Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

COMPOSABLE GEO-REFERENCED MULTI-RESOLUTION MULTI-AGENT CA-BASED DEVS, KIB, AND PDE MODELS

Hessam S. Sarjoughian
Chao Zhang

Arizona Center for Integrative Modeling & Simulation
School of Computing and Augmented Intelligence
699 S. Mill Avenue
Arizona State University
Tempe, AZ 85281, USA

ABSTRACT

Including geometry in non-spatial automata elevates their expressiveness. This provides the context required to understand many natural and built systems and facilitate their development. Indeed, the scope and types of questions asked by domain experts are continually rising due to the varied and intertwined structures and dynamics of hybrid systems. This is especially evident for heterogeneous models required to solve complex problems. They benefit from using different modeling formalisms and simulation frameworks. In this paper, an approach targeting the development of composable, heterogeneous, multi-resolution, spatiotemporal models formalized according to modular, cellular automata, and multi-agent models grounded in parallel DEVS, Modelica, and Geo-referenced Knowledge Interchange Broker methods is proposed. This approach is used to develop a co-simulation framework supported by the DEVS-Suite and OpenModelica simulators and the Functional Mock-up Interface. A multi-scale model for human breast cancer biology highlights the use of the developed approach and the co-simulation framework.

1 INTRODUCTION

Numerous dynamical systems that are built from simple to complex parts are inherently structurally and behaviorally multi-scale. Simulations of such systems involve combining separate models, each with its own temporal and spatial characteristics. Models have some parts that should be described at high resolutions while others are described at lower resolutions (Davis and Anderson 2003). Numerous research spanning several decades, as exemplified in Section 2 have shown the benefits of multi-resolution (aka multi-scale) models and simulations.

There are distinct reasons for needing multi-resolution models. Simulations of such multi-scale are necessary when different kinds of models created by individual experts or groups should be used and/or models necessarily differ in their spatiotemporal scales and also having varied structure and behavior complexities. Multi-scale simulations may also be needed to achieve computational efficiency while retaining acceptable accuracy (e.g.,

Considering simulation of human biology, different kinds of models that cut across a broad range of biological scales, from molecule to cell to tissue, are beneficial (Dada and Mendes 2011; Hunt et al. 2013). Molecular dynamics models can be defined to have discrete and continuous characteristics with varying macro and micro scales. Also, various modeling methods can be applied within the same phenomenon, e.g., particle diffusion is specified as multi-agent models at macro-scale or as partial differential equations at micro-scale. In-silico simulation can help better understand and replicate actual, in-vivo, and in-vitro biological processes.

978-1-6654-7661-4/22/$31.00 ©2022 IEEE

As an example of multi-scale computational (simulated) biology for studying human breast cancer, three kinds of models are developed (Chang et al. 2015). Cellular and molecular phenomena are modeled as discrete agent-based and differential equations. Cellular Automata are used to model cells and their migration. Ordinary Differential Equations are used to model molecules that bind to the surface receptors of cells. Partial Differential Equations can be used to model the environment that acts as a diffusion process to control the migration of cells.

Model composability can be considered from the standpoint of component-based, heterogeneous, and multi-resolution abstractions. The development of composite heterogeneous models is challenging (Simon 1962). Many approaches and frameworks use direct data read/write, object-orientation, and essentially software techniques to glue together simulation models. There exist many such works that lean more on simulation interoperability and less on model composability. For example, DEVS (Zeigler et al. 2018) and Modelica (Åström et al. 1998), as super-formalisms (Sarjoughian 2006), can be used to include other types of models. Continuous models can be embedded in DEVS models. Modelica can embed DEVS agent models using a cellular automaton (Sanz and Urquia 2021). These differential and algebraic equations (DAEs) and process-oriented discrete-event models are combined using function calls with continuous-time input/output ports and connectors. Such approaches are restrictive in view of composing heterogeneous model types with varying spatiotemporal scales and resolutions.

In this paper, the Knowledge Interaction Modeling approach (Sarjoughian 2006) is proposed for specifying the interactions among multi-agent Cellular Automata DEVS and PDE models that have distinct Geo-referenced spatiotemporal resolutions. This allows specifying the interactions among distinct modeling formalisms as standalone Geo-referenced KIB models. A modeling and simulation framework is developed using the DEVS-Suite and OpenModelica simulators integrated with the DEVS-FMI interface (ACIMS 2021).

2 COMPOSITIONS OF MULTI-RESOLUTION MODELS

A selection of component-based cellular-based modeling methods supported with frameworks and tools are briefly described. The emphasis is on the composition of simulatable models that can be chiefly categorized as strongly monomorphic or polymorphic. The roles of these methods in view of visual model development and rendering as well as visualizing their simulated dynamics are also considered. Detailed expositions for select model compositions can be found in their respective citations. The descriptions for select works are intended to highlight the unique capabilities proposed composition of heterogeneous models having independent temporal and spatial scales. This approach for multi-resolution heterogeneous modeling with its supporting co-simulation framework is founded on strong separation between model specification (model composability) and simulation execution (simulation interoperability).

2.1 Homogeneous Models

A variety of mathematical formalisms have been developed to characterize systems, for example, as continuous, discrete-time, and discrete event model types. It is not uncommon for a model type to be combined with other model types. Many simulation frameworks and tools are commonly used. For example, in a component-based modeling framework, a part of a discrete-event model can be abstracted to have continuous behavior even though the whole model is simulated as if all components have discrete-event dynamics. This idea, implemented in programming languages, can be found in many simulators. More generally, the Functional Mockup Interface and others have been developed to integrate different kinds of simulators, each supporting a particular kind of model structure and behavior.

Multi-agent models: These simulators are monolithic in the sense that each provides one approach to model specification and an execution protocol. They are generally efficient since the agents have direct access to read and write each other's state variables. A side effect is they are not strictly modular (i.e., each agent has input/output ports and communicates with others through message passing). They are considered "lightly" composable as other model types may be encapsulated in them. Each may provide built-in visualization for the specific kind of model that can be simulated. It is common to use external visualization tools for run-time or post simulations. An example is Multi-Agent Simulator Of Neighborhoods (MASON) (Luke et al. 2005). As an object-oriented simulator, it is extended with Physiomimetic Mechanism Modules defined as classes supported with required and provided interfaces (Hunt et al. 2013).

Cellular automata DEVS models: This is a cellular automata modeling based on the classic DEVS formalism, commonly in 2D and 3D spaces. The atomic model behavior is defined by a set of rules. The coupled model behavior is due to the atomic models and their relationships. Each cellular automata can have its own tessellation and dimensions. Cellular automata models can be coupled using their inputs and outputs to define other cellular automate models. The Cell-DEVS approach is implemented in CD++ simulator (Wainer 2017). Cell-DEVS models and non-spatial DEVS models can be composed. The approach is mostly monolithic from the DEVS model composition perspective since the overarching model and execution are governed by the Cell-DEVS. Such a simulator can be combined with simulators supporting other model types by encapsulating it in a Cell-DEVS atomic component. Simulation data can be stored in a log file and then animated in 2D space and plotted as linear time trajectories.

Composable cellular automata models: This approach to cellular automata composability is to distinctly specify cell-to-cell interactions within a cellular automata model from specifying cellular automata interacting with other models (Mayer and Sarjoughian 2009). The Discrete Time Multi-Component (DTM) is a special class of cellular automata where every cell share its state with its designated neighbors. A Composable Cellular Automata (CCA) is a standalone model with the ability to be composed with other CCAs or I/O modular models via external mapping functions. All cells within any CCA update simultaneously on a common time interval. Any cell can communicate its local state with any other cells in terms of (explicit or implicit) influencer-influencee relationships. The output from each cell will be mapped to the CCA and then mapped to the input of the external model. Conversely, the input from the external system will be mapped to the entire network and then to the individual cells. The CCA formalism intentionally excludes the specification for these external mapping functions (Mayer and Sarjoughian 2009). A realization of the CCA-DTM is developed for GRASS. GRASS GIS can be used to visualize stored simulation data.

2.2 Heterogeneous Models

Continuous-time and discrete-time models are combined to define a widely used class of composite models. Formalisms such as hybrid-automata (Alur 2015) define concise syntax and semantics for specifying mixed continuous and discrete models. Other modeling methods are essentially super-formalisms (Sarjoughian 2006) where one model type embeds others. Considering DEVS models, continuous models can be embedded in them (Zeigler, Muzy, and Kofman 2018). Modelica can embed DEVS agent models assigned to a cellular automaton (Sanz and Urquia 2021). Such approaches are restrictive in view of composing models, each having its own spatiotemporal resolution. Differential and algebraic equations (DAEs) and discrete-event models are combined using continuous-time input/output ports and connectors with function calls. Model composition approaches with different groundings can be placed in two categories as described below. There exist many such works that lean more on simulation interoperability and less on model composability.

Multiple single-scale cellular automata models: This approach to multi-scale modeling is based on Complex Automata (CxA) that supports defining a collection of single-scale sub-models (Hoekstra et al. 2010; Chopard et al. 2018). In principle, multiple sub-models can be combined to define larger models. All the sub-models are references that share a common spatial specification. The individual cells' spatial specifications define the spatial specification of the composed sub-models. Discrete changes to the states, using certain update rules, are allowed. The update rules are constrained by propagation, collision, and boundary crossing for propagation (sending data to neighbors), collision (updating the state of cells), and boundary conditions. Non-contiguous spatial sub-models use scale bridging. Algorithms for micro-macro conduits can be defined using sampling, splitting, homogenizing, projecting, and boundary condition alignments. Complex automata models can be visually developed as components with connectors, but it cannot be said if run-time simulation data visualization is supported since the tool is not available.

Cellular automata and agents models Cellular automata and component-based (aka agent-models) can be composed using the Knowledge Interchange Broker (KIB) approach (Sarjoughian 2006). An example is the composition of CCA-DTM models with parallel DEVS agents models. A KIB specifies the interactions between the CA and agent models in terms of their timing differences and concurrent bi-directional data transformations between models that have distinct variable types, structural relationships, and behavioral functions. The input, output, and state variables of model types have distinct characteristics. Variables can be piecewise-constant, piecewise-continuous, and discrete-event. Components form some uniform structure (e.g., multi-dimensional cellular automata), or components form some arbitrary (flat or hierarchical) structure. Geo-KIB supporting spatial conversions for two-dimensional geographic data maps with different regions is proposed *Geo-referenced KIB* (Boyd and Sarjoughian 2020). The Geo-KIB is used for modeling agent-based human-landscape dynamics. Interactions between models that different spatial-temporal data types and scales can be modeled without making any changes to the composed models. Cellular Automata models that have different spatiotemporal scales and map regions can be composed. The difference between the composed spatial models can be based on either map cell sizes or boundaries.

3 COMPOSABLE CELLULAR AUTOMATA MODELS

The rise of system complexity points to developing frameworks that can aid in developing composable models at multiple spatiotemporal resolutions. In particular, the concept of cellular automata has shown itself to be a cornerstone for numerous scientific and engineering pursuits. This is, in part, due to the inclusion of geometric abstraction that results in Cellular Automata being more expressive than other kinds of automata (von Neumann and Burks 1966). A cellular automaton has a set of individual cells that have a set of common properties and synchronously change their dynamics with respect to one another using a set of shared rules. The cells change their states under a uniform or non-uniform local neighborhood configuration. It is important to observe that the use of space and time together in modeling dynamic systems has a unique benefit in the same way Partial Differential Equations are more expressive than Ordinary Differential Equations. This view is obvious in the natural and built worlds and presents Cellular Automata to be well-suited for understanding and developing hybrid systems, among others (Mayer and Sarjoughian 2016).

Cellular Automata are commonly used for single abstraction levels of a system. For example, a CA specification for an airport roadway system can be defined to have seconds and centimeter scales. The movement of people in the airport terminals connected by the roadways can have minute and meter scales. Thus, an airport can be modeled at multiple abstraction levels (temporal and spatial resolutions), one for the movement of people and another for the movement of vehicles. It is useful to have such abstractions, on the one hand, be independent of one another and, on the other hand, to be composable.

3.1 Discrete-Time Multicomponent Modeling

The Cellular Automata modeling approach, as in many others, allows abstracting systems to be self-contained. That is, the CA cells have input, output, state, and functions, and as a system, they have some inner structure and behavior. The input and output for a Cellular Automaton are fundamentally those that belong to its cells that are connected among the parts (neighborhood). The behavior is defined as the collection of the state of all cells computed using a time-based function applied to each cell and its neighborhood.

As noted above, it is useful to have composable multiple Cellular Automata. A modeling formalism called CCA-DTM (Composable Cellular Automata Discrete-time Multicomponent) is defined for specifying CAs that have exogenous input and output modularity (Mayer and Sarjoughian 2009). The modeling formalism has a discrete-time characteristic and is based on multicomponent system specification (Zeigler, Muzy, and Kofman 2018). In a CCA-DTM, states of all cells are uniformly computed using a shared discrete time base. The behavior of every influenced cell can change due to its own state as well as the state of one or more influencing neighbors. The (partial) state of every influencing neighbor is its output. The neighbors' outputs constitute the influenced endogenous inputs of the CCA-DTM. The internal structure and behaviors of the cells and their relationships are separated from any exogenous I/O the CCA may have. As a standalone model, CCA-DTM can be composed of any other discrete-time I/O system specification. In other words, the CCA-DTM formalism is developed such that it can be coupled with I/O modular CA and non-CA discrete-time models.

3.2 Discrete Event System Specification Modeling

The composition of models that have strong input/output modularity (i.e., communication between different models is only via input and output ports) is fundamental for correct-by-construction of complex, time-based models (Sarjoughian). To achieve strong internal and external I/O modularity, the CCA-DTM is extended to support the Parallel DEVS formalism. This is achieved by replacing models of cells from discrete time to discrete event system specification and multicomponent to coupled models. The result is CCA-DEVS where DEVS atomic models are for cells that are geometrically arranged and coupled together to represent Composable Cellular Automata DEVS models (Zhang, Sarjoughian, and Seok 2020). In this approach, the cells and the CCA are all I/O modular, thus support strong means for composing CCA-DEVS models with any other model that has a well-defined time base and I/O modular.

The individual cells in a CCA-DEVS model have input and output ports connected to each other and also to other modules. As a result, hierarchical cellular automata models can be specified. Furthermore, the discrete-event cellular automaton and the discrete-event cells have a common continuous time base. The external I/O mapping functions for CCA-DEVS conform to the atomic and coupled DEVS system specifications. A formal specification of the CCA-DEVS can be found in (Zhang, Sarjoughian, and Seok 2020). An input/output modular discrete-event cellular automata framework based on the Discrete Event System Specification (DEVS) has been developed. This framework is built using the Cellular Automata DEVS (Zhang and Sarjoughian 2017) that is a part of the DEVS-Suite simulator (ACIMS 2021).

The differences between CCA-DTM and CCA-DEVS are important to be noted from the vantage point of the heterogeneous model composition. The CCA-DEVS formalism is modularized by using ports of I/O for communication, while the CCA-DTM system only has external ports for coupling with other models. The hierarchical structure of DEVS is organized from the atomic model to the coupled model, while the CCA-DTM is a standalone I/O system without any hierarchical structure. The cells in CCA-DTM do not have defined input and output ports with explicit connections. To compose discrete-time CCA with other CCA or non-CCA models, simulation synchronization for all the sub-systems (CCA or non-CCA) must be defined using a natural number time base. The discrete time base for CCA-DTM is defined as $T = \mathbb{N}$. Time is a set of finite time intervals $\{h_0, \ldots, h_k\}$) where $h_i \in \{(\mathbb{N} \cup \{0\}) - \{\infty\}\}, i \in \{0, \ldots, k\}$. The CCA-DEVS uses continuous time base is defined $T = R^+_{0,\infty}$. This time base allows composing I/O modular cellular automata and other types of automata to have mixed continuous or discrete event time bases.

3.2.1 Multi-agent Modeling

Agent-based and Cellular Automata models, from a basic perspective, have a common basis in terms of creating whole models from parts. The parts in an agent-based model (ABM) may not necessarily have spatial structures (i.e., agents do not have to have any spatial structures). In contrast, cellular automata models have regularly formed spatial structures. Each agent in ABM and each cell in CA is an independent entity. These entities as a collection can have some emergent behaviors.

Movement is a key feature for some cellular automata models. Individual and collective movements of agents can lead to behaviors with specific patterns for the whole model. Cells can be modeled as agents moving from one grid cell to another. For example, one agent moves towards the high-density regions of a PDE model while another agent moves randomly.

The CCA-DEVS can be used to model agents that move (Zhang and Sarjoughian 2017). It is advantageous to use CCA-DEVS due to being discrete – composability makes agents with agents and agents with the environment separable. The discrete event paradigm allows agents to interact via time-based events.

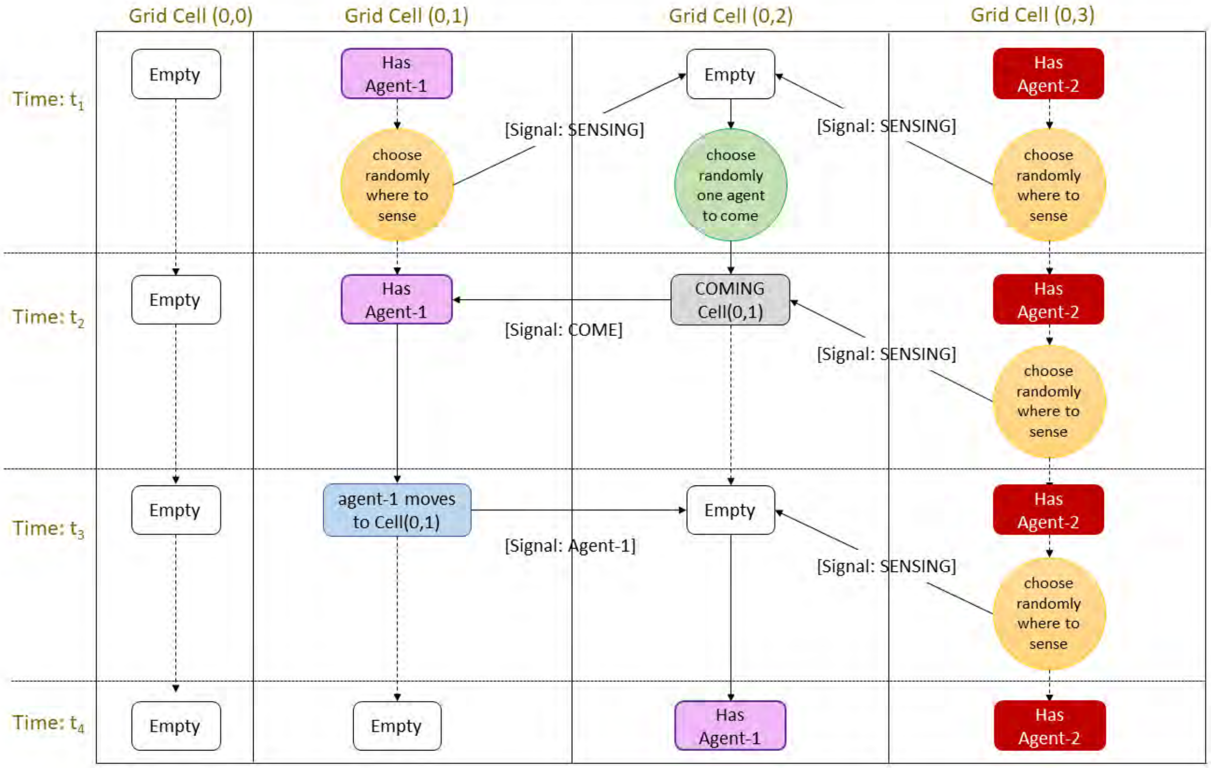


Figure 1: Illustration of the CCA-DEVS multi-agent movement algorithm.

Agent movement in discretized cellular automata space is represented as it changes its grid cell position to its neighbor position. The neighbor cell changes its state to have an agent instead of being empty. Any grid cell can be initialized as `Has Agent-i` or `Empty`. If the cell has the property of `Has Agent-i`, it sends the `SENSING` output event to one of its neighbors (i.e, a target cell) or it does not send any output event for some time duration. When the agent receives the `COME` input event from an neighbor grid Cell that is `Empty`, it changes it state to `Agent-i Moves to Cell(m,n)`, sets its state `Empty` and send an input to $cell(m,n)$. This logic is defined in the agent's external, internal, and output functions.

A grid cell may have no agents (i.e., its state is `Empty`). Such a grid cell may receive multiple `SENSING` input events from its neighbors with state `Has Agent-i`. Then the empty grid cell randomly selects one neighbor Cell, sets its state to `Coming Cell(m,n)`, and sends the `COME` event as input to the selected source cell. The empty cell state changes from `Coming Cell(m,n)` to 'Empty' again and waits for the agent to come. After the grid cell receives the signal with `Agent-i`, it becomes the cell with `Has Agent-i`.

The CCA-DEVS formalism supports multi-agent models and various collision conditions where there exist simultaneous input events and/or output events. In a two-dimension space, the agents may move in eight directions (i.e., Moore neighborhood), but only one direction is chosen. The empty cell may have more than one neighbor whose agent wants to move into it. In both cases, the agents randomly determine where to move and for the empty cells to decide which agents to accept. The external, internal, confluent, output, and time advance functions in parallel DEVS are employed to schedule all input and output events, state changes, and timing.

An example for a CCA-DEVS with four grid cells is shown in Figure 1. In this illustration, $cell(0,0)$ and $cell(0,2)$ are empty and $cell(0,1)$ and $cell(0,3)$ having Agent 1 and Agent 2. Agents randomly choose to move to grid $cell(0,2)$, a cell without an agent. Agent 1 in $cell(0,1)$ is shown to be randomly chosen to move to $cell(0,2)$. The empty $cell(0,0)$ is not chosen by Agent 1 (i.e., $cell(0,0)$ does not receive any sensing signal from Agent 1). The empty $cell(0,2)$ chooses Agent 1 to occupy it. Then, $cell(0,2)$ sends a message to Agent 1 which in turn moves to $cell(0,2)$, with $cell(0,1)$ becoming `Empty`. In Figure 1 the rounded rectangles boxes show the state information of the particular grid cells, the circles show decision functions, the arrows with dash lines show internal transition activities, the arrows with solid lines show external transition activities, and the arrows with `[msg:` $\cdots$`]` denote output functions.

3.2.2 Diffusion Modeling

The CCA-DEVS can be used to define partial differential equation models. Diffusion models can have simple to complex dynamics. A simple model has a constant rate of change in the diffusion gradient. It can be specified using the finite difference scheme (Recktenwald 2004) where the partial differential equations are discretized in space and time. Stability is a key part of defining PDE models (Cellier and Kofman 2006). Satisfying some desired accuracy and precision of the individual and collective dynamics of PDE models is subject to the models that it is composed with. The CCA-DEVS can specify cellular automata PDE Diffusion models.

Finite difference solvers can be used for simulating PDE models. A finite difference scheme is essentially a numerical method where space is discretized on a regular grid of points with state variables changing using discrete time steps. The density of the diffusion is defined using uniform grid with locations (x,y) and time instances $t_n = t_0 + n\Delta t, n \in \mathbb{N}$. Considering two-dimensional PDEs, discretization can be achieved using explicit FTCS (Forward-Time Central-Space) method with a limit on time discretization satisfying some stability condition (e.g., $\Delta t \leq \frac{\Delta x^2}{4D}$ for $\Delta x = \Delta y$ with D as a diffusion coefficient) (Recktenwald 2004).

4 COMPOSITIONS OF MULTI-AGENT AND DIFFUSION MODELS

The composable cellular automata modeling can be used to define and compose agent-based and diffusion model types. These two model types can have the same or different spatial dimensions and resolutions. Agents may move within their cellular space while the diffusion process is taking place. The CCA-DEVS models can have regular grid space structures with continuous and/or discrete time bases. In an agent CCA-DEVS model, each CA grid cell can have an agent model (i.e., a DEVS atomic model). An agent's movement is defined by assigning it to a grid cell (i.e., a grid cell has a Boolean state variable having value equal to either occupied or empty (see Section 3.2.1). When the continuous time base is used, an agent can have different movement speeds while any number of other agents do not move and/or play a role in the movement of other agents.

4.1 Cellular Automata DEVS, PDE and KIB Models

It is beneficial to use strongly polymorphic composite modeling instead of strongly monolithic composite modeling for certain types of systems. An example is cancer biology (Chang et al. 2015). Heterogeneous models can be composed using the Knowledge Interchange Broker (KIB) approach instead of using data sharing or message passing. Given the DEVS and PDE model types, each with its own syntax and semantics, another model defines their dynamical interactions. This interaction has its own syntax and semantics. The interactions are defined in terms of mapping functions, timing, control, and concurrency. The KIB approach leads to the composed models as well as their interactions to be strongly modular. This is key when the models to be composed are yo be strictly independent of one another (each model has its syntax and semantics). It is straightforward to develop a KIB DEVS model for Composable Cellular Automata models (Zhang, Sarjoughian, and Seok 2020).

4.2 Geo-referenced KIB

The Geo-referenced Knowledge Interchange Broker (Geo-KIB) approach lends itself to specify spatial scaling and resolution between cellular automata models (Boyd and Sarjoughian 2020). It has *GeoMaps* for data transformations between CAs that have independent spatial scales, grid cell sizes (resolutions), and boundaries. A GeoMap has structures and values associated with Geo-referenced data transformations for composed CAs. The number of rows and the number of columns define the grid cell sizes. They are used to define *MapRegions*. A MapRegion has north, south, west, and east bounds. The boundaries can be absolute or relative to CA dimensions. Theoretical approaches exist for transforming space from one resolution to another (Goodchild 2011). Geo-KIB uses two-dimensional cellular automata models that have linear spatial scales and mappings.

An input region (source map) can be mapped to an output region (destination map). A two-dimensional space with values specified for a destination map can be defined given a source map. The boundaries of the destination map can then be matched with the source map to find an overlapping area. The value of one destination cell is determined by finding an average of the values from the corresponding (overlapping) area of the source map with weighted area sizes. For each source map cell, its value is multiplied by the size of the area of overlap between the source and destination cells. The products are added from all the corresponding source cells to create a weighted sum. Similarly, the sizes of the overlapping areas are added to find a total area. The weighted sum is divided by the total area to find the weighted average value for the destination cell. Because of differences in resolution, the destination cell may not align with the boundaries of cells from the source map. It is possible for a destination cell to overlap multiple cells from the source map in different area proportions.

It should be noted the term Geo-reference is generally used for surface modeling. Nonetheless, it captures the concept for spatial (dis)aggregation between CAs with different scales and resolutions. Such a use-case is where multi-agent and diffusion cellular automata have different spatial resolutions.

4.3 Modeling Human Breast Cancer

It is beneficial to gain insight into the composite modeling approach described using cancerous cells interacting with their environment. On the one hand, the diffusion modeling is appropriate to define the gradient formation of chemokine (signaling proteins) named L12. On the other hand, the Composable Multi-agent Cellular Automata DEVS modeling approach lends itself for describing the chemotaxis movement of cancer cells. The chemotaxis is constrained by chemokine gradient. The spread of cancer involves cancer cells and diffusion processes to interact with one another. Models of cancer focuses on a signaling axis involving CXCL12 ligand and CXCR4 and CXCR7 receptors and their bindings to CXCL12+, CXCR4+, and CXCR7+ cells (Chang et al. 2015). The diffusion of soluble CXCL12 is modeled using 2D Partial Differential Equation (PDE). The cancer cells are modeled as agents in 2D space. The CXCL12+ cells secret CXCL12 chemokine. The CXCR4+ cells move towards the chemoattractant CXCL12, and the CXCL12+ and CXCR7+ cells move randomly. Interactions between the cells and with its environment take place at discrete time steps. More details can be found in (Zhang, Sarjoughian, and Seok 2020; Zhang 2021).

The diffusion process of the CXCL12 phenomenon is defined based on discretizing the space continuum into a grid of square shapes. In CCA-DEVS, three types of cancer cells (CXCR4+, CXCR7+, and CXCL12+) are modeled as agents with the ability to move in a 2D cellular space. The environment for the cancer cells to move within is modeled as CXCL12 chemokine. The environment serves as gradient diffusion to control the movement and speed of cancer cells. The diffusion space is can be discretized to be the same size as the grid cells for the cancer cells. Each grid cell of CXCL12 chemokine is a 4th-order ODE developed in OpenModelica solving by RK4 with the step size of 10ms, which is easily exported as a

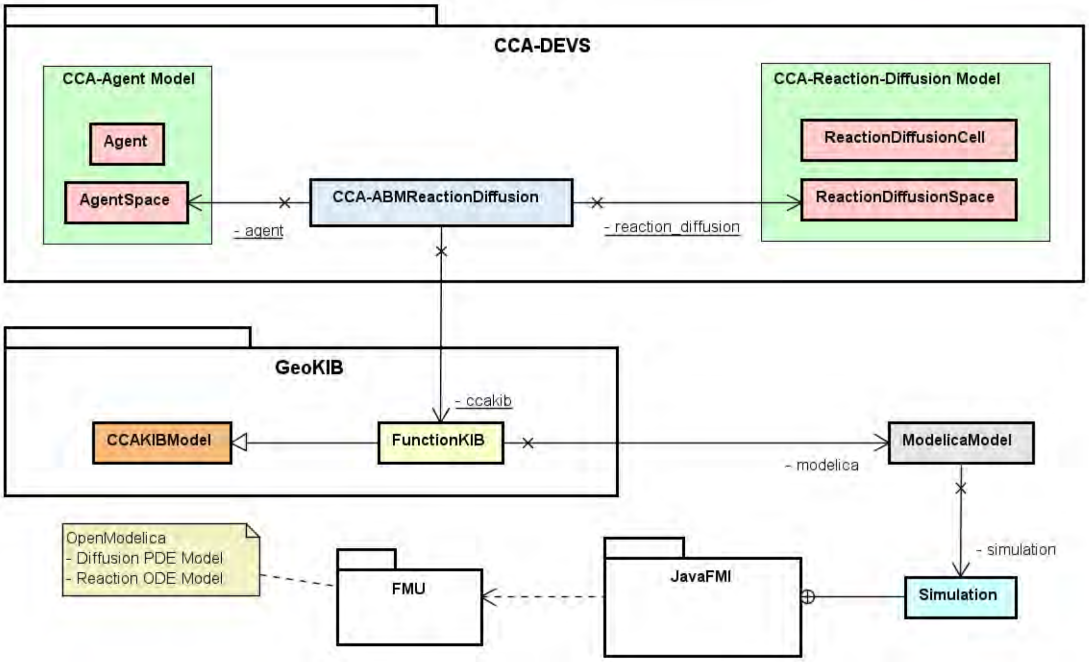


Figure 2: UML class diagram depicting the essential classifiers and relationships of the Geo-referenced CCA-DEVS, KIB, and OpenModelica simulation framework.

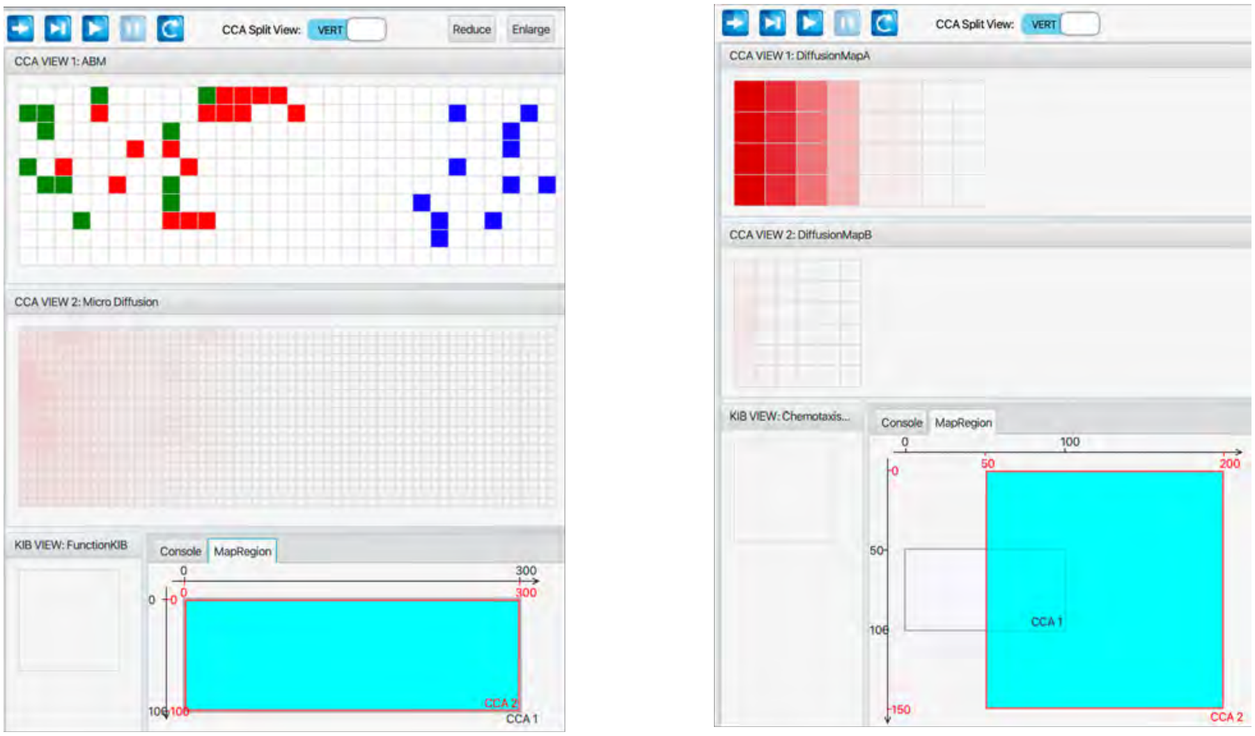
Functional Mock-Up Unit (FMU). The spatial x and y dimensions for each grid cell hosting one cancer cell is 10×10 μm. The cellular spatial dimension of the environment is 60×20.

Simulating this hybrid model for a 24-hour period shows the expected chemotaxis of the CXCR4+ cells. These cells move toward the region of the environment with high-density CXCL12 chemokine while the CXCR7+ and CXCL12+ cancer cells move randomly (Zhang 2021). The CXCL12+ Cell agents move randomly while releasing some amount of L12 protein. The coupling between these the CCA multi-agent and diffusion models is uni-directional. It is from the CXCL12+ Cell agents to the L12 chemotaxis diffusion process. The benefit of separating the diffusion and agent models is important for tracking and inspecting the dynamics of the cells selectively and individually.

5 CO-SIMULATION OF HYBRID AGENT-BASED MULTI-AGENT AND DIFFUSION MODELS

The DEVS-Suite, OpenModelica, and FMI are used to create, simulate, and evaluate the above breast cancer phenomena. The UML class diagram in Figure 2 shows (i) the CCA-DEVS, (ii) GeoKIB, and (iii) OpenModelica with the JavaFMI (https://bitbucket.org/siani/javafmi/wiki/Home/). This simplified depiction of the implemented co-simulation framework highlights support for the composition of multi-resolution, multi-agent CA-based DEVS and Modelica models with the Geo-referenced GeoKIB.

When the CCA-DEVS master simulation starts, the JavaFMI `Simulation` is instantiated and initialized by loading the generated FMU for the human breast cancer model. The FMU data inside the `Simulation` is overwritten by the CCA-DEVS multi-agent model executed in the DEVS-Suite simulator in each simulation cycle. Then, the `doStep()` operation provided in the JavaFMI `Simulation` causes the OpenModelica RK4 solver to execute. Upon completing the FMU simulation steps, the output from FMU is read and used by the CCA-DEVS model via the `ModelicaModel`. This sequential procedure for writing and reading data in the master simulator repeats until the `Simulation` stops or is terminated. The diffusion model is modeled and simulated using OpenModelica. The cancer cells are developed using the DEVS-Suite simulator. These simulators are integrated using JavaFMI for co-simulation.



(a) Composition with complete overlapping regions

(b) Composition with partially overlapping regions

Figure 3: Simulation and visualization framework for composable multi-resolution CA-based DEVS, KIB, and Modelica models. (a) CCA multi-agent and diffusion models with identical geometric regions. (b) CCA diffusion models with overlapping geometric regions.

A set of experiments are defined, developed, simulated, and shown to be comparable to a custom-built multi-scale BrSimulator (Chang et al. 2015). The simulation experiments of the composite breast cancer model, using Two One-Sided Test and Chi-Squared Test, are shown to be statistically comparable to those that can be obtained from the custom-built BrSimulator (Zhang, Sarjoughian, and Seok 2020). This comparison is carried out to note the developed heterogeneous model composition approach supports the degree of fidelity employed in realistic simulations of human breast cancer biology. This co-simulation framework is used to simulate some four hundred thousand cells with CCAs having different spatial and resolution scales on a desktop computer with Intel i7 2.60 GHz CPU, 64 GB RAM, and Windows 10 OS.

The co-simulator framework offers multi-scale multi-agent CAA-DEVS visualization. Figure 3 shows run-time combined animations for agent-based and diffusion CCA-DEVS models that have the same spatial scales but different resolutions. Another example is shown where two models have different spatial scales and resolutions. Other features include the flexibility to examine models' dynamics independently as linear and superdense input, output, and state time trajectories. The content of the input and output messages exchanges between neighboring cells and customizable Cellular Automata animations can be tracked and examined step-by-step. Animation of message passing can be set in the simulator's control panel for playback. These features do not require writing scripts or developing and integrating code into the simulator. The simulator also supports black-box testing, an important capability for validating simulations. These capabilities support step-by-step run-time simulation, debugging, testing, and storing large-scale simulation results in a PostgreSQL database for post evaluations and visualization.

6 CONCLUSION

This research describes a multi-agent Cellular Automata DEVS and Modelica models that can have distinct spatiotemporal scales and resolutions. Heterogeneous model composition is achieved using the Geo-referenced Knowledge Interchange Broker modeling approach supported with strong input/output modularity and componentization. Cellular automata models can have their independent temporal, spatial, and resolution structures and behaviors. Along this path, heterogeneous model composition aids model verification, simulation validation, formulation of experiments, and visual analytics in an autocatalytic fashion. The co-simulator framework offers unique features for configurations and run-time visualizations. The CA-DEVS and CCA-DEVS models with GeoKIB can help better model, simulate, and understand cell-to-cell and CCA-to-CCA interactions.

It is anticipated that the proposed composable modeling approach and co-simulation framework to lead to research in two directions. First, enabling increases in model scale and complexity should serve simulation studies that require compositions of very large multi-resolution heterogeneous models (e.g., models with billions of parts and numerous configurations) supported with high-performance and distributed computing platforms. Second, aiding domain experts in biological, environmental, and social systems to systematically develop and simulate hybrid models to pose questions and find answers to them. This suggests some models should have extreme spatial and time scales with complex structures and behavior dynamics under varying configurations.

ACKNOWLEDGMENTS

We acknowledge the reviewers' helpful critiques and particularly suggestions by one of them on an early version of this paper. We are also thankful to Rodrigo Castro, Theresa Roeder, and Gabriel Wainer.

REFERENCES

ACIMS 2021. "DEVS-Suite Simulator". https://acims.asu.edu/software/devs-suite/. [accessed 5th May 2022].

Alur, R. 2015. *Principles of Cyber-Physical Systems*. London: MIT Press.

Åström, K. J., H. Elmqvist, S. E. Mattsson et al. 1998. "Evolution of Continuous-time Modeling and Simulation". In *ESM*, 9–18.

Boyd, W. A., and H. S. Sarjoughian. 2020. "Composition of Geographic-Based Component Simulation Models". In *Proceedings of the 2020 Winter Simulation Conference*, edited by R. Thiesing, T. Roeder, K.-H. G. Bae, S. Lazarova-Molnar, Z. Zheng, B. Feng, and S. Kim, 2257–2268. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Cellier, F. E., and E. Kofman. 2006. *Continuous System Simulation*. New York, NY: Springer Science & Business Media.

Chang, S. L., S. P. Cavnar, S. Takayama, G. D. Luker, and J. J. Linderman. 2015. "Cell, Isoform, and Environment Factors Shape Gradients and Modulate Chemotaxis". *PloS One* 10(4):e0123450.

Chopard, B., J.-L. Falcone, P. Kunzli, L. Veen, and A. Hoekstra. 2018. "Multiscale Modeling: Recent Progress and Open Questions". *Multiscale and Multidisciplinary Modeling, Experiments and Design* 1(1):57–68.

Dada, J. O., and P. Mendes. 2011. "Multi-scale Modelling and Simulation in Systems Biology". *Integrative Biology* 3(2):86–96.

Davis, P. K., and R. H. Anderson. 2003. "Improving the Composability of Department of Defense Models and Simulations". Technical report, RAND CORP, Santa Monica, CA.

Goodchild, M. F. 2011. "Scale in GIS: An Overview". *Geomorphology* 130(1-2):5–9.

Hoekstra, A. G., J. Kroc, and P. M. Sloot. 2010. *Simulating Complex Systems by Cellular Automata*. Heidelberg: Springer-Verlag.

Hunt, C. A., R. C. Kennedy, S. H. Kim, and G. E. Ropella. 2013. “Agent-based Modeling: A Systematic Assessment of Use Cases and Requirements for Enhancing Pharmaceutical Research and Development Productivity”. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine* 5(4):461–480.

Luke, S., C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan. 2005. “MASON: A Multiagent Simulation Environment”. *Simulation* 81(7):517–527.

Mayer, G. R., and H. S. Sarjoughian. 2009. “Composable Cellular Automata”. *Simulation* 85:735–749.

Mayer, G. R., and H. S. Sarjoughian. 2016. “Building a Hybrid DEVS and GRASS Model using a Composable Cellular Automaton”. *International Journal of Modeling, Simulation, and Scientific Computing* 7(01):1–31.

Recktenwald, G. W. 2004. “Finite-difference Approximations to the Heat Equation”. *Mechanical Engineering* 10:1–27.

Sanz, V., and A. Urquia. 2021. “Combining PDEVS and Modelica for Describing Agent-based Models”. *Simulation*.

Sarjoughian, H. S. “Restraining Complexity and Scale Traits for Component-based Simulation Models”. In *Proceedings of the 2017 Winter Simulation Conference*, edited by E. H. Page, G. Wainer, V. Chan, A. D’Ambrogio, G. Zacharewicz, and N. Mustafee, 675–689. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Sarjoughian, H. S. 2006. “Model composability”. In *Proceedings of the 2006 Winter Simulation Conference*, edited by D. Nicol, R. Fujimoto, B. Lawson, J. Liu, F. Perrone, and F. Wieland, 149–158. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Simon, H. A. 1962. “The architecture of complexity”. 467–482.

von Neumann, J., and A. W. Burks. 1966. *Theory of Self-reproducing Automata*. Urbana: University of Illinois Press.

Wainer, G. A. 2017. *Discrete-event Modeling and Simulation: A Practitioner’s Approach*. Boca Raton: CRC Press.

Zeigler, B. P., A. Muzy, and E. Kofman. 2018. *Theory of Modeling and Simulation: Discrete Event & Iterative System Computational Foundations*. 3rd ed. London: Academic press.

Zhang, C. 2021. *A Multi-Scale, Component-Based, Composable Cellular Automata Modeling and Simulation Framework*. Ph. D. thesis, School of Computing and Augmented Intelligence, Arizona State University, Tempe, Arizona, USA.

Zhang, C., and H. S. Sarjoughian. 2017. “Cellular Automata DEVS: A Modeling, Simulation, and Visualization Environment”. In *ACM International Conference Proceeding Series*, 11–19.

Zhang, C., H. S. Sarjoughian, and M. G. Seok. 2020. “A Framework for Composable Cellular Automata DEVS Modeling, Simulation, and Visualization”. In *2020 Spring Simulation Conference*, edited by F. Barros, X. Hu, H. Kavak, and A. Del Barrio, 1–12. Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

HESSAM S. SARJOUGHIAN is an Associate Professor of Computer Science and Computer Engineering in the School of Computing and Augmented Intelligence (SCAI) at Arizona State University (ASU), Tempe, Arizona, and co-director of the Arizona Center for Integrative Modeling & Simulation (https://acims.asu.edu/). He is a core faculty in the School of Complex Adaptive Systems at Arizona State University. His research interests include model theory, poly-formalism modeling, collaborative modeling, simulation for complexity science, and M&S frameworks/tools. He can be contacted at sarjoughian@asu.edu.

CHAO ZHANG is a founder of a startup company for Pet Activity and Health Tracker devices with support for Web 3.0 application. He received his Ph.D. from the School of Computing and Augmented Intelligence at Arizona State University in May 2021. He can be contacted at imchaozhang@gmail.com.