Holonic Cellular Automata:
Modelling Multi-level Self-organisation of Structure and Behaviour

Ada Diaconescu[1], Sven Tomforde[2] and Christian Müller-Schloer[3]

[1]Telecom ParisTech, LTCI, Paris, FR
[2]Kassel University, Kassel, DE
[3]Leibniz University, Hanover, DE
ada.diaconescu@telecom-paristech.fr

Abstract

Complex organisms, such as multi-cellular ones, have neither emerged spontaneously, nor evolved directly, from a disorganised mass of quarks. Stable intermediary sub-systems, like atoms and uni-cellular organisms, had to occur first and serve as reusable blocks for more complex systems to build upon. The occurrence of *structured* systems, featuring internal *diversity*, from uniform self-adaptive sub-systems is a key phenomenon to study in this context. We believe this phenomenon relies on the interactions among self-adaptive sub-systems, both at the *micro*-level (directly between sub-systems) but most importantly via *macro*-levels (indirectly via aggregate information and control from/to all sub-systems). To study this, we have developed a hierarchical control simulator based on self-adaptive cellular automata (CA). This paper presents our Holonic Cellular Automata (HCA) simulator, and the preliminary results showing the occurrence of structure / diversity from micro-macro feedback loops among self-adaptive CAs starting in the same states. This provides a promising basis for further investigations into the range of possibilities concerning structure creation, as a key enabler for the emergence of complex systems.

1 Introduction

Living organisms, especially multicellular ones, cannot be understood if studied as large collections of self-organising quarks – e.g. Simon (1962), Reilly and Ingber (2018). Nor have they emerged spontaneously (or evolved directly) from disorganised quarks. Stable atoms had to first occur from sub-atomic particles, then form stable unicellular organisms, and only then could multi-cellular creatures occur.

Similarly, complex adaptive artificial systems, e.g. smart cities and power grids, are difficult to design as monolithic processes that self-assemble and evolve from huge collections of atomic resources (e.g. fine-grained algorithms). Intermediary sub-systems must be designed to self-assemble at smaller scales first; then provide building blocks for progressively more complex systems, offering wider functions– e.g. Simon (1962), Powers (2008), op Akkerhuis (2010).

Morphogenetic Engineering, Doursat et al. (2012), emphasises the key role of *structure* and *diversity* in self-organising complex systems – e.g. developing an anthill rather than a sand dune; an animal rather than a cauliflower; a human society rather than a school of fish; a smart home rather than an agent system playing prisoner's dilemma.

We aim to study how structured heterogeneous systems can occur from uniform self-adaptive sub-systems; and how this process can be engineered and controlled. We believe that *multi-level feedback control* is key to such developments, by shaping various *interactions* among self-adaptive sub-systems, both at the *micro*-level (directly between sub-systems) and most importantly via *macro*-levels (via aggregate information and control from/to all sub-systems).

We developed a hierarchical control simulator based on cellular automata (CA) with adaptive rules. This was based on our theoretical work, Diaconescu et al. (2016), on key properties for engineering holonic systems (i.e. recursively self-encapsulated hierarchies). The presented Holonic Cellular Automata (HCA) simulator organises CA into several levels (sec. 4), which interact via: i) *aggregate state* information (bottom-up); and ii) *adaptation control* signals (top-down). CA at different levels execute *different rule sets*, at *different paces*. Each CA at a lower level L_m is mapped to a single cell of a CA at a higher level L_{m+1}. The entire state of a lower CA is aggregated (based on the percentage of its live cells relative to a threshold) and used to set the state of the corresponding cell in a higher CA (live or dead). Conversely, the state of each cell in a higher CA controls the rule adaptation of the corresponding lower CA. These interactions are replicated between successive levels, up to the topmost level which only executes static rules. CA execute in parallel, with aggregate states and adaptation control travelling bottom-up and top-down through the HCA levels.

The simulator resembles *hierarchical CA* previously used for modelling complex systems (sec. 3). The main difference is in transforming mere bottom-up data abstractions into complete feedback loops between levels, leading to multi-level control and self-adaptation. With respect to the multi-level model categorisation in Uhrmacher et al. (2005), the proposed HCA simulator relies on a discrete, deterministic and qualitative model; with heterogeneous behaviours, and upward and downward causation between levels. Preliminary results show how structure / diversity of CA states can occur and develop based on such inter-level feedbacks (micro-macro) among self-adaptive CA. Resulting structures depend on the CA rule sets and on several configuration parameters – e.g. aggregate state *thresholds* and relative *execution speeds*, at all levels.

The main contributions of this paper include[1]:

- highlight reusable engineering principles for developing complex systems via multi-level adaptive control;
- propose a multi-level adaptive control simulator, based on Holonic Cellular Automata (HCA);
- show encouraging preliminary results supporting both the engineering principles' viability and the simulator's usefulness as an experimental platform for studying them.

This provides a promising basis for further investigations into the processes leading to structure creation.

2 Holonic Structure Concepts

We use the term 'structure' in a twofold manner. Firstly, single-level *state structure* refers to the differentiation of states of sub-systems (within one level). In the HCA, this occurs when different CA groups at the bottom level L_0 go through different states (within identical state spaces). Secondly, *multi-level structure* refers to the differentiation of interrelations among sub-systems. In the HCA, these are concrete CA levels, with higher CA computing aggregates of lower CA states (Fig. 1). Multi-levels can be implemented either via actual sub-systems that represent different levels (explicit levels), or as mere conceptual abstractions (implicit levels). In the former case (explicit), higher systems (*supra-systems*) aggregate data from, and send control signals to, lower systems (*sub-systems*). E.g., in neural networks, actual neurons at higher levels (central) monitor and control neurons at lower levels (somatic) – Holst and Mittelstaedt (1950), Kramer et al. (1981), Diaconescu et al. (2018). In the latter case (implicit), supra-systems are mere abstractions, representing processes of data aggregation and adaptation control from and to sub-systems. E.g. opinion formation in social networks; or pheromone traces in ant colonies.

The main properties common to both kinds of multi-level structures are (Diaconescu et al. (2016)): i) data *aggregation* (bottom-up), with information loss; ii) specific *processing* of data aggregates (optional); iii) feedback *control* (top-down), reacting to aggregates; iv) *different paces* of cycles – aggregation, process and control – at different levels. We refer to such systems as 'holonic' – Simon (1962), Koestler (1967).

The HCA simulator features built-in multi-level structure (explicit), where the number of levels, CAs per level and level configurations can be varied. The aim is to study the

[1]Please see **https://github.com/adadiaconescu/hca/wiki** for further documentation, simulation videos and experimental data.

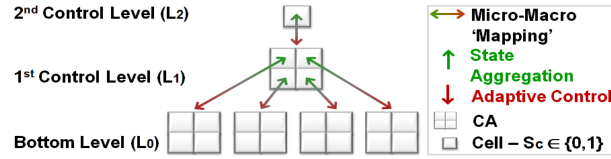


Figure 1: HCA Multi-level Structure (example for 3 levels)

formation of micro-level state structures, depending on such variations. Future work can also study the formation of upper levels and inter-level feedbacks, rather than fixing them.

3 Related Work

We focus on related work from two main research areas: multi-scale control systems (theoretical) and hierarchical cellular automata (modelling and simulation).

On the theoretical side, several research works modelled complex systems (e.g. organisms) as hierarchies of self-adaptive self-organising sub-systems, based on feedback controls – e.g. Simon (1962), Koestler (1967), Simon (1996), McGregor and Fernando (2005), Powers (2008), Flack (2017), Reilly and Ingber (2018). *Information abstraction* is a key feature of *upward causation* – e.g. in McGregor and Fernando (2005) via *redescriptions* of lower levels for higher levels; or in Flack (2017), via collective *coarse-graining*. *Downward causation* Flack (2017) was also identified as phenomena governing sub-system adaptations, based on collectively-computed macro-states.

Merging these two principles – *upward abstraction* and *downward causation* – leads to multi-level control loops, e.g. as promoted by Hierarchical Perceptual Control Theory (HPCT) Powers (2008) (for nervous systems): *"each perceptual signal at one level in the hierarchy is a function of multiple perceptions at a lower level. Control of a perception at one level requires adjustment of reference signals sent to lower systems, which control the perceptions on which the state of the higher-level perception depends."* Our HCA simulator implements this, for studying how multi-level control can produce state differentiation (structure).

On the modelling side, Hierarchical Cellular Automata Dunn (2010) interconnect multi-level CA for simulating complex phenomena as interrelated processes, at multiple scales – e.g. Adamides et al. (1992) for large-scale chip integration; Weimar (2001) for catalytic surface reactions; Dunn (2010) for landscape ecology; Dascalu et al. (2011) for eplidemiology; or Qin et al. (2018) for visual saliency. CA at sub-levels are coupled to CA at supra-levels via *abstraction functions*, and in some cases supra-CA are also coupled to sub-CA (e.g. Weimar (2001)). The key difference in our case is that the HCA's macro-micro couplings are control signals for rule adaptations (downward causation).

Multi-level models have also been proposed based on the multi-agent paradigm, to analyse existing complex systems

– e.g. in systems biology, Montagna and Omicini (2017). Our aim is to offer a generic simulator for studying multi-level phenomena and help identify key design principles.

4 Holonic Cellular Automata (HCA)

4.1 Overview and Notation

A Holonic Cellular Automaton (HCA) consists of several *levels* (L_k), each containg one or several CA ($CA_{k,i}$). Table 1 summarises the main HCA concepts and notations.

CA at adjacent HCA levels exchange two kinds of information (subsec. 4.2). Firstly, bottom-up communication transmits *aggregate states* ($O_{k,i}$) of sub-CA to set the *cell states* of supra-CA ($CS_{k+1,j,i}$). Secondly, top-down communication transmits the *cell states* of supra-CA as control signals (or *goals* $G_{k,i} = SC_{k+1,j,i}$) to sub-CA, which adapt their active rules ($R_{k,i}$). Inter-level communication relies on a predefined *mapping* – $map(CA_{k,i}; C_{k+1,j,i})$ – between each sub-CA ($CA_{k,i}$) and a cell of a supra-CA ($C_{k+1,j,i}$). Fig. 2 exemplifies a 3-level HCA – with bottom-up transfer of state aggregates ($O_{0,i}$ from L_0 to L_1; and $O_{1,1}$ from L_1 to L_2); and top-down transfer of control goals (G_1 from L_2 to L_1; and G_0 from L_1 to L_0). It also illustrates mappings between a CA cell at L_2 and a CA at L_1 (orange); and between two cells at L_1 and two CA at L_0 (blue and yellow).

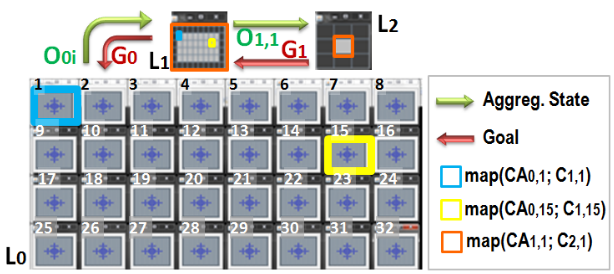


Figure 2: Exp1: Initialising 3-level HCA, $SCA_{1,1} = Null$

When started, an HCA is executed in *cycles*, each cycle triggering the sequential activation of adjacent HCA levels. When a level is activated, all CA at this level are executed in parallel (within one simulation *step*). Each CA in an active level: i) exchanges information with its mapped CA at the supra- and sub-levels (as above); ii) sets its rules ($R_{k,i}$) depending on its goals (cf. 4.2); and steps (executes its rules). When all CAs have finished executing, the level is deactivated and the next upper level is activated (cf. 4.2). Each cycle starts by activating the bottom level (L_0) and finishes by activating the top level (L_M); after that, the cycle restarts.

In the presented experiments, all CA at the bottom level start in the same state, then step in parallel; for experimental repeatability reasons, they synchronise (wait for each other) between steps. Still, the HCA supports starting CA at sequential steps (hence differentiating initial states) and running CAs asynchronously. All CA are non-toroidal (live boundaries). This helps to break symmetry at higher levels, yet it can be a reasonable assumption (e.g. environmental differentiation) and can be changed in future experiments.

Notation	Description
L_k	Level k, with $k = 1..M$, M the number of HCA levels
$CA_{k,i}$	Cellular Automata i at level L_k, $i = 1..N_k$, N_k the no. of CA at L_k
$size(CA_{k,i})$	Size of $CA_{k,i}$ in nr. of cells, $size(CA_{k,i}) = Sz_{k,i} = X_{k,i} \times Y_{k,i}$, $X_{k,i} \equiv$ width, $Y_{k,i} \equiv$ height
$C_{k,i,s}$	Cell s of $CA_{k,i}$, $s = 1..Sz_{k,i}$
$SC_{k,i,s}$	State of Cell $C_{k,i,s}$, $SC_{k,i,s} \in \{0, 1\}$, $0 \equiv$ false/dead, $1 \equiv$ true/live
$O_{k,i}$	Aggregate State of $CA_{k,i}$
Th_k	Threshold for calculating Aggregate States of all $CA_{k,i}$ at L_k
$G_{k,i}$	Goal of $CA_{k,i}$; $G_{k,i} \in \{0, 1\}$
$R_{k,i}$	Active Rules of automaton $CA_{k,i}$
$map(C_{k,i,s}; CA_{k-1,j})$	Mapping between cell $C_{k,i,s}$ and automaton $CA_{k-1,j}$; implies bottom-up transfer of aggregate state and top-down transfer of goal (subsec. 4.2)
$SCA_{k,i}$	Macro-State of $CA_{k,i}$ (set of states of all its cells); $SCA_{k,i} = \{SC_{k,i,s} \mid \forall\ s = 1..Sz_{k,i}\ \}$
$StpM_k$	Step Multiplier of level L_k– the number of activations of L_k after which $CA_{k,i}$ actually execute $R_{k,i}$.

Table 1: Main HCA Concepts and Notations

4.2 Inter-level Mapping and Communication

Aggregate states are transferred between sub- and supra-levels as shown in Eq. 1. The top level (L_M) is not concerned by this transfer. To simplify, we only used one CA at L_1 and L_2 in our experiments: each CA_0 maps to one cell at CA_1; CA_1 maps to the one cell of CA_2.

$$SC_{k+1,i} \leftarrow O_{k,i}\ ,\quad i = 1..N_k\ ,\ map(C_{k+1,i}; CA_{k,i}) \quad (1)$$

A CA's aggregate state is calculated based on the CA's number of live cells relative to a threshold (Eq. 2).

$$O_{k,i} = \begin{cases} 1, & \text{if } \sum_{s=1}^{S_{k,i}} SC_{k,i,s} >= Th_k \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Goals are sent from supra- to sub-levels as in Eq. 3. The bottom level (L_0) is not concerned by this transfer.

$$G_{k,i} \leftarrow SC_{k+1,i},\quad map(C_{k+1,i}; CA_{k,i}) \quad (3)$$

4.3 HCA Stepping Cycle

Several schemes are possible for activating HCA levels. Presented experiments were based on a sequential bottom-up stepping cycle (Fig. 3, for a 3-level HCA) – going from the bottom level through the middle level(s) up to the top level, then restarting. Algorithm 1 defines the procedure that an active level executes. When the experiment starts, all CA are set to initial states and L_0 is activated. We detail the stepping sequence below, for 3 levels (extensible to M).

When L_0 is activated, it checks if its current step index allows it to run (depending on its step multiplier $StpM_0$). If so, then $CA_{0,i}$ get their goals ($G_{0,i}$) from L_1 and adapt their rules accordingly (cf. 4.4); execute the rules; and calculate their state aggregates ($O_{0,i}$, Eq. 2). L_0 then activates L_1 and deactivates itself. When activated, L_1 checks if its step index allows it to execute. If so, then $CA_{1,j}$ get their cell states from the aggregates at L_0 (Eq. 1); gets their goals ($G_{1,j}$) from the cell states of CA at L_2 (Eq. 3); and adapt their rules accordingly (cf. 4.4). They then execute their rules and calculate aggregates ($O_{1,1}$). L_1 then activates L_2 and deactivates itself. When L_2 is activated, if its step index allows it to execute, then $CA_{2,i}$ get their cells' states from the aggregates of CA at L_1, and execute their static rules. L_2 then activates L_0 and deactivates itself. The cycle restarts.

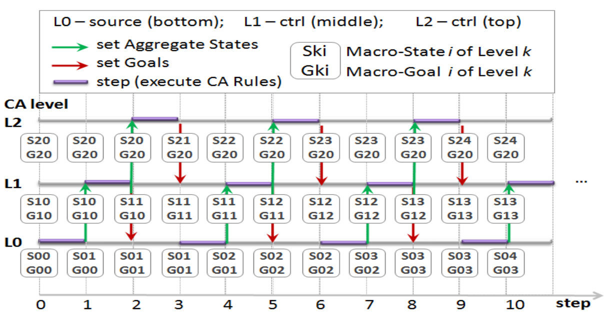


Figure 3: Stepping Cycle between CA Levels

4.4 CA Rules

Different CA rules operate at different HCA levels. CA rules calculate each cell's next state based on its current state and the state of its *four* neighbours (top, down, left, right). We use two kinds of rules: *adaptive* (for bottom and middle levels) and *static* (for the top level). Adaptive rules swap between two sets of actual CA rules: i) *expansive* (R_{Exp}) – increasing the number of a CA's live cells; and ii) *regressive* (R_{Reg}) – decreasing the number of a CA's live cells. A CA controlled by adaptive rules activates R_{Exp} if its goal is 1 and activates R_{Reg} if its goal is 0 (Eq. 4).

$$R_{k,i,active} = \begin{cases} R_{k,Exp} & \text{if } G_{k,i} == 1 \\ R_{k,Reg}, & \text{if } G_{k,i} == 0 \end{cases} \tag{4}$$

Algorithm 1 CA Stepping at Level L_k

procedure LEVEL INIT PROCEDURE (L_K)
  $stepIndex_k \leftarrow 0$
  for $CA_{ki} \in L_k$ **do**
    $SCA_{k,i} \leftarrow$ *init state* ▷ init. all CA states
  while $L_k active == true$ **do**
    for $CA_{k,i} \in L_k$ **do**
      Execute **CA Step Procedure** for CA_{ki}
    $stepIndex_k \leftarrow stepIndex_k + 1$
    if $k == M$ **then** ▷ if at top, restart from L_0
      $k = -1$
    $L_{k+1}active ==$ *true* ▷ activate next level up
    $L_k active == false$ ▷ deactivate this level
procedure CA STEP PROCEDURE ($CA_{k,i}$)
  if $stepIndex_k \neq StpM_k$ **then**
    exit procedure
  if $isBottomLevel ==$ *false* **then** ▷ get aggregates
    $SCA_{k,i} \leftarrow$ *Aggregate States from sub-CA*
  if $isTopLevel ==$ *false* **then** ▷ get goals
    $Goal_{k,i} \leftarrow$ *Goals from State of supra-CA*
  if $Goal_{k,i} ==$ *true* **then** ▷ adapt CA rules
    $Rules_{active} \leftarrow R_{exp}$
  else
    $Rules_{active} \leftarrow R_{reg}$
  execute $Rules_{active}$
  calculate *Aggregate State of CA (for supra-CA)*

Experiments were run on a 3-level HCA, with L_0 using adaptive rules $R_{0,Exp}$ and $R_{0,Reg}$ (Fig. 4); L_1 using adaptive rules $R_{1,Rxp}$ and $R_{1,Reg}$ (Fig. 6); and L_2 using static rules $R_{2,Inv}$. Fig. 5 illustrates the behaviour of $R_{0,Exp}$ and $R_{0,Reg}$, for a CA of size 21x21cells, starting from an initial state of 5 live cells (central cross shape), and from a full board state, respectively. $R_{2,Inv}$, at L_2, simply inverses the current state of each cell: if $SC_{2,i,t} == 1$ (live) then $SC_{2,i,t+1} == 0$ (dead); else $C_{2,i,t+1} == 1$ (live).

R0,exp	Neighbour Count				
Current State	0	1	2	3	4
0	0	1	0	0	1
a 1	1	1	1	1	1

R0,reg	Neighbour Count				
Current State	0	1	2	3	4
0	0	0	0	0	0
b 1	0	0	0	1	1

Figure 4: Bottom Level (L_0) Rules: a) expand; b) regress

5 Experimental Results for a 3-Level HCA

5.1 Common Settings

We focus the presentation on the 3-level HCA used for experiments (Fig. 2), even if the concepts, notations and algorithms apply to HCA with any number of levels (M).

The bottom level L_0 consists of 32 CA – numbered from $CA_{0,1}$ to $CA_{0,32}$ – each of size 21x21 cells, arranged in an

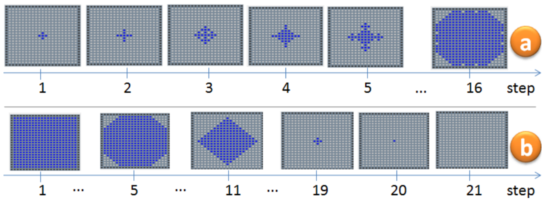


Figure 5: Behaviour of Rules at L_0: a) expand; b) regress

R1,exp	Neighbour Count				
Current State	0	1	2	3	4
0	0	1	1	1	1
a 1	1	1	1	1	1

R1,reg	Neighbour Count				
Current State	0	1	2	3	4
0	0	0	0	0	1
b 1	0	0	0	0	1

Figure 6: Middle level (L1) Rules: a) expand; b) regress

8x4 matrix (for ease of visual mapping to cells in $CA_{1,1}$). To help discuss the impact of goals from L_1 on $CA_{0,i}$'s states, we categorise CA at L_0 into three types (Fig. 7):

- $CA_{0,Corners}$: $CA_{0,i}$; with $i = \{1, 8, 25, 32\}$;
- $CA_{0,Border}$: $CA_{0,i}$; $i = [2..7] \wedge \{9, 16, 17, 24\} \wedge [26..31]$;
- $CA_{0,Core}$: $CA_{0,i}$; with $i = [10..15] \wedge [18..23]$.

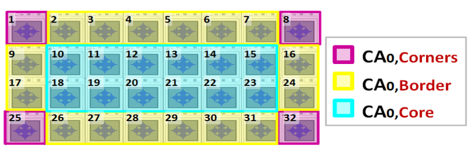


Figure 7: CA Types at the Bottom Level (CA_0)

The middle level (L_1) has a single CA ($CA_{1,1}$) of size 32 cells (8width x 4height) – with each cell mapped to a CA at L_0: map($CA_{0,i}$;$C_{1,1,i}$), $i = 1..32$. The top level (L_2) has a single CA ($CA_{2,1}$) of size 1 cell – mapped to the CA at L_1: map($CA_{1,1}$;$C_{2,1,1}$). The goals of L_0 and L_1 are initialised $G_0 = G_1 = 1$ (meaning that $R_{0,Exp}$ and $R_{1,Exp}$ are active). Goals are irrelevant for L_2, which always uses $R_{2,Inv}$.

The following parameters were selected (more or less) arbitrarily for the presented simulations but can be varied for further experiments (future work): the size and number of CAs at L_0, the expanding/regressive rules and initial states at each level, non-toroidal CAs and boundary conditions.

5.2 General Behavioural Analysis

The above settings lead to four possible states at $CA_{1,1}$ (Table 2) – $Null$ (all cells dead), $Full$ (all cells alive), $Core$ (only cells with 4 neighbours are alive) and $Cross$ (only cells with 3 or 4 neighbours are alive). The state transition scheme depends on further configurations (i.e. Th_k and $StpM_k$). State dynamics at the middle level (L_1) are key to

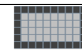
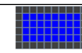
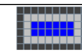
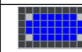
$CA_{1,1}$ State ID	$Null$	$Full$	$Core$	$Cross$
$CA_{1,1}$ State				
No. Live Cells	0	32	12	28
% Live Cells	0%	100%	37.5%	87.5%
$G_{0,Corners}$	0	1	0	0
$G_{0,Border}$	0	1	0	1
$G_{0,Core}$	0	1	1	1

Table 2: States of the Middle Level ($CA_{1,1}$)

the goal patterns ($G_{0,i}$) set at L_0, which drive state differentiation at L_0 ($SCA_{0,i}$) and hence the occurrence and dynamics of *macro-state structures*. They also drive the goals at L_1 ($G_{1,1}$), via aggregates $O_{1,1}$ sent to L_2.

The states of $CA_{1,1}$ (at L_1) set the goals of $CA_{0,i}$ (at L_0) (as in Table 2), and hence their active rules – $R_{0,Exp}$ or $R_{0,Reg}$. E.g., when $SCA_{1,1}$=$Null$, $CA_{0,i}$ receive $G_{0,i} = 0$ and set $R_{0,Active} = R_{0,Reg}$. Or, when $SCA_{1,1} = Cross$, then $CA_{0,Core}$ and $CA_{0,Border}$ get $G_0 = 1$ (activate $R_{0,Exp}$), while $CA_{0,Corners}$ get $G_0 = 0$ (activate $R_{0,Reg}$).

Hence, $CA_{1,1}$'s $Null$ and $Full$ states (at L_1) do *not* cause any differentiation at $CA_{0,i}$ (at L_0). Yet, importantly, $CA_{1,1}$'s $Core$ and $Cross$ states (at L_1) lead to $CA_{0,i}$'s state differentiation (at L_0) among its three groups – $CA_{0,Core}$, $CA_{0,Border}$ and $CA_{0,Corners}$. Different patterns of $Core$ and $Cross$ states at L_1 lead to various *macro-state structures* and dynamics at L_0. Each CA group at L_0 can converge to any of four state types: 1) *dead*: no live cells; 2) *alive stuck*: live cells, but no change; 3) *oscillating*: cycling through a state sequence; and 4) *chaotic*: following an aleatory state sequence. So far, we observed diverse combinations of the first three state types (subsec. 5.3 and 5.4).

Initial simulation steps are common to all experiments. In brief, $CA_{1,1}$ and $CA_{2,1}$ start in $Null$ state and CA_0, i in an initial state with 5 live cells (1.13% of 441 cells, below Th_0=10%, hence $O_{k,i}$=0). When $R_{0,Exp}$ produces enough live cells in $CA_{0,i}$ (bottom) to cross Th_0, then $O_{0,i}$=1 are sent to $CA_{1,1}$ (middle), which passes to $Full$. Hence, aggregate $O_{1,1}$=1 is sent to $CA_{2,1}$ (top), which also passes to $Full$. Rules $R_{2,Inv}$ (top) inverse $CA_{2,1}$'s state 1 (live) to 0 (dead), hence sending goal $G_1 = 0$ back to $CA_{1,1}$ (middle). $CA_{1,1}$ adapts its rules to $R_{1,Reg}$, executes them, and passes from $Full$ to $Core$. Hence, it sends G_0=1 to $CA_{0,Core}$; and G_0=0 to $CA_{0,Corners}$ and $CA_{0,Border}$ (bottom). The sequence from here depends on experimental settings.

The exact dynamics of the L_0 macro-state structures depends on: a) how fast (number of steps) R_{Exp} and R_{Reg} at L_0 and L_1 take aggregate states above and below the thresholds Th_0 and Th_1, respectively, from given states; b) the actual values of Th_0 and Th_1 relative to the CA board sizes; and, c) the relative differences in execution times at L_0, L_1 and L_2, based on $StpM_0$, $StpM_1$ and $StpM_2$, respectively.

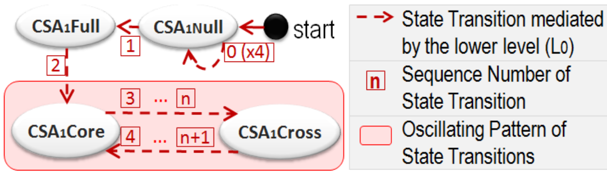


Figure 8: State Transitions of Middle level (L1)

5.3 Macro-States with One Oscillation

Experiment Configuration. Aggregate state thresholds were set to $Th_0 = 0.1$ (at L_0) and $Th_1 = 0.5$ (at L_1). Hence, $CA_{0,i}$ at L_0 must have more than 10% live cells to send a 'live' aggregates to L_1 ($O_{0,i}$=1); and $CA_{1,1}$ at L_1 over 50% live cells to send $O_{1,1}$=1 to L_2. Step multipliers were set to 1 for all levels ($StpM_0 = StpM_1 = StpM_2 = 1$), meaning that CA at all levels executed at each cycle.

State Transitions at L_1 and L_2. Fig. 8 shows $CA_{1,1}$'s state transitions (at L_1). In short, $CA_{1,1}$ starts in $Null$ (for 4 steps); then passes through $Full$ (1 step) and $Core$ (1 step). From step 7, it oscillates between $Cross$ and $Core$ states for the rest of the experiment. These transitions set the goal patterns $G_{0,i}$ at L_0: for $CA_{0,Corners}$, G_0 changes from 1 to 0 at step 8, then remains unchanged; for $CA_{0,Border}$, G_0 changes from 1 to 0 at step 8, then oscillates between 0 and 1; and for $CA_{0,Core}$, G_0 remains unchanged at 1. Based on $CA_{1,1}$'s ensuing aggregates ($O_{1,1}$), G_1 changes from 1 to 0 at step 7, then oscillates between 0 and 1.

Macro-State Structure at L_0. The macro-state structures 'emerging' at L_0 are summarised in Tables 3 and 4. The HCA behaviour converges to: $CA_{0,Core}$ get stuck in a live state; surrounded by $CA_{0,Border}$ that oscillate between two states; and with the $CA_{0,Corners}$ in dead state.

Fig. 2 depicts the HCA in one of its 'starting-up' states, where $CA_{0,i}$ (at L_0) have not yet reached their first 'live' aggregates ($O_{0,i}$=1); and hence the HCA's inter-level feedbacks (via goal changes and rule adaptations) have not yet been triggered. Hence, all goals are set to 1, $CA_{1,1}$ and $CA_{2,1}$ are in $Null$, and do *not* yet send goals to sub-levels.

In Table 3, *Diversity Count* shows how many diverse states a CA creates in an experiment; *Final Behaviour* is the attractor state or oscillatory pattern to which the CA converges, after *1st Step of Final Behaviour*. E.g., $CA_{i,Corners}$ create 10 diverse states before dying off, at step 11.

Table 4 shows snapshots of the most important HCA states, after the inter-level feedbacks were triggered:

- (a) $CA_{0,i}$'s aggregates (bottom) cross Th_0=10% for the 1st time, sending $O_{0,i}$=1 to $CA_{1,1}$ (middle). This leads to $SCA_{1,1}$=$Full$ (middle), sending $G_{0,i}$=1 to L_0 (bottom), and $O_{1,1}$=1 (100% > Th_1=50%) to L_2 (top). This leads to $SCA_{2,1}$=$Full$ (top), which is inversed via $R_{2,Inv}$ to $SCA_{2,1}$=$Null$ (shown), hence sending G_1=0 to $CA_{1,1}$, which activates $R_{1,Reg}$;
- (b) $CA_{1,1}$ (L_1) runs $R_{1,Reg}$ and goes from $Full$ to $Core$. It sends to L_0: $G_{0,Core}$=1 and $G_{0,Border}$=$G_{0,Corners}$=0. It also causes $SCA_{2,1}$=$Null$ (37.5% < Th_1 = 50%), which is inversed via $R_{2,Inv}$ to $SCA_{2,1}$=$Full$ (shown), sending G_1=1 to $CA_{1,1}$, which activates $R_{1,Exp}$;
- (c) $CA_{1,1}$ runs $R_{1,Exp}$ and goes from $Core$ to $Cross$. It sends to L_0: $G_{0,Core}$=$G_{0,Border}$=1 and $G_{0,Corners}$=0. It also causes $SCA_{2,1}$=$Full$ (87.5% > Th_1 = 50%), which is inversed via $R_{2,Inv}$ to $SCA_{2,1}$=$Null$ (shown), hence sending G_1=0 to $CA_{1,1}$, which activates $R_{1,Reg}$;
- (d) $CA_{1,1}$ runs $R_{1,Reg}$ and goes to $Core$ state; from here, it oscillates between $Cross$ (e) and $Core$ (d);
- (e) $CA_{1,1}$ runs $R_{1,Exp}$ and goes to $Cross$ state; from here, it oscillates between $Core$ (d) $Cross$ (e).

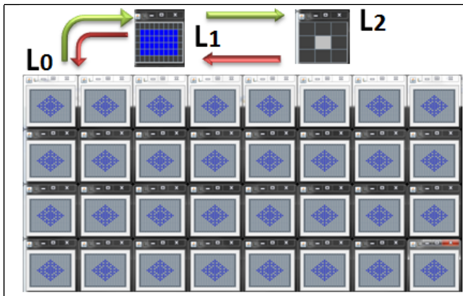
CA_0 Group	Diversity Count	Final Behaviour	1st Step of Final Bhvr.
$Corners$	10	Null	11
$Borders$	13	Oscil. (2 states)	12
$Core$	16	Stuck	16

Table 3: Exp1, Summary of HCA Convergence at L_0

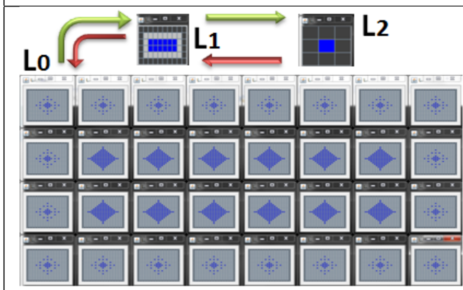
5.4 Macro-States with Two Oscillations

Experiment Configuration. Aggregated state thresholds were set to $Th_0 = 0.1$ (L_0) and $Th_1 = 0.9$ (L_1). Hence, $CA_{0,i}$ must have more than 10% live cells to produce a 'live' aggregate ($O_{0,i}$=1); and $CA_{1,1}$ over 90% live cells (for $O_{1,1}$=1). Step multipliers were set to $StpM_0 = StpM_1 = 1$ and $StpM_2 = 2$. This means that CA at L_0 and L_1 execute at every cycle, while CA at L_2 only once every two cycles.

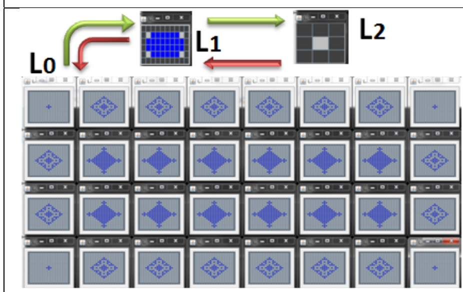
State Transitions at L_1 and L_2. Fig. 9 shows $CA_{1,1}$'s state transitions. In short (as in 5.3), $CA_{1,1}$ starts in $Null$ (for 4 steps), then goes to $Full$ (1 step) and $Core$ (1 step). However, starting with step 7, it cycles through states $Null$ (1step), $Cross$ (5steps), $Full$ (1step) and $Core$ (1step). These transitions set the dynamics of goals at L_0: for $CA_{0,Corners}$, G_0 changes from 1 to 0 at step 8, then oscillates between 0 (8 steps) and 1 (1 step); for $CA_{0,Border}$, G_0 changes from 1 to 0 at step 8, then oscillates between 0 (2 steps) and 1 (6 steps); and for $CA_{0,Core}$, G_0 changes from 1 to 0 at step 9, then oscillates between 0 (1 step) and 1 (7 steps). Based on $CA_{1,1}$'s ensuing aggregates ($O_{1,1}$), G_1 changes from 1 to 0 at step 7, then oscillates between 0 (2 steps) and 1 (6 steps). This is because only $Full$ state triggers $O_1 = 1$ (100% > $Th_1 = 90\%$) and hence $G_1 = 0$, but $CA_{2,1}$ only executes every 2 steps ($StpM_2 = 2$).



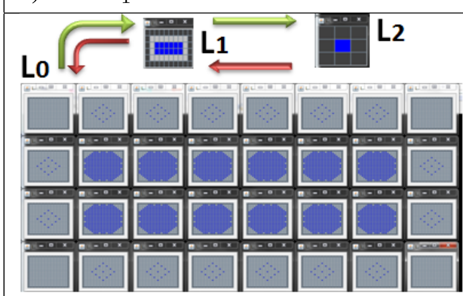
a) 1st $LS_1 = Full$ State



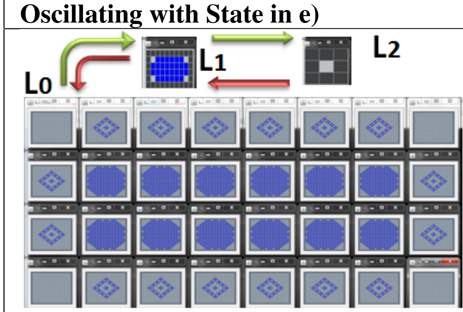
b) 1st $LS_1 = Core$ State



c) 1st $LS_1 = Cross$ State



d) Final $LS_1 = Core$ State, Oscillating with State in e)



e) Final $LS_1 = Cross$ State, Oscillating with State in d)

Table 4: Exp1, HCA's Macro-State Structures at L_0

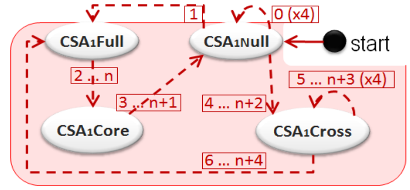
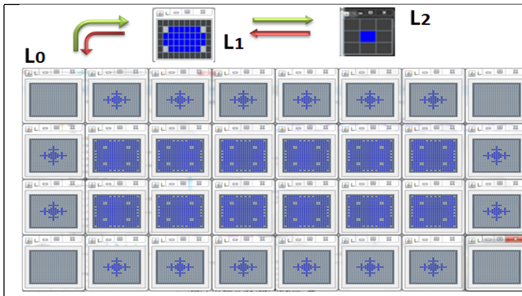


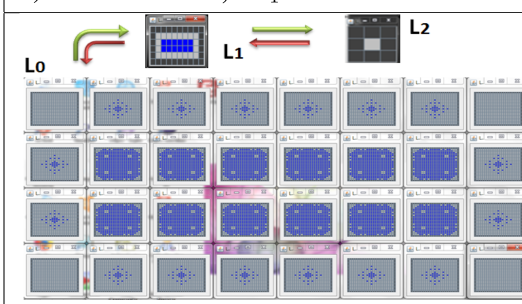
Figure 9: Exp2, State Transitions of Middle Level (L_1)

CA_0 **Position**	**Divers. Count**	**Final Behaviour**	**1st Step of Final Bhvr.**
$Corners$	10	Null	11
$Borders$	8	Oscil-1 (8 states)	8
$Core$	43	Oscil-2 (12 states)	67

Table 5: Exp2, Summary of HCA Convergence at L_0



a) 2 x Oscillations, $LS_1 = Cross$ State



b) 2 x Oscillations, $LS_1 = Core$ State

Table 6: Exp2, HCA's Macro-State Structures at L_0

Macro Structure at L_0. The macro-state structures forming at L_0 are summarised in Tables 5 and 6. Interestingly, L_0 converges to a behaviour where two CA_0 groups – $Core$ and $Border$ – oscillate through different state cycles. Namely (Table 5): $CA_{0,Core}$ oscillate through 12 states, in cycles of 93 steps; Border $CA_{0,Border}$ oscillate through 8 states, in cycles of 8 steps; and $CA_{0,Corners}$ die off. Table 6 shows two snapshots exemplifying HCA's final behaviour, each time with $CA_{0,Border}$ and $CA_{0,Core}$ in different states.

5.5 Summary of Other Experiments

Table 7 summarises further selected experimental results.

Configuration				Group Convergence		
Th_0	Th_1	$StpM_1$	$StpM_2$	*Core*	*Border*	*Corner*
0.1	0.5	1	2 .. 4	dead	dead	dead
0.1	0.5	2	3	stuck	oscil	dead
0.1	0.5	2	4	oscil	dead	dead
0.1	0.5	4	4	stuck	oscil-1	oscil-1
0.1	0.9	1	3	oscil-1	oscil-2	dead
0.1	0.9	1	4 .. 5	stuck	dead	dead
0.1	0.9	1	6	dead	dead	dead
0.1	0.9	2	2 .. 4	stuck	oscil	dead
0.1	0.9	4	4	stuck	oscil-1	oscil-1
0.1	0.9	2 .. 3	6	oscil-1	oscil-2	dead
0.1	0.9	4	6	stuck	oscil-1	oscil-1

Table 7: Summary of Experiments, HCA Convergence at L_0

6 Conclusions and Future Work

We presented a Holonic Cellular Automata (HCA) simulator for multi-level adaptive control systems. The aim is to offer a generic tool for studying the impact of inter-level feedbacks on complex system behaviour, focusing on the formation of macro-state structures at the micro-level.

The main contributions of this paper include:

- highlighting *engineering principles* for developing artificial complex systems via multi-level control structures: i) micro-macro state aggregation, with information loss; ii) macro-level processing of state aggregates (optional); iii) macro-micro adaptation control signals; and, iv) different execution times for feedbacks at different levels.

- proposing a *multi-level control system simulator*, based on Holonic Cellular Automata (HCA) – CA hierarchy featuring the principles above. HCA helps study these principles, showing the impacts of key parameters (e.g. aggregates calculation or execution times) on the formation of macro-structures and behaviours (e.g. static or cyclic).

- showing *encouraging preliminary results* supporting both the viability of the engineering principles and the usefulness of the simulator for further studying them.

On the long term, the purpose of our research is two-fold: i) to thoroughly understand the essential principles behind the apparent success of multi-level/holonic structures in natural systems (Simon (1996)) ; and, ii) to translate these principles into reusable engineering artefacts to help us design, develop and maintain complex artificial systems, such as artificial life and (socio-)cyber-physical systems.

References

Adamides, E., Tsalides, P., and Thanailakis, A. (1992). Hierarchical cellular automata structures. *Parallel Computing*, 18(5):517 – 524.

Dascalu, M., Stefan, G., Zafiu, A., and Plavitu, A. (2011). Applications of multilevel cellular automata in epidemiology. In *Automatic Control, Modelling & Simulation*, pages 439–444.

Diaconescu, A. et al. (2016). Goal-oriented holonics for complex system (self-)integration: Concepts and case studies. In *IEEE Intl Cnf SASO 2016, Augsburg, Germany*, pages 100–109.

Diaconescu, A., Mata, P., and Bellman, K. (2018). Self-integrating organic control systems: from crayfish to smart homes. In *Intl Wrksp SAOS'18, Braunschweig, Germany*.

Doursat, R., Sayama, H., and Michel, O. (2012). *Morphogenetic Engineering*. Springer.

Dunn, A. (2010). *Hierarchical Cellular Automata Methods*, pages 59–80. Springer Berlin Heidelberg, Berlin, Heidelberg.

Flack, J. C. (2017). Coarse-graining as a downward causation mechanism. *Philosophical Trans. of Royal Soc. of London A: Mathematical, Physical and Eeng. Sciences*, 375(2109).

Holst, E. V. and Mittelstaedt, H. (1950). The principle of reafference: Interactions between the central nervous system and the peripheral organs. *Naturwissenschaften*, 37:464–476.

Koestler, A. (1967). *The Ghost in the Machine*. GATEWAY EDITIONS, Henry Regnery Co., 1 edition.

Kramer, A., Krasne, F., and Bellman, K. (1981). Different command neurons select different outputs from a shared premotor interneuron of crayfish tail-flip circuitry. *Science*, 214(4522):810–812.

McGregor, S. and Fernando, C. (2005). Levels of description: A novel approach to dynamical hierarchies. *ALife*, 11(4).

Montagna, S. and Omicini, A. (2017). Agent-based modelling in multicellular systems biology. In Adamatti, D. F., editor, *Multi-Agent Based Simulations Applied to Biological and Environmental Systems*, ACIR, pages 159–178. IGI Global.

op Akkerhuis, J. (2010). *The Operator Hierarchy. A chain of closures linking matter, life and artificial intelligence*. PhD thesis, Radboud University Nijmegen.

Powers, W. T. e. a. (2008). Perceptual control theory - a model for understanding the mechanisms and phenomena of control. In Forssell, editor, *Perceptual Control Theory: Science and Applications*, pages 18–34. Living Control Systems Pub.

Qin, Y., Feng, M., Lu, H., and Cottrell, G. W. (2018). Hierarchical cellular automata for visual saliency. *I. J. of Computer Vision*.

Reilly, C. B. and Ingber, D. E. (2018). Multi-scale modeling reveals use of hierarchical tensegrity principles at the molecular, multi-molecular, and cellular levels. *Extreme Mechanics Letters*, 20:21 – 28.

Simon, H. A. (1962). The architecture of complexity. *American Philosophical Society*, 106.

Simon, H. A. (1996). *The Sciences of the Artificial*. MIT Press.

Uhrmacher, A. M., Degenring, D., and Zeigler, B. (2005). *Discrete Event Multi-level Models for Systems Biology*, pages 66–89. Springer Berlin Heidelberg, Berlin, Heidelberg.

Weimar, J. R. (2001). Coupling microscopic and macroscopic cellular automata. *Parallel Computing*, 27(5):601 – 611. Cellular automata: From modeling to applications.