corresponds to a possible input event
- Y is the set of output events. Each element corresponds to a possible output event
- S is the states set. At any given time, the state of the atomic model is $s \in S$.
- $\tau_a: S \rightarrow \mathbb{R}_0^+$ is the time advance function. When the atomic model transitions to the state s , it will remain in this state for $\tau_a(s)$ time units or until the model receives one or more inputs.
- $\tau_{int}: S \rightarrow S$ is the internal transition function. After spending $\tau_a(s)$ time units in the state s without receiving any input event, the atomic model transitions to the state $s' = \tau_{int}(s)$.
- $\gamma: S \rightarrow Y$ is the output function. When the atomic is about to change its state due to an internal transition, the atomic model generates $\gamma(s) \subseteq Y$ output events. This function is triggered right before calling the internal transition function.
- $\tau_{ext}: S \times \mathbb{R}_0^+ \times X \rightarrow S$ is the external transition function. It is triggered when the atomic model receives a bag of inputs $x \subseteq X$ after e time units since the atomic model transitioned to its current state s (i.e., $0 \leq e < \tau_a(s)$). When triggered, $\tau_{ext}(s, e, x)$ determines the new state of the atomic model.
- $\tau_{con}: S \times \mathbb{R}_0^+ \times X \rightarrow S$ is the confluent transition function. This transition function decides the next state in cases of collision between external and internal events (i.e., $e = \tau_a(s)$). Typically, $\tau_{con}(s, \tau_a(s), x) = \tau_{ext}(\tau_{int}(s), 0, x)$.

The formal definition of a coupled model is described as follows:

$$M = \langle X, Y, C, EIC, IC, EOC \rangle, \quad (5)$$

where X is the set of inputs; Y is the set of outputs; C is the set of DEVS subcomponents; EIC is the external input coupling relation, from external inputs of M to component inputs of $c_i \in C$; IC is the internal coupling relation, from components outputs of $c_i \in C$ to component inputs of $c_j \in C$; and EOC is the external output coupling relation, from component outputs of $c_i \in C$ to external outputs of M .

In most cases, cellular models described with DEVS require expertise in programming, leading to multiple specific single-use programs which are difficult to reuse for related experiments (Wainer, 2006). The Cell-DEVS formalism (Wainer, 2009) proposes a simple and generic way of defining discrete-event cellular models with advanced timing behavior. Each cell is a continuous-time model in which state transitions

rest of the code remains intact. The `scenario` JSON object (lines 2 to 4) specifies that we want a 2-dimensional scenario of 3 by 3 cells. The coordinates of the origin cells are (-1, -1). The default cell configuration (lines 6 to 15) is like the previous example. However, we define a default neighborhood. In line 12, we define a von Neuman neighborhood of range 1 (see Figure 3a). The vicinity factor of all the cells in this neighborhood is set to 0.25. This vicinity factor matches with the shared border relationship for all the cells in this neighborhood except the cell itself. In line 13, we overwrite the von Neumann vicinity factor of one cell over itself to keep using the same definition of the vicinity factor.

Code 7. Example of a JSON configuration file for a classic Cell-DEVS scenario.

```

1 {
2   "scenario": {
3     "shape": [3, 3], "origin": [      - 1,  - 1] , " wrapped " : false
4   },
5   "cells": {
6     "default": {
7       "delay": "inertial",
8       "cell_type": "SIR",
9       "state": {"p": 100, "s": 1, "i": 0, "r": 0},
10      "config": {"rec": 0.2, "susc": 0.8, "vir": 0.4},
11      "neighborhood": [
12        {"type": "von_neumann", "vicinity": 0.25, "range": 1},
13        {"type": "relative", "vicinity": 1, "neighbors": [[0, 0]]}
14      ]
15    },
16    "infected": {
17      "state": {"s": 0.9, "i": 0.1},
18      "cell_map": [[1, 1]]
19    }
20  }
21 }

```

The infected JSON object (lines 16 to 20) defines an alternative cell configuration that sets the initial ratio of infected people to 0.1. The `cell_map` array indicates that this alternative configuration only affects cell (1, 1), while the rest of the scenario uses the default configuration.

5.5. Extending the SIR Model including Geographical Information Systems

In Cárdenas et al. (2021) we introduced an extended SIRDS compartmental model with irregular topologies and multiple infection phases with different recovery factors. In this model, individuals that die due to the disease transition to a new `D` compartment. Furthermore, recovered people may lose their immunity and become susceptible to the disease again. The population is also divided into age segments. We computed the vicinity factor as the length of the shared border divided by the perimeter of the region represented by the influenced cell.

We integrated GIS models with Cadmium for simulation based on real geographies. GIS software integrates physical location data with descriptive information (e.g., population or weather conditions). With GIS, we can analyze patterns and relationships in a geographical context. For example, we can monitor and forecast

changes to understand trends motivated by geographic characteristics. Figure 13 shows the workflow used to conduct these experiments.

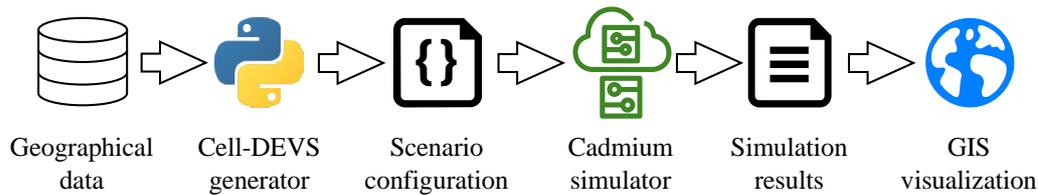


Figure 13. GIS integration for asymmetric Cell-DEVS models

First, a Cell-DEVS scenario generator reads geographical information from a GIS database. The generator adds a cell in the Cell-DEVS scenario for every physical location. Furthermore, it manipulates the descriptive characteristics of the physical location to define their initial state and the vicinity factor of neighboring cells. For the proposed SIRDS model, the generator uses population data to generate the initial state of the cells. Additionally, it computes relationships between physical locations to determine the cell neighborhoods and the vicinity factors to be applied. The resulting asymmetric Cell-DEVS scenario is stored in a JSON scenario configuration file. Cadmium executes the simulation of the scenario and outputs the simulation results. Then, simulation results are displayed on a map corresponding to the area under study using GIS tools and methodologies.

Davidson & Wainer (2021) presented an extended compartmental model that followed the proposed workflow for integrating GIS models with asymmetric Cell-DEVS scenarios. They adapted a GeoJSON file with geographical information about the Ontario province to create an asymmetric Cell-DEVS scenario in which each cell corresponds to a region of Ontario. For every pair of regions, a vicinity factor was computed as the shared border length between regions divided by the total border length of each region. The neighborhood of the cell is comprised of all the cells which vicinity factor is greater than zero. Simulation results were then adapted to a format compatible with the DEVS Web Viewer tool (St-Aubin et al., 2021). This software represents simulation results of asymmetric Cell-DEVS scenarios on a map, making it easier to understand and analyze simulation results. Figure 14 shows the simulation results of a pandemic scenario over the city of Ottawa.

For each district, the ratio of infected people is displayed in red tones. As simulation advances, all the regions turn darker. Darker cells correspond to districts with more cases of contagion. Additionally, the authors validated the model and calibrated all the configuration parameters (e.g., virulence and recovery rates) to reproduce the effect of the COVID-19 disease in Ontario. Simulation results were compared to the reports of the Government of Ontario in these regions from January 1st, 2020, to February 2nd, 2021 for validation and calibration. The structure of Cadmium allowed us to reuse the model for exploring multiple scenarios by just modifying the JSON configuration file, improving the experimentation phase.

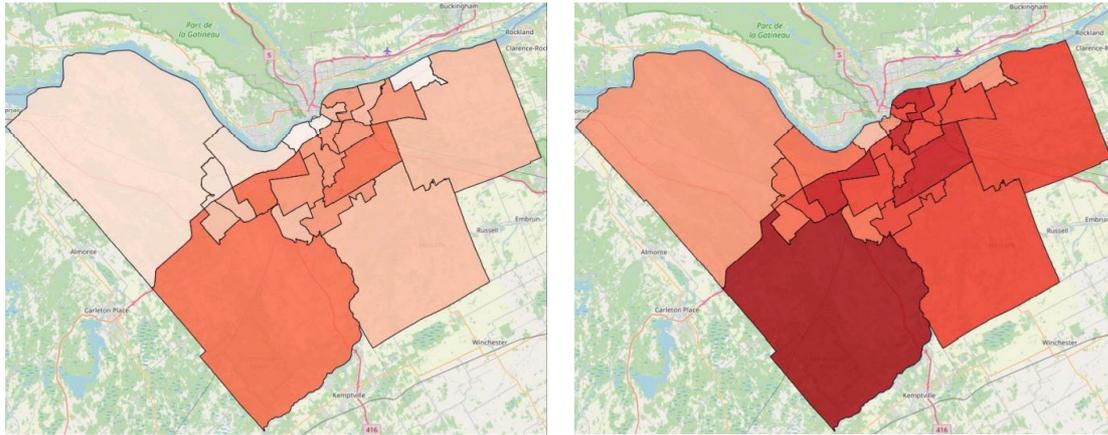


Figure 14. GIS visualization of simulation results for asymmetric Cell-DEVS models.

6. Conclusions

Cellular models provide a simple means of describing the behavior, dynamics, and structure of natural systems with spatial features. While the classic Cell-DEVS formalism presents advantages over other approaches (e.g., discrete-event nature, performance, or ease of integration with other simulation, visualization, and analysis tools), it is not able to effectively model scenarios with irregular topologies. This issue makes modeling scenarios based on real geographies a complex and inefficient task. Furthermore, if relations other than spatial are needed to be considered (i.e., closeness, friendship, political opinions, or other social aspects) classic Cell-DEVS modelers need to interpret the results of the cellular model accordingly, which is difficult to perceive.

We dealt with these issues by introducing the asymmetric Cell-DEVS formalism, which extends the classic approach by defining the relationships between cells using a graph-based approach. Asymmetric Cell-DEVS combines concepts of CA with metapopulation models to capture the behavior of more realistic natural systems using cellular models. Neighborhoods do not depend only on spatial associations but on an abstract concept called the vicinity factor. The vicinity factor of one cell over another can represent any relationship (e.g., spatial relationship, cultural kinship, or any other spatial relation) and thus allows modelers to define more complex and realistic cellular scenarios. Asymmetric Cell-DEVS is a generalization of classic Cell-DEVS. Therefore, we can still model grid-based cellular models. Its discrete-event nature avoids unnecessary computations by only triggering the state transition function of active cells. Additionally, as it relies on the DEVS formalism, we can easily combine asymmetric Cell-DEVS models with other DEVS models.

We also presented a new version of the Cadmium simulator, which supports the development of asymmetric Cell-DEVS models. It provides an API to develop classic and asymmetric Cell-DEVS scenarios, focusing on the definition cell behavior. This tool is being used for integrating asymmetric Cell-DEVS models with GIS tools, which allows to generate realistic Cell-DEVS scenarios based on actual geography data automatically, providing means for the analysis of simulation results using advanced visualizations.

References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., 2016. TensorFlow: A System for Large-Scale Machine Learning, in 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). USENIX Association, Savannah, GA, pp. 2656283.
- Adiga, A., Dubhashi, D., Lewis, B., Marathe, M., Venkatramanan, S., Vullikanti, A., 2020. Mathematical Models for COVID-19 Pandemic: A Comparative Analysis. *J. Indian Inst. Sci.* 100, 7936807. <https://doi.org/10.1007/s41745-020-00200-6>
- Ashlock, D., Kreitzer, M., 2020. Evolving Diverse Cellular Automata Based Level Maps, in: Ciancarini, P., Mazzara, M., Messina, A., Sillitti, A., Succi, G. (Eds.), *Proceedings of 6th International Conference in Software Engineering for Defence Applications, Advances in Intelligent Systems and Computing*. Springer International Publishing, Cham, pp. 10623. https://doi.org/10.1007/978-3-030-14687-0_2
- Baetens, J.M., De Baets, B., 2012. Cellular automata on irregular tessellations. *Dyn. Syst.* 27, 4116430. <https://doi.org/10.1080/14689367.2012.711300>
- Belloli, L., Vicino, D., Ruiz-Martin, C., Wainer, G., 2019. Building DEVS Models with the Cadmium Tool, in 2019 Winter Simulation Conference (WSC). Presented at the 2019 Winter Simulation Conference (WSC), IEEE, National Harbor, MD, USA, pp. 45659. <https://doi.org/10.1109/WSC40007.2019.9004917>
- Bin, S., Sun, G., Chen, C.-C., 2019. Spread of Infectious Disease Modeling and Analysis of Different Factors on Spread of Infectious Disease Based on Cellular Automata. *Int. J. Environ. Res. Public. Health* 16, 4683. <https://doi.org/10.3390/ijerph16234683>
- Bray, T., 2017. The JavaScript Object Notation (JSON) Data Interchange Format (RFC No. 8259). RFC Editor.
- Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S., Schaub, T., 2016. The GeoJSON Format (RFC No. 7946). RFC Editor.
- Cárdenas, R., Henares, K., Ruiz-Martín, C., Wainer, G., 2020. Cell-DEVS Models for the Spread of COVID-3; .
Bandini, S., Podlaski, K. (Eds.), *Cellular Automata. ACRI 2020. Lecture Notes*

- in Computer Science. Presented at the Cellular Automata for Research and
 Applications. *Journal of Cellular Automata and Discrete Complex Systems*, 16(4):6249.
https://doi.org/10.1007/978-3-030-69480-7_24
- Cárdenas, R., Martin, C.R., Wainer, G., Dobias, P., Rempel, M., 2021. Studying the
 Spread of Diseases Using Geographical Data and Irregular Topologies with
 Cell-DEVS, in 2021 Annual Modeling and Simulation Conference (ANNSIM).
 Presented at the 2021 Annual Modeling and Simulation Conference (ANNSIM),
 IEEE, Fairfax, VA, USA, pp. 1612.
<https://doi.org/10.23919/ANNSIM52504.2021.9552115>
- Cárdenas, R., Trabes, G., 2022. Cadmium 2: An object-oriented C++ M&S platform for
 the PDEVs formalism. [Online; Accessed on: June, 3, 2022] Available at
https://github.com/SimulationEverywhere/cadmium_v2.
- Chang, K.-T., 2019. Geographic Information System, in: International Encyclopedia of
 Geography. American Cancer Society, pp. 1610.
<https://doi.org/10.1002/9781118786352.wbieg0152.pub2>
- Davidson, G., Wainer, G., 2021. Studying COVID-19 Spread Using a Geography Based
 Cellular Model, in 2021 Winter Simulation Conference (WSC). Presented at the
 2021 Winter Simulation Conference (WSC), IEEE, Phoenix, AZ, USA, pp. 16
 12. <https://doi.org/10.1109/WSC52266.2021.9715520>
- Dennunzio, A., Formenti, E., Manzoni, L., Mauri, G., Porreca, A.E., 2017.
 Computational complexity of finite asynchronous cellular automata. *Theor.
 Comput. Sci.* 664, 1316143. <https://doi.org/10.1016/j.tcs.2015.12.003>
- Fatès, N., 2013. A Guided Tour of Asynchronous Cellular Automata, in: Kari, J.,
 Kutrib, M., Malcher, A. (Eds.), *Cellular Automata and Discrete Complex
 Systems*, Lecture Notes in Computer Science. Springer Berlin Heidelberg,
 Berlin, Heidelberg, pp. 15630. https://doi.org/10.1007/978-3-642-40867-0_2
- Gansner, E.R., Koutsofios, E., North, S.C., Vo, K.-P., 1993. A technique for drawing
 directed graphs. *IEEE Trans. Softw. Eng.* 19, 2146230.
<https://doi.org/10.1109/32.221135>
- Gerhardt, M., Schuster, H., 1989. A cellular automaton describing the formation of
 spatially ordered structures in chemical systems. *Phys. Nonlinear Phenom.* 36,
 2096221. [https://doi.org/10.1016/0167-2789\(89\)90081-X](https://doi.org/10.1016/0167-2789(89)90081-X)
- Hanski, I., 1994. A Practical Model of Metapopulation Dynamics. *J. Anim. Ecol.* 63,
 151. <https://doi.org/10.2307/5591>

- Hanski, I., Simberloff, D., 1997. The Metapopulation Approach, Its History, Conceptual Domain, and Application to Conservation, in: *Metapopulation Biology*. Elsevier, pp. 5626. <https://doi.org/10.1016/B978-012323445-2/50003-1>
- J. c\j kntqw"J 0'Dcucpc.'F 0'Uko qp.'O 0'Uej cmgt.'M0'F gwuej .'C0"423405I q"qt'I tqy ö<" the key to the emergence of invasion in tumour progression? *Math. Med. Biol.* 29, 49665. <https://doi.org/10.1093/imammb/dqq011>
- Heide-Jørgensen, M.P., Richard, P.R., Dietz, R., Laidre, K.L., 2013. A metapopulation model for Canadian and West Greenland narwhals: Narwhal metapopulation. *Anim. Conserv.* 16, 3316343. <https://doi.org/10.1111/acv.12000>
- Huang, J.-L., Koh, C.-K., Cauley, S.F., 2009. Logic and circuit simulation, in: *Electronic Design Automation*. Elsevier, pp. 4496512. <https://doi.org/10.1016/B978-0-12-374364-0.50015-1>
- Ingerson, T.E., Buvel, R.L., 1984. Structure in asynchronous cellular automata. *Phys. Nonlinear Phenom.* 10, 59668. [https://doi.org/10.1016/0167-2789\(84\)90249-5](https://doi.org/10.1016/0167-2789(84)90249-5)
- Janelle, D.G., 2005. Time6Space Modeling, in: *Encyclopedia of Social Measurement*. Elsevier, pp. 8516856. <https://doi.org/10.1016/B0-12-369398-5/00347-9>
- Kermack, W.O., McKendrick, A.G., 1927. A contribution to the mathematical theory of epidemics. *Proc. R. Soc. Lond. Ser. Contain. Pap. Math. Phys. Character* 115, 7006721. <https://doi.org/10.1098/rspa.1927.0118>
- Khalil, H., Wainer, G., Dumnigan, Z., 2020. Cell-DEVS Models for CO₂ Sensors Locations in Closed Spaces, in *2020 Winter Simulation Conference (WSC)*. Presented at the 2020 Winter Simulation Conference (WSC), IEEE, Orlando, FL, USA, pp. 6926703. <https://doi.org/10.1109/WSC48552.2020.9383937>
- Muneepeerakul, R., Weitz, J.S., Levin, S.A., Rinaldo, A., Rodriguez-Iturbe, I., 2007. A neutral metapopulation model of biodiversity in river networks. *J. Theor. Biol.* 245, 3516363. <https://doi.org/10.1016/j.jtbi.2006.10.005>
- Murray, A.B., Paola, C., 1994. A cellular model of braided rivers. *Nature* 371, 54657. <https://doi.org/10.1038/371054a0>
- Muzy, A., Innocenti, E., Aiello, A., Santucci, J.F., Wainer, G., 2005. Specification of discrete event models for fire spreading. *Simulation* 81, 1036117. <https://doi.org/10.1177/0037549705052230>
- Sonnenschein, M., Vogel, U., 2001. Asymmetric cellular automata for the modelling of ecological systems, in: *Sustainability in the Information Society*. Presented at the EnviroInfo.

- St-Aubin, B., Menard, J., Wainer, G., 2021. A Web Based Modeling and Simulation Environment to Support the DEVS Simulation Lifecycle, in 2021 Annual Modeling and Simulation Conference (ANNSIM). Presented at the 2021 Annual Modeling and Simulation Conference (ANNSIM), IEEE, Fairfax, VA, USA, pp. 1612. <https://doi.org/10.23919/ANNSIM52504.2021.9552123>
- Stroustrup, B., 2013. The C++ programming language, Fourth edition. ed. Addison-Wesley, Upper Saddle River, NJ.
- Vepi, N., 2013. A Web Based Modeling and Simulation Environment to Support the DEVS Simulation Lifecycle, in 2013 Annual Modeling and Simulation Conference (ANNSIM). Presented at the 2013 Annual Modeling and Simulation Conference (ANNSIM), IEEE, Fairfax, VA, USA, pp. 1612. <https://doi.org/10.23919/ANNSIM52504.2013.9552123>
- Tariq, J., Kumaravel, A., 2016. Construction of cellular automata over hexagonal and triangular tessellations for path planning of multi-robots, in 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC). Presented at the 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), IEEE, Chennai, pp. 166. <https://doi.org/10.1109/ICCIC.2016.7919686>
- Toffoli, T., Margolus, N., 1987. Cellular Automata Machines: A New Environment for Modeling, MIT Press series in scientific computation. MIT Press, Cambridge, Mass.
- Vangheluwe, H., 2000. DEVS as a common denominator for multi-formalism hybrid systems modelling, in: Proceedings of the IEEE International Symposium on Computer-Aided Control System Design. Presented at the IEEE International Symposium on Computer-Aided Control Systems Design, IEEE, Anchorage, AK, USA, pp. 129-134. <https://doi.org/10.1109/CACSD.2000.900199>
- Wainer, G., 2009. Discrete-event simulation: computational analysis, synthesis, and design of dynamic models series. CRC Press, Boca Raton.
- Wainer, G., 2006. Applying Cell-DEVS Methodology for Modeling the Environment. SIMULATION 82, 635-660. <https://doi.org/10.1177/0037549706073698>
- Wainer, G., 2002. CD++: a toolkit to develop DEVS models. Softw. Pract. Exp. 32, 1261-1306. <https://doi.org/10.1002/spe.482>
- Wainer, G., Giambiasi, N., 2001. Timed Cell-DEVS: Modelling and Simulation of Cell Spaces, in: Discrete Event Modeling and Simulation Technologies. pp. 187-214.

- Wang, S., Van Schyndel, M., Wainer, G., Rajus, V.S., Woodbury, R., 2012. DEVS-based Building Information Modeling and simulation for emergency evacuation, in: Proceedings - Winter Simulation Conference. IEEE.
<https://doi.org/10.1109/WSC.2012.6465087>
- Ward, D.P., Murray, A.T., Phinn, S.R., 2000. A stochastically constrained cellular model of urban growth. *Comput. Environ. Urban Syst.* 24, 5396558.
[https://doi.org/10.1016/S0198-9715\(00\)00008-9](https://doi.org/10.1016/S0198-9715(00)00008-9)
- Watts, D.J., Muhamad, R., Medina, D.C., Dodds, P.S., 2005. Multiscale, resurgent epidemics in a hierarchical metapopulation model. *Proc. Natl. Acad. Sci.* 102, 11157611162. <https://doi.org/10.1073/pnas.0501226102>
- Wolfram, S., 2002. *A new kind of science*. Wolfram Media, Champaign, IL.
- Wolfram, S., 1984. Cellular automata as models of complexity. *Nature* 311, 4196424.
<https://doi.org/10.1038/311419a0>
- Zeigler, B.P., Praehofer, H., Kim, T.G., 2000. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*, 2nd ed. ed. Academic Press, San Diego.
- Zhong, S., Huang, Q., Song, D., 2009. Simulation of the spread of infectious diseases in a geographical environment. *Sci. China Ser. Earth Sci.* 52, 5506561.
<https://doi.org/10.1007/s11430-009-0044-9>