Composition of Composable Cellular Automata with Respect to Their Dimensional Attributes

Gary R. Mayer
Department of Computer Science
Southern Illinois University Edwardsville
61 Circle Drive
Edwardsville, IL 62026, USA
gamayer@siue.edu

Keywords: Composable Cellular Automata, hybrid simulation, multi-formalism modeling, composability

Abstract

The Composable Cellular Automata (CCA) specification, introduced in a previous article, formally defines a system for building cellular automata models that can be composed with other models. The purpose is to build hybrid simulation models with uniquely modeled subsystems. It is feasible that two or more subsystems of a large, complex system are modeled using CCA. For the purposes of refining domain abstraction, reducing composition complexity, or improving model execution, the need to compose two or more CCA to create a single CCA may exist. It is then important for a modeler to understand the implications that specific disparities between the CCA have in their composition. To that end, this paper builds upon the original publication to describe closure properties of formal composition of multiple CCA with respect to their dimensional attributes—cell indices and time.

1. INTRODUCTION

A hybrid simulation model is one in which the subsystem models are disparate yet interact to represent a complex system. The disparities can include differences in formalism, structure, timing, and behavior. The usefulness in maintaining the disparities is that the subsystem simulation models can best represent their respective subsystems. However, the subsystem model differences must be managed to ensure a correct simulatable system model and simulation results. Managing software simulation systems entails dealing with various complexities and the interactions between subsystem components is one of those complexities. [1, 2] There are a number of ways in which these disparate models can be made to interact through formal means and they have trade-offs in terms of complexity management and robustness of model representation within a domain. One approach, *interoperability*, focuses on data communication between disparate simulation systems. High-level Architecture (HLA) is an example of this. [3] The other approach, *composition*, address interactions between heterogeneous model types whose execution approach is the same. [4, 5] These two approaches may also be combined.

In [6], the Composable Cellular Automata (CCA) formalism was introduced. Generally, cellular automata (CA) are useful as simulation model representatives for many systems that can be uniformly tessellated and treated as an interacting set of smaller systems. One of the key limitations of typical CA that the CCA specification overcomes is the provision of a formal mechanism for the cellular automata system to be composed with other formal models. The specification paper also discusses mapping one CCA to another. This approach retains the two CCA as individual entities. However, there may be times in which a modeler seeks to merge two or more CCA together in order to reduce the overall complexity of the simulation system. Toward this end, this paper defines closure properties of the dimensional attributes of Composable Cellular Automata (CCA), ensuring that the mechanisms employed by the CCA formalism and exhibited by the resulting CCA remain valid.

The section that immediately follows provides a summary introduction of the composable cellular automata specification. Section 1.1.3. introduces properties that apply to all CCA and lays the foundation for the subsequent discussions. Next, in Section 2., CCA composition is discussed from the perspective of cell indices and time. A discussion of the utility of CCA composition is provided next. Finally, Section 4. provides a summary of what is presented in the paper.

1.1. Composable Cellular Automata

CCA are formally specified cellular automata. They are defined in [6] and summarized here. There are two major components of CCA, a network and the cells contained within the network. The network encapsulates the cells such that all external input and output (I/O) must be handled by the network itself. The cells represent the individual automaton that maintain state, produce output based upon state, and undergo state transition based upon current state and input.

Ref. [6] discusses an approach to *mapping* input and output (I/O) between two CCA networks. This is feasible and remains a viable solution, especially if the two CCA are distributed or extremely disparate across their elements. However, it does not address a direct composition approach to managing CCA that are alike. In other words, how CCA can be composed with other CCA and treated as a single CCA if they differ in specific ways. The utility of this is that it may simplify the composition of the CCA models with an additional, non-CCA model; thus, making the hybrid model more simplistic (see [1] and [7] as examples). Another purpose to composing two or more CCA is to reduce execution overhead; removing the need for multiple simulators, as an example. This paper discusses how CCA may be composed with respect to their dimensional properties—cell representation and time. The overall goal is to describe approaches to formally compose CCA, and provide an understanding of the implications of doing such when faced with specific disparities between the CCA systems.

1.1.1. Overview of the CCA Specification

A Composable Cellular Automata network, N, is defined as

$$N = \langle X_N, Y_N, D, \{M_{ijk}\}, T, F \rangle, \tag{1}$$

and each cell component, M_{ijk}, within the network is specified by

$$M_{ijk} = \langle X_{ijk}, Y_{ijk}, Q_{ijk}, I_{ijk}, \delta_{ijk}, \lambda_{ijk}, T \rangle. \tag{2}$$

As discussed in [6], the first two elements of N, X_N and Y_N, are the input from an external system to the network and output from the network to an external system, respectively. D is a set of indices that uniquely identify each cell within the set of homogeneous cells, $\{M_{ijk}\}$, that belongs to the network (subscript ijk is the index for a unique element of a three-dimensional network). T is a finite set of time-ordered, time intervals that structures the discrete-time dynamics of the network. The last set, F, contains the mapping functions between the CCA cells and the network as a whole.

The cells, M_{ijk}, are defined by, X_{ijk} and Y_{ijk}, which are the input to and output from each cell. Each is a union of data internal and external to the network. $X_{ijk} = \dot{X}_{ijk} \cup \overline{X}_{ijk}$, which represents input from the cell's influencers and external input mapped to this cell, respectively. $Y_{ijk} = \dot{Y}_{ijk} \cup \overline{Y}_{ijk}$, which represents output to the cells that this cell influences and output that acts as part of the external output from this network, respectively. Q_{ijk} is the set of possible states for the component. I_{ijk} is the set of indices that identify this cell's influencers (i.e., its neighborhood). The component's state transition function is δ_{ijk}, and the output function is λ_{ijk}. $\delta_{ijk} : Q_{ijk} \times X_{ijk} \to Q_{ijk}$ and $\lambda_{ijk} : Q_{ijk} \to Y_{ijk}$. T is the same set of time-ordered, time intervals that exists in the network tuple in (1). The significance of this is that it ensures that every cell in the network is using the same set of time-ordered, time intervals and, therefore, every cell undergoes state transition at the same discrete time. It should also be noted that a specific cell has no a priori knowledge of the network, including characteristics such as total number of cells in the network and connectivity (e.g., Moore versus von Neumann networks).

All external input and output from the cells are managed through the network's mapping functions within F. The network encapsulates the cells and, as such, external entities must rely upon the interfaces offered by the mapping functions. These mapping functions are defined by the modeler and handle both input to the cells from external systems and output from the cells to those systems. The mapping functions may aggregate or disaggregate input and output. These mapping functions may also be specified such that some cells have no direct mapping to or from external systems. External entities may not directly provide data to or receive data from specific cells. Cells and the network itself are not required to produce output or make use of input. As such, values may be $\emptyset$. Imagine a watershed model where an external environment model inserts rain at the top of a mountain, the CCA models the watershed down the mountain, and the output is the rainfall that drains into a riverbed. Mappings would be created such that all input go to the cells representing the top of the mountain and only output from those cells representing the bottom of the mountain would be provided to an external system model.

1.1.2. Mapping of Two CCA

Two CCA mapped to one another may be the same or different in terms of structure, possible states, and transition functions—any of the network and component elements. Fig. 1 and Fig. 2 provide general examples of mapping from a two-dimensional to a three-dimensional CCA system and back, respectively. These mappings require that the output mapping function from one CCA be coordinated with the input mapping function to the other. For example, the two-dimensional CCA output mapping function would output all cell values and retain them as nine distinct values. The three-dimensional CCA input mapping function would input nine values and map them to each quadrant in two of that CCA model's three layers.

The use of the mapping functions can always be used to ensure correct composition of the two model subsystems. For instances where the interaction between the subsystems is not easily defined by a homogeneous cellular automata neighborhood, like the examples in Fig. 1 and Fig. 2, it may be the only option. However, if the example were to be changed such that the two-dimensional CCA had the same tessellation as a layer of the three-dimensional CCA, and the two-dimensional CCA only interacted with the top layer of the three-dimensional CCA, then it may be advantageous to treat the two CCA as

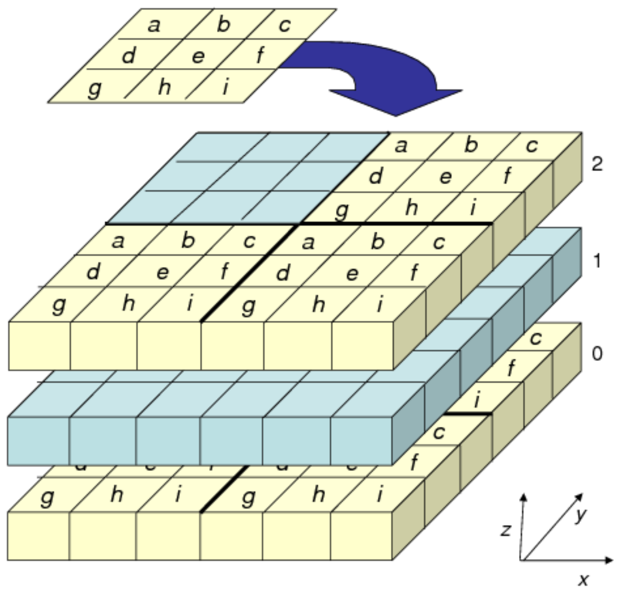

Figure 1. CCA Mapping: 2D external output to 3D external input. Cells without letters indicate no external input is received.

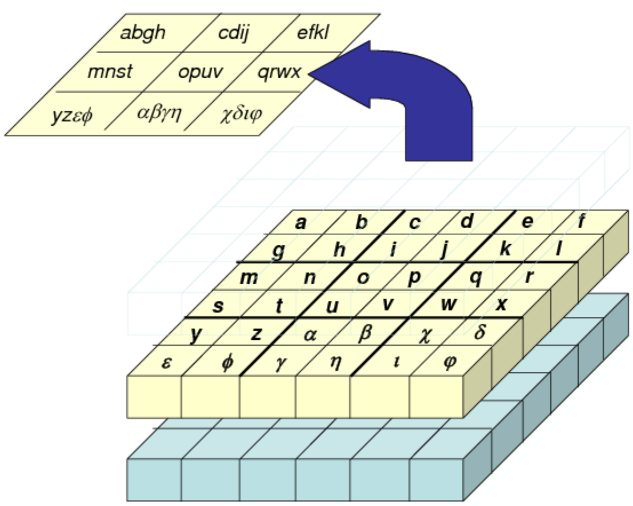

Figure 2. CCA Mapping: 3D external output to 2D external input. Cells without letters indicate no external output is generated. (Top layer outlined for clarity.)

one in order to simplify the model development and simulation.

1.1.3. Basic Properties of CCA

Let Φ and Ψ represent two distinct composable cellular automaton. The network specifications for Φ and Ψ are $N^{\Phi} = \langle X_N^{\Phi}, Y_N^{\Phi}, D^{\Phi}, \{M_{ijk}^{\Phi}\}, T^{\Phi}, F^{\Phi}\rangle$ and $N^{\Psi} = \langle X_N^{\Psi}, Y_N^{\Psi}, D^{\Psi}, \{M_{ijk}^{\Psi}\}, T^{\Psi}, F^{\Psi}\rangle$, respectively. Similarly, the specifications for the cell components of Φ and Ψ are $M_{ijk}^{\Phi} = \langle X_{ijk}^{\Phi}, Y_{ijk}^{\Phi}, Q_{ijk}^{\Phi}, I_{ijk}^{\Phi}, \delta_{ijk}^{\Phi}, \lambda_{ijk}^{\Phi}, T^{\Phi}\rangle$ and $M_{ijk}^{\Psi} = \langle X_{ijk}^{\Psi}, Y_{ijk}^{\Psi}, Q_{ijk}^{\Psi}, I_{ijk}^{\Psi}, \delta_{ijk}^{\Psi}, \lambda_{ijk}^{\Psi}, T^{\Psi}\rangle$, respectively.

Definition 1. *$\Phi = \Psi$ is defined as all subset elements of tuples Φ and Ψ being equal. Formally,*

$$\Phi = \Psi \Leftrightarrow (X_N^{\Phi} = X_N^{\Psi}) \wedge (Y_N^{\Phi} = Y_N^{\Psi}) \wedge (D^{\Phi} = D^{\Psi}) \wedge \\ (\{M_{ijk}^{\Phi}\} = \{M_{ijk}^{\Psi}\}) \wedge (T^{\Phi} = T^{\Psi}) \wedge (F^{\Phi} = F^{\Psi}), \tag{3}$$

where $\{M_{ijk}^{\Phi}\} = \{M_{ijk}^{\Psi}\} \Leftrightarrow \forall ijk : (X_{ijk}^{\Phi} = X_{ijk}^{\Psi}) \wedge (Y_{ijk}^{\Phi} = Y_{ijk}^{\Psi}) \wedge (Q_{ijk}^{\Phi} = Q_{ijk}^{\Psi}) \wedge (I_{ijk}^{\Phi} = I_{ijk}^{\Psi}) \wedge (\delta_{ijk}^{\Phi} = \delta_{ijk}^{\Psi}) \wedge (\lambda_{ijk}^{\Phi} = \lambda_{ijk}^{\Psi}) \wedge (T^{\Phi} = T^{\Psi})$.

Definition 2. *Let M_{abc} and M_{xyz} be two components of the same CCA network, N. Then, $M_{abc} \approxeq M_{xyz}$ is defined as the two cell components being **homogeneous**. Formally,*

$$\forall(a,b,c) \in D, \forall(x,y,z) \in D, (a,b,c) \neq (x,y,z) : M_{abc} \approxeq M_{xyz} \models \\ X_{abc} = X_{xyz}, Y_{abc} = Y_{xyz}, Q_{abc} = Q_{xyz}, f_{abc}(D) = f_{xyz}(D), \\ \delta_{abc} = \delta_{xyz}, \lambda_{abc} = \lambda_{xyz}, T_{abc} = T_{xyz}, \tag{4}$$

where $f_{abc}(D) \mapsto I_{abc}$ and $f_{xyz}(D) \mapsto I_{xyz}$ are the functions relating the network set of indices to a cell's set of influencers.

Note that the relationship between the cell and its influencers is what is being stated as equal, not the set of influencers themselves.

Definition 3. *$\Phi \approxeq \Psi$ is defined as Φ and Ψ being **homogeneous**—having $\{M_{ijk}^{\Phi}\} \approxeq \{M_{ijk}^{\Psi}\}$. Formally,*

$$N^{\Phi} \approxeq N^{\Psi} \Leftrightarrow \{M_{ijk}^{\Phi}\} \approxeq \{M_{ijk}^{\Psi}\} \tag{5}$$

where $\{M_{ijk}^{\Phi}\} \approxeq \{M_{ijk}^{\Psi}\}$ is as defined in Definition 2.

By observation, it should be understood that if $\Phi = \Psi$, then $\Phi \approxeq \Psi$.

Definition 4. *$\Phi \sim \Psi$ is defined as Φ and Ψ being **similar**, differing only in the time intervals subset, T. Formally,*

$$\Phi \sim \Psi \equiv \{\Phi - \{T^{\Phi}\}\} = \{\Psi - \{T^{\Psi}\}\}, \tag{6}$$

where $T^{\Phi} \neq T^{\Psi}$. Note that $\Phi \sim \Psi \models T \in \{M_{ijk}^{\Phi}\} \neq T \in \{M_{ijk}^{\Psi}\} \because T \in N = T \in \{M_{ijk}\}$.

Given that the discrete-time sets, T, are different between the cell components of the two network systems, Φ and Ψ are not homogeneous as defined in Definition 3.

Definition 5. ***Composition** of two CCA is a disjoint union of the CCA plus a composition tuple containing any new external I/O mappings to the resultant set of cells and any* *new influencers to specific cells. In other words, it is the union of each of the subset elements of the CCA tuples and each subset element is pairwise disjoint, the union of the network mapping function set with an* ***mapping composition set****, and the union of the influencer set of each cell with an* ***influencer composition set****. Formally, let* $\Xi = \{F', \{I'_{ijk}\}\}$ *be the composition tuple, where* F' *is the mapping composition set and* $\{I'_{ijk}\}$ *is the influencer composition set. Then, composition* $\equiv \Phi \odot_{\Xi} \Psi = \langle\{X_N^{\Phi} \cup X_N^{\Psi}\}, \{Y_N^{\Phi} \cup Y_N^{\Psi}\}, \{D^{\Phi} \cup D^{\Psi}\}, \{\{M_{ijk}^{\Phi}\} \uplus \{M_{ijk}^{\Psi}\}\}, \{T^{\Phi} \cup T^{\Psi}\}, \{F^{\Phi} \cup F^{\Psi} \cup F'\}\rangle$*, and* $\{\{M_{ijk}^{\Phi}\} \uplus \{M_{ijk}^{\Psi}\}\} = \{\{M_{ijk}^{\Phi}\} \sqcup \{M_{ijk}^{\Psi}\}\} = \forall ijk : \langle\{X_{ijk}^{\Phi} \cup X_{ijk}^{\Psi}\}, \{Y_{ijk}^{\Phi} \cup Y_{ijk}^{\Psi}\}, \{Q_{ijk}^{\Phi} \cup Q_{ijk}^{\Psi}\}, \{I_{ijk}^{\Phi} \cup I_{ijk}^{\Psi} \cup I'_{ijk}\}, \{\delta_{ijk}^{\Phi} \cup \delta_{ijk}^{\Psi}\}, \{\lambda_{ijk}^{\Phi} \cup \lambda_{ijk}^{\Psi}\}, \{T^{\Phi} \cup T^{\Psi}\}\rangle$*. Note that* $\Xi = \emptyset \Rightarrow \Phi \odot_{\Xi} \Psi = \Phi \odot_{\emptyset} \Psi = \Phi \sqcup \Psi$.

Theorem 1. *If* $\Phi = \Psi$*, then* $\Phi \odot_{\emptyset} \Psi = \Phi$.

Proof. Let $\{I'_{ijk}\} = \emptyset$. Using Definition 1, substitute Ψ for Φ into the equation for $\Phi \odot_{I'_{ijk}} \Psi$ given in Definition 5. Then, $\Phi \odot_{\emptyset} \Psi = \langle\{X_N^{\Phi} \cup X_N^{\Phi}\}, \{Y_N^{\Phi} \cup Y_N^{\Phi}\}, \{D^{\Phi} \cup D^{\Phi}\}, \{\{M_{ijk}^{\Phi}\} \uplus \{M_{ijk}^{\Phi}\}\}, \{T^{\Phi} \cup T^{\Phi}\}, \{F^{\Phi} \cup F^{\Phi}\}\rangle$, and $\{\{M_{ijk}^{\Phi}\} \uplus \{M_{ijk}^{\Phi}\}\} = \forall ijk : \langle\{X_{ijk}^{\Phi} \cup X_{ijk}^{\Phi}\}, \{Y_{ijk}^{\Phi} \cup Y_{ijk}^{\Phi}\}, \{Q_{ijk}^{\Phi} \cup Q_{ijk}^{\Phi}\}, \{I_{ijk}^{\Phi} \cup I_{ijk}^{\Phi}\}, \{\delta_{ijk}^{\Phi} \cup \delta_{ijk}^{\Phi}\}, \{\lambda_{ijk}^{\Phi} \cup \lambda_{ijk}^{\Phi}\}, \{T^{\Phi} \cup T^{\Phi}\}\rangle$. By definition of a union of sets, $\{\{M_{ijk}^{\Phi}\} \uplus \{M_{ijk}^{\Phi}\}\} = \{M_{ijk}^{\Phi}\}$, and the first tuple simplifies to $\langle\{X_N^{\Phi}\}, \{Y_N^{\Phi}\}, \{D^{\Phi}\}, \{\{M_{ijk}^{\Phi}\}\}, \{T^{\Phi}\}, \{F^{\Phi}\}\rangle$, which is the specification for Φ. $\therefore$ if $\Phi = \Psi$, then $\Phi \odot_{\emptyset} \Psi = \Phi$. $\square$

2. PROPERTIES OF DIMENSIONAL ATTRIBUTES

2.1. Representation

Definition 6. *If two CCA are homogeneous and have the same set of cell identifiers, D, and the same set of cell influencers,* $\{I_{ijk}\}$*, then the two CCA possess the same domain representation,* $\mathfrak{D}$*. Formally,*

$$(D^{\Phi} = D^{\Psi}) \wedge (\forall ijk : I_{ijk}^{\Phi} = I_{ijk}^{\Psi}) \Leftrightarrow \mathfrak{D}^{\Phi} = \mathfrak{D}^{\Psi}. \qquad (7)$$

D models the tessellation of the domain space while $\{I_{ijk}\}$ captures the abstraction of domain element interactions (specified by the network). Arbitrary values can be assigned to i, j, and k as labels in D. However, for the purposes of evaluation of a CCA there are two approaches. First, from a domain-neutral perspective, all indices must be assumed to use the same coordinate system, start at $(0,0,0)$, and then be numbered sequentially based upon movement in a respective dimension. Alternatively, the semantics of the values with respect to the domain must be considered. Thus, stating $D^{\Phi} = D^{\Psi}$ entails that all of the same discrete-elements of the domain are being represented by D. As examples of differences in D, consider a grid-shaped tessellation versus a hexagonal one. For differences in $\{I_{ijk}\}$ consider a Moore network versus a von Neumann network.

Definition 7. ***Regions****,* $\mathfrak{R}$*, of CCA network, N, are a set of references to distinct sets of cells in the network, to which a specific external input value in* X_N *is mapped. Formally,* $f : (\mathfrak{r}, x_N) \mapsto \{\bar{x}_{ijk}\}$*, where* $f \in F$*,* $\mathfrak{r} \in \mathfrak{R}$*,* $x_N \in X_N$*,* $\bar{x}_{ijk} \in \overline{X}_{ijk}$*,* $ijk \in D$*, and* $0 \leq |\{\overline{X}_{ijk}\}| \leq |D|$. Regions are a domain-dependent implementation concept (similar to ports in a Discrete-Event System (DEVS) specification implementation).

2.2. Composition with Disparate Indices

Confining a discussion of CCA differences to the cells themselves, $\{\Phi - \{D^{\Phi}, F^{\Phi}\}\} = \{\Psi - \{D^{\Psi}, F^{\Psi}\}\}$. Note that F is dependent on the indices in D, and so must be considered.

Theorem 2. *Two homogeneous single-celled CCA,* Φ *and* Ψ*, differing by set D (and, potentially, F and* $\{I_{ijk}\}$ *as well) can be composed into a third CCA,* Ω*.*

Proof. Let $\Phi \approxeq \Psi$, let $D^{\Phi} = \{(a,b,c)\}$ and $D^{\Psi} = \{(e,f,g)\}$; let $F^{\Phi} = \{f^{\Phi}\}$ and $F^{\Psi} = \{f^{\Psi}\}$, where $f^{\Phi} : (\mathfrak{r}^{\Phi}, x_N^{\Phi}) \mapsto \bar{x}_{abc}$ and $f^{\Psi} : (\mathfrak{r}^{\Psi}, x_N^{\Psi}) \mapsto \bar{x}_{efg}$; let $I_{abc}^{\Phi} = \{(a,b,c)\}$ and $I_{efg}^{\Psi} = \{(e,f,g)\}$, where $c \neq g$ and the remaining indice variables may be arbitrary values; and let $F' = \emptyset$ and $\{I'_{ijk}\} = \emptyset$.

Using Definition 5: $\Omega = \Phi \odot_{\Xi} \Psi \Rightarrow D^{\Omega} = D^{\Phi} \cup D^{\Psi} = \{(a,b,c), (e,f,g)\}$, $F^{\Omega} = F^{\Phi} \cup F^{\Psi} \cup F' = \{f^{\Phi}, f^{\Psi}\}$; and $\forall ijk : I_{ijk}^{\Omega} = I_{ijk}^{\Phi} \cup I_{ijk}^{\Psi} \cup I'_{ijk} = \{\{I_{abc}^{\Omega}\}, \{I_{efg}^{\Omega}\}\}$, where $\{I_{abc}^{\Omega}\} = \{I_{abc}^{\Phi}\}$ and $\{I_{efg}^{\Omega}\} = \{I_{efg}^{\Psi}\}$. The unions of the remaining, non-disparate tuple elements sets are elementary where $\{\}^{\Phi} = \{\}^{\Psi} = \{\}^{\Omega}$. $\therefore N^{\Omega} = \langle X_N^{\Omega}, Y_N^{\Omega}, D^{\Omega}, \{M_{ijk}^{\Omega}\}, T^{\Omega}, F^{\Omega}\rangle$ and $\{M_{ijk}^{\Omega}\} = \langle X_{ijk}^{\Omega}, Y_{ijk}^{\Omega}, Q_{ijk}^{\Omega}, I_{ijk}^{\Omega}, \delta_{ijk}^{\Omega}, \lambda_{ijk}^{\Omega}, T^{\Omega}\rangle$ define Ω. Note that F now contains two mapping functions, one that maps to each cell. $\square$

Corollary 1. *Two homogeneous single-celled CCA,* Φ *and* Ψ*, differing by set D (and, potentially, F and* $\{I_{ijk}\}$*) can be composed into a third CCA,* Ω*, if* F' *contains a mapping from the network to the collection of all cells within the network after composition (i.e.,* $F' = \{f\}$*, where* $\forall ijk, f : (\mathfrak{r}, x_N) \mapsto \{\bar{x}_{ijk}\}$*), and* $I'_{ijk} = \emptyset$*.*

Proof. Let $\Phi \approxeq \Psi$, let $F^{\Phi} = \{f_{\phi}\}$, where $\forall ijk \in D^{\Phi}, f_{\phi} : (\mathfrak{r}, x_N) \mapsto \{\bar{x}_{ijk}\}$; $F^{\Psi} = \{f_{\psi}\}$, where $\forall ijk \in D^{\Psi}, f_{\psi} : (\mathfrak{r}, x_N) \mapsto \{\bar{x}_{ijk}\}$; and $F' = \{f_{\omega}\}$, where $\forall ijk \in D^{\Omega}, f_{\omega} : (\mathfrak{r}, x_N) \mapsto \{\bar{x}_{ijk}\}$, and let $I'_{ijk} = \emptyset$.

Then, $\Omega = \Phi \odot_{\Xi} \Psi \Rightarrow F^{\Omega} = \{F^{\Phi} \cup F^{\Psi} \cup F'\} = \{f_{\phi}, f_{\psi}, f_{\omega}\}$. As before, the unions of the remaining tuple element sets are elementary and Ω is properly defined. Note that three external I/O mappings now exist—one to each cell and a third to both cells. $\square$

Corollary 2. *Two homogeneous single-celled CCA, Φ and Ψ, differing by set D (and, potentially, F and $\{I_{ijk}\}$) can be composed into a third CCA, Ω, if I'_{ijk} contains the index of the cell with which ijk is composed (i.e., $I'_{abc} = \{(e,f,g)\}$ and $I'_{efg} = \{(a,b,c)\}$), and $F' = \emptyset$.*

Proof. Let $\Phi \approxeq \Psi$, let $\forall ijk : \{I'_{ijk}\} = \{\{I^{\Phi}_{abc}\}, \{I^{\Psi}_{efg}\}\}$, where $\{I^{\Phi}_{abc}\} = \{(e,f,g)\}$ and $\{I^{\Psi}_{efg}\} = \{(a,b,c)\}$, and let $F' = \emptyset$.

Then, $\Omega = \Phi \odot_{\Xi} \Psi \Rightarrow \forall ijk : I^{\Omega}_{ijk} = \{I^{\Phi}_{ijk}\} \cup \{I^{\Psi}_{ijk}\} \cup \{I'_{ijk}\} = \{\{I^{\Omega}_{abc}\}, \{I^{\Omega}_{efg}\}\}$, where $\{I^{\Omega}_{abc}\} = \{(a,b,c),\ (e,f,g)\}$ and $\{I^{\Omega}_{efg}\} = \{(e,f,g),(a,b,c)\}$. As before, the unions of the remaining tuple element sets are elementary and Ω is properly defined. Note that the two cells now influence each other as well as themselves. □

Corollary 3. *Given the associative property of a binary union, multiple homogeneous CCA; differing only by sets D, F, and $\{I_{ijk}\}$; can be composed in any order. $(\Phi \odot_{\Xi} \Psi) \odot_{\Xi} \Upsilon = \Phi \odot_{\Xi} (\Psi \odot_{\Xi} \Upsilon)$. Furthermore, as the unions are disjoint, each mapping composition set, F', and influencer composition set, I'_{ijk}, may be given with respect to each composition or all at once with the last composition as a union of all respective composition sets.*

Note that when it stated that the composition set is "with respect to" each composition, this means that the mappings and influencer indices are valid for the two CCA being composed (i.e., $ijk \in \{D^{\Phi} \cup D^{\Psi}\}$).

Whether or not the composed CCA have additional mappings and/or influence over a subset of the other to which it is composed, a network mapping to a set of dimensionally-disparate cells is a viable system. Thus, multiple CCA, differing only by sets D, F, and $\{I_{ijk}\}$ can be said to be *closed under composition*. This implies that a CCA may be built from a composition of one (dimensionally-unique) cell at a time or from a composition of cell sets containing at least one dimensionally-unique cell member between them.

Note that if two composed, homogeneous CCA contain a common "edge" cell then, even if the influencer composition set is $\emptyset$, the composed CCA will integrate the two CCA based upon the union of the influencers of the edge cell. For example, let Φ and Ψ be two two-celled CCA where $D^{\Phi} = \{(a,b,c),(e,f,g)\}$ and $D^{\Psi} = \{(e,f,g),(p,q,r)\}$; and let $I^{\Phi}_{ijk} = \{\{I^{\Phi}_{abc}\}, \{I^{\Phi}_{efg}\}\}$, where $\{I^{\Phi}_{abc}\} = \{(a,b,c),(e,f,g)\}$ and $\{I^{\Phi}_{efg}\} = \{(e,f,g),(a,b,c)\}$. Similarly, let $I^{\Psi}_{ijk} = \{\{I^{\Psi}_{efg}\}, \{I^{\Psi}_{pqr}\}\}$, where $\{I^{\Psi}_{efg}\} = \{(e,f,g),(p,q,r)\}$ and $\{I^{\Psi}_{pqr}\} = \{(p,q,r),(e,f,g)\}$; and let $F' = \emptyset$ and $I'_{ijk} = \emptyset$. Then, $\Omega = \Phi \odot_{\emptyset} \Psi \Rightarrow I^{\Phi}_{efg} \cup I^{\Psi}_{efg} \cup \emptyset = I^{\Omega}_{efg} = \{(e,f,g),(a,b,c),(p,q,r)\}$.

Note also that directional influence and automata networks such as torus can be implemented by excluding or including specific indices in I'_{ijk}. Furthermore, if in the preceding example the restriction for $F' = \emptyset$ is removed and instead $F' = \{f_{\omega}\}$, where $\forall ijk \in D^{\Omega}, f_{\omega} : (\mathfrak{r}, x_N) \mapsto \{\bar{x}_{ijk}\}$; and $F^{\Phi} = \{f_{\phi}\}$, where $\forall ijk \in D^{\Phi}, f_{\phi} : (\mathfrak{r}, x_N) \mapsto \{\bar{x}_{ijk}\}$; and $F^{\Psi} = \{f_{\psi}\}$, where $\forall ijk \in D^{\Psi}, f_{\psi} : (\mathfrak{r}, x_N) \mapsto \{\bar{x}_{ijk}\}$; then all three external I/O mapping functions, $\{f_{\phi}, f_{\psi}, and f_{\omega}\}$, will have at least one common cell to which they map.

2.3. Composition of Disparate Discrete-Time Segments

Attention is now turned to the composition of two CCA that differ with respect to their discrete-time segments, T. In this regard, $\Phi \sim \Psi$. It may not make much sense to compose two CCA with the same dimensional representation if they only vary in time. However, there is no dependency of D, I_{ijk}, or F on T; therefore, whether they are equal or not does not matter to what follows.

Theorem 3. *If two similar single-celled CCA are composed, the resultant CCA is a system with the network and all cells possessing discrete-time segments from both.*

Proof. Let $\Phi \sim \Psi$; and $T^{\Phi} = \{t^{\Phi}_1, t^{\Phi}_2, \ldots, t^{\Phi}_m\}$ and $T^{\Psi} = \{t^{\Psi}_1, t^{\Psi}_2, \ldots, t^{\Psi}_n\}$, where $T^{\Phi} \neq T^{\Psi}$ and $0 < m \leq n \leq \mathbb{N}^*$ and $\mathbb{N}^* \equiv \mathbb{N} - \{0, \infty\}$.

$\Omega = \Phi \odot_{\Xi} \Psi \Rightarrow T^{\Omega} = T^{\Phi} \cup T^{\Psi}$ at the network-level and $\forall ijk : T^{\Omega} = T^{\Phi} \cup T^{\Psi}$ at the cellular level. $T^{\Omega} = T^{\Phi} \cup T^{\Psi} = \{t^{\Omega}_1, t^{\Omega}_2, \ldots, t^{\Omega}_{m+n}\}$. Given that T is independent of index ijk, even at the cell-level, T^{Ω} at the cellular level is also $\{t^{\Omega}_1, t^{\Omega}_2, \ldots, t^{\Omega}_{m+n}\}$, retaining equality of discrete-time segments at the network and cellular levels. □

Once composed, all of the discrete-time segments are applied to all of the cells, M_{ijk}. Through the understanding of *similar* from Definition 4 and applying Definition 3, the resultant network is now a homogenenous set of cell components.

3. APPLICATION

In [1], an agent-environment hybrid model was created to exemplify a poly-formalism modeling approach. Fig. 3 represents the simulation models as they were developed. Agents were modeled using DEVS-Suite, a Java implementation of DEVS [8], to represent sugar farmers. The environment was modeled as two separate CCAs in the Geographic Resources Analysis Support System (GRASS)–a Geographical Information System (GIS). [9] One CCA modeled the landscape that was impacted by weather and agent activity. The other modeled the sugar that grew on the landscape and was harvested by the agents. The agent and the environment models are disparate in many ways including formalism, timing, a domain representation. Between the two models is an interaction model (IM). The IM composes the two models by explicitly modeling the interaction between them, accounting for the disparities to ensure a correct model and simulation results. Thus, agent models hand requests for environment information or orders for environmental change to the IM. The IM then handles data transformation and timing control mechanisms to interact with the environment model CCAs. Response data from the CCAs to the agent model are handled similarly in a reverse fashion.

As can be seen in Fig. 3, a set of mapping functions is used between the agent model and the environment subsystem models. Also, there exists mapping functions between the two CCA. These mapping functions, F_{sugr} and F_{land}, are the same mapping functions from the CCA network that were discussed in Section 1.1.. As GRASS does not have an innate sense of timing, a DEVS timer model had to be created to drive each of the CCAs. This timing model is a state machine that cycles through the CCA state transition and output functions. A separate set of GRASS-to-DEVS components, such as the façade, were also required for interaction with the IM and agent model. Interactions between the CCA did not require the IM as there was no disparity between the two CCA models' structure or timing. Behavioral differences were managed using the network mapping functions of the two CCA to transport and manipulate data to and from the associated cell components. Overall, the purpose of this figure is to illustrate the application of the CCA within a research project and highlight some of the complexities that may arise in implementation due to the need for mapping functions.

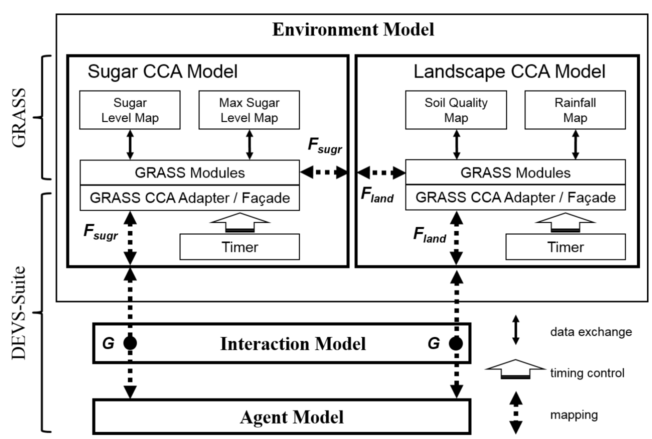

Figure 3. Hybrid Agent-Environment Model Showing Mapped CCAs.

The utility of the information in Section 2. lies in expanding the flexibility of the Composable Cellular Automata specification. The introduction of the CCA specification was necessitated by the need for a cellular automata formalism that could be composed with other, non-CA models to create a robust hybrid model. Continued use of the CCA specification has demonstrated the benefit of using multiple CCA in a hybrid model or making reuse of existing CCA. Building upon the CCA specification, composition of multiple CCA must be rigorously defined to ensure that the formalism is adhered to and that the implemented models and simulation results remain valid.

Fig. 4 illustrates three CCA that may be used in a hybrid model. The basic specification requires that the three CCA interact through their network mapping functions. If a non-CCA subsystem model were added, then additional mapping functions would be required between it and each of the CCA with which it interacts. Consider an example where components of a system are modeled and simulated independently at first. Then, it is desired to reuse the existing models to model more complete picture of the system. In this case, the continued use of mapping functions becomes burdensome and requires extra overhead; especially in cases demonstrated in Fig. 3 where extra components are required to implement CCA with environments such as GRASS.

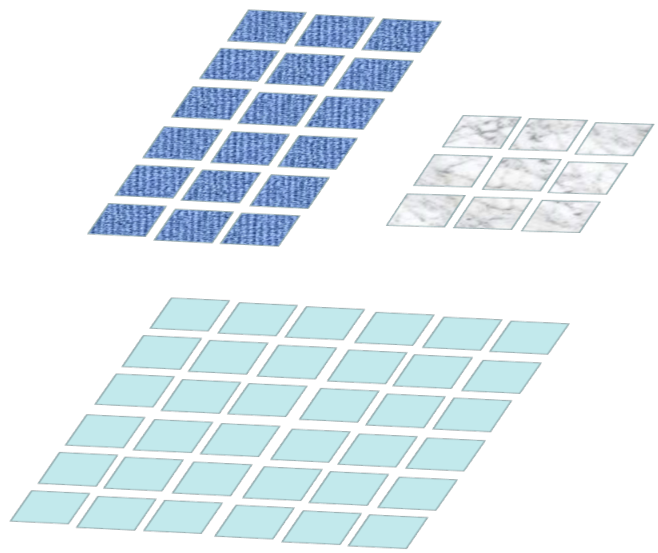

Figure 4. Exemplar Model Showing Separate CCAs That May Be Mapped.

The closure properties discussed here provide similar benefit as those received from DEVS models. Closure under coupling in DEVS allows a hierarchy of coupled and atomic models to be built, wherein the modeler may treat the topmost coupled model as a single atomic model and therefore, the hierarchy as a single system. Fig. 5 illustrates the three CCA from Fig. 4 arbitrarily composed. How they are organized to interact is defined by the mapping composition set and influencer composition set. For the purposes of this example, how exactly these subsystem models interact with one another is unimportant. What is important is that these cell components can now be imagined to be surrounded by a single network layer. That one network will now contain all of the mapping functions to and from the composed set of CCA cell components. Thus, all external systems can now interact with a single network component that encapsulates the CCA cell components. In implementation environments like those shown in Fig. 3, this can significantly reduce the complexity of the simulation model system.

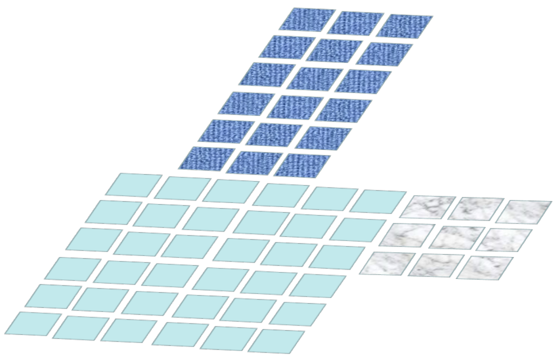

Figure 5. Exemplar Model Showing Composed CCAs That May Be Treated As One.

4. CONCLUSION

With adherence to specification and an understanding of the dependencies between tuple elements, composable cellular automata (CCA) can be directly composed with each other. CCA are defined as a network sextuple, which contain cellular automata (cells) defined by a septuple of elements. Composition of CCA is defined as a disjoint union of the tuple elements, with the addition of a mapping composition set and a influencer composition set that can redefine the external input/output and stitch two CCA together, respectively. The properties of set unions are therefore used as a foundation for composition approaches.

The paper discusses CCA composition from the perspective of dimensional properties—cell indices and time—for homogeneous CCA. It discusses how two CCA that differ in their index sets can be composed to create a new CCA. This is valid for compositions containing mapping composition sets, influencer compositions sets, or containing or lacking both. The paper also described why the composition exhibits an associate property such that multiple CCA can be composed in any order. Thus, the CCA specification exhibits a closure under composition. Lastly, two CCA that differ only in their discrete-time segments may also be composed, yielding a CCA containing all discrete-time segments from both original CCA.

The reader is cautioned that proper composition ensures that the system results can be verified to conform to specification. It does not assure validation with respect to the domain. Even if the original subsystems are valid, the composed system cannot be assumed to be valid. For example, changing the tessellation that represents the domain space or adding additional cell influencers may impact the overall simulation results. Similarly, while valid results may be obtained with CCA using two different sets of discrete-time segments, merging the two may not yield valid results. Just as a software developer should test the integration of two correct software components, a software modeler should validate the resultant composed model.

4.1. Future Work

The Composable Cellular Automata specification provides encapsulation mechanisms and strict adherence to loose coupling between CCA components. It emphasizes information hiding in that cells are not cognizant of the network structure and external entities must use the netowrk mapping functions to interact with the contained cell components. As such, it should be possible to extend the specification to include inhomogeneous cellular automata. For example, how would heterogeneous CCA that with different definitions of a neighborhood interact? What does it mean for CCA that are disparate in their state transition functions to be coupled? Developing answers to these questions require a relaxation of the basic specifications assumption of homogeneous cell components, M_{ijk}. All structural and behavioral inconsistencies will have to be formally examined to ensure model and simulation result correctness.

REFERENCES

[1] G. R. Mayer. *Composing Hybrid Discrete Event System and Cellular Automata Models*. PhD thesis, Arizona State University, 2009.

[2] H. S. Sarjoughian, G. R. Mayer, I.I. Ullah, and C.M. Barton. Managing hybrid model composition complexity: Human-environment simulation models. In L. Yilmaz, editor, *Concepts and Methodologies for Modeling and Simulation: A Tribute to Tuncer Ören*, chapter 6, pages 107–134. Springer International Publishing, Switzerland, 2015.

[3] IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules. *IEEE Std. 1516-2010*, 2010.

[4] H. S. Sarjoughian. Model composability. In L. F. Perrone, F. P. Wieland, J. Liu, G. Lawson, M. Nicol, and R. M. Fujimoto, editors, *Proceedings of the 2006 Winter Simulation Conference (WinterSim '06)*, pages 149–158, Monterey, California, December 2006. IEEE Press.

[5] H. S. Sarjoughian and G. R. Mayer. Heterogeneous model composability. In G. A. Wainer and P. J. Mosterman, editors, *Discrete-Event Modeling and Simulation: Theory and Applications*, chapter 5, pages 111–137. CRC Press, Boca Raton, FL, 2011.

[6] G. R. Mayer and H. S. Sarjoughian. Composable cellular automata. *Simulation: Transactions of the Society for Modeling and Simulation International*, 85:735–749, 2009.

[7] C. M. Barton, I. I. Ullah, and Bergin. Landuse, water, and mediterranean landscape: modelling long-term dynamics of complex socio-ecological systems. *Philosophical Transactions of the Royal Scoeity A*, 368:5275–5297, 2010.

[8] ACIMS. DEVS-Suite simulator, 2015. http://sourceforge.net/projects/devs-suitesim/ (accessed March 2015).

[9] GRASS. Geographic Resources Analysis Support System, 2015. http://grass.osgeo.org/ (accessed March 2015).

Biography

Dr. Gary Mayer has been a faculty member in the Department of Computer Science at Southern Illinois University Edwardsville since 2009. His research activities have included modeling and simulation, robotics, and educational robotics. His graduate degrees are in computer science and his undergraduate education is in aeronautical engineering. Prior to becoming a professor, Dr. Mayer served as an active duty US Air Force officer with qualifications in both program management and aeronautical engineering.