

# Emergence and Control of Macro Spatial Structures in Perturbed Cellular Automata, and Implications for Pervasive Computing Systems

Marco Mamei<sup>1</sup>, Andrea Roli<sup>2</sup>, Franco Zambonelli<sup>1</sup>

1) Dipartimento di Scienze e Metodi dell'Ingegneria – Università di Modena e Reggio Emilia  
Via Allegri 13 – Reggio Emilia– ITALY franco.zambonelli@unimo.it, mamei.marco@unimo.it

2) Dipartimento di Scienze – Università "G.D'Annunzio" Viale Pindaro 42 – Pescara – ITALY. a.roli@unich.it

## ABSTRACT

*Predicting the behavior of complex decentralized pervasive computing systems before their deployment in a dynamic environment, as well as being able to influence and control their behavior in a decentralized way, will be of fundamental importance in the near future. In this context, this paper describes the general behavior observed in a large set of asynchronous cellular automata (CA) when external perturbations (expressions of a dynamic environment) influence the internal activities of CA cells. In particular, we observed that stable macro-level spatial structures emerge from local interactions among cells, a behavior that does not emerge when cellular automata are not perturbed. Because perturbed cellular automata express characteristics that strongly resemble those of pervasive computing systems, similar sorts of macro-level behaviors are likely to emerge in that context and need to be studied, controlled, and possibly fruitfully exploited. On this basis, the paper also reports the results of a set of experiments showing how it is possible to control, in a decentralized way, the behavior of perturbed cellular automata, to make some desired patterns emerge.*

**Keywords:** Cellular Automata, Self-organization, Pervasive Computing, Multiagent Systems.

## 1 INTRODUCTION

Computing will soon become pervasive and autonomous. On the one hand, our everyday environments will be more and more populated by multitudes of decentralized and networked computing systems (e.g., multi-agent systems [ZamJW03], ad-hoc networks of mobile computing devices [Bor02], sensor networks [EstC02], clouds of “smart dust” [Pis01] and spray computers [Zam04]). On the other hand, most of these systems will be able to autonomously perform activities on our behalf – typically by interacting with each other – letting us “out of the loop” [Ten00, KepC03].

The potential applications of future pervasive computing scenarios are endless, promising to impact all our activities and to increase the quality of our life and work. For this reason, it is of dramatic importance to understand and predict

– possibly prior to their deployment – how such systems will behave. It has been recently discovered that the Internet, the Web, and the Gnutella network – the only deployed examples of large-scale decentralized and autonomous computing systems – have structurally evolved in rather peculiar and unexpected ways, strongly impacting on reachability of information and reliability [AlbJB99, AlbJB00, RipIF02]. Researchers in the area of pervasive computing should learn from this, and should start asking *now* whether similar situations may occur *in the future*, and which tools and methodologies to exploit to control them.

This paper is a little step in that direction. In particular, we present and discuss a number of experiments that we have performed on a large set of cellular automata (from now on CA) [Wol94, Bar97, Wol02], in which we have tried to mimic, in a minimalist way, the key characteristics likely to be exhibited by future pervasive computing systems:

- *Locality in interactions:* unlike the Web and the Gnutella network, whose interaction/structural patterns are not locally constrained, future pervasive computing systems will typically interoperate on the basis of local patterns, as deriving from, e.g., short-range wireless communications and/or local directed networks. Such characteristic is naturally reflected in CA, which evolve via local interactions in a lattice.
- *Autonomy of components:* unlike “traditional” computing systems and parallel applications, whose activities are typically driven by centralized flows of control and are typically subject to strict synchronization constraints, pervasive computing systems tend to be fully decentralized, with components executing in total autonomy. Such characteristics can be reflected by asynchronous CA, i.e., CA in which state transitions in cells occur asynchronously, according to local internal dynamics [IngB84].
- *Perturbation of the environment:* pervasive computing systems will be dived in and interact with dynamic environments, whether the physical world or some virtual computational world, that can continuously

influence the local activities of distributed components. Such characteristics can be emulated by perturbing the internal state of CA cells.

In a first set of experiments, we show that asynchronous CA, when perturbed, exhibit peculiar interesting behavior. In particular, during their dynamic evolution, and despite the out-of-equilibrium situation induced by perturbations of the external environment, stable macro-level spatial patterns may emerge from local interactions among cells, a behavior that does not emerge when the cellular automaton is synchronous and not perturbed. On this basis, the paper argues that macro-level patterns, similar to the ones exhibited by CA, will be observed, in terms of globally coordinated patterns of activity, as soon as multitudes of interacting computer-based systems will start populating our networks and our physical spaces. Such coordinated patterns of activity will be likely to dramatically influence the overall behavior of that systems at a very large scale.

A second set of experiments explores how and to which extent it is possible to exert some sort of decentralized control over the perturbed CA, so as to influence their evolution and make desired patterns emerge. To be potentially applicable to pervasive and decentralized computing scenarios, such a methodology must account for the impossibility to fully control the behavior of every components of the system and must instead acknowledge that only a fraction of them can be reached. Two methodologies of these kinds have been experienced, one of which with rather successful results. This lets us envision the possibility of actively controlling the emergence of macro-spatial patterns either defensively, to prevent their possible damaging effects on the system, or constructively, as a tool to enforce global coordination patterns.

The remainder of this paper is organized as follows. Section 2 introduces cellular automata, and characterizes asynchronous dynamics and perturbations. Section 3 presents and analyzes the behavior observed in perturbed cellular automata, e.g., the emergence of macro-level spatial structures. Section 4 discusses the implications of this emergent behavior for future pervasive computing systems. Section 5 presents the two methodologies we have experienced to control the behaviors emerging from a perturbed CA, and discuss how they may be useful to control the global behavior of decentralized pervasive computing systems. Section 6 discusses related work in the area. Section 7 concludes and outlines future work.

## 2 PERTURBED CELLULAR AUTOMATA

In this section, we will shortly introduce first the basic background concepts underlying cellular automata, then the specific class of asynchronous cellular automata, and finally the way to induce perturbations on cellular automata.

### 2.1 Basic Characteristics

Generally speaking, CA are regular lattices of cells, each one being a finite-state automaton. Starting from an initial global

state (determined by the local states of all its cells) a CA dynamically evolves by having cells update their local state depending on a (typically simple) state transition function of their state and of the state of neighboring cells.

More formally, a CA is statically defined by a quadruple

$$A = (S, d, N, f)$$

$S$  is the finite set of possible states a cell can assume.  $d$  is the dimension of the automaton, i.e. cells are organized into  $d$ -dimensional discrete grid (possibly with wraparound edges).  $N$  is the neighborhood structure, defining which cells can "influence" each cell in local state transitions.  $N$  is typically uniform and isotropic.  $f$  is the local transition rule, i.e., a function  $f: S^N \rightarrow S$  mapping a configuration of states in a neighborhood  $N$  into a single state.  $f$  is typically the same for each cell (uniform CA).

In this paper, we mostly focus on CA with binary cells ( $S = \{0, 1\}$ , two states that in our figure correspond to a cell being white or black, respectively), arranged in 2-dimensional square grids with wraparound borders ( $d=2$ ). In the following,  $N$  will simply indicate the number of neighbors of a cell. In any case, as we will discuss later on, our results apply to a larger class of CA (i.e., CA with different dimensions, larger state sets for cells, and in which cells are connected according to irregular lattices).

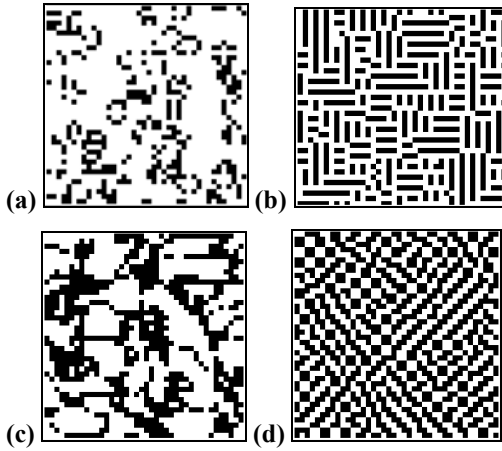
We emphasize that 2-dimensional CA are particularly useful to study the behavior of spatially distributed systems (physical and biological, other than computational ones), and to visualize their global evolution in a very simple way, by associating different colors to the states of the cells and by representing cells in a grid-shaped landscape (see Figure 1).

Given the quadruple  $A$ , which specifies the "static" characteristics of an automaton, we also have to define the update dynamics, in order to have the complete description of a CA. The usual definition of CA implies *synchronous dynamics*: cells update their state in parallel at each time step. However, despite being the most studied, synchronous CA are hardly representative of those real-world phenomena whose execution is characterized by the interactions of a population of autonomous interacting elements (as it is the case of distributed and pervasive computing systems). In these cases, asynchronous dynamics have to be introduced.

### 2.2 Asynchronous Dynamics

Accordingly to the most accepted terminology, a CA is *asynchronous* if cells can update their state independently from each other, according to a dynamics that can be either *step-driven* or *time-driven* [IngB84]. In *step-driven* dynamics, a kind of global daemon is introduced, whose job is to choose at each time step one (and only one) cell to update, accordingly to a specific sequence. In *time-driven* dynamics, instead, each cell is assumed to have an "internal clock" triggering the cell update. We focused on this latter case since in decentralized pervasive computing systems, autonomous processes execute and interact asynchronously

accordingly to a local internal clock.



**Figure 1: Synchronous vs. asynchronous dynamics in CA.** (a) Synchronous CA in one state of a cyclic attractor:  $S=\{0\text{-white},1\text{-black}\}$ ;  $N = 8$ ;  $f = \{\text{a cell in state } S=0 \text{ move to state } S=1 \text{ iff it has 2 neighbors in state } S=1; \text{ a cell in state } S=1 \text{ stays in that state iff it has 1 or 2 neighbors in state } S=1\}$ . (b) Its asynchronous counterpart in a fixed-point attractor. (c) Synchronous CA in one state of a cyclic attractor: (d)  $S=\{0,1\}$ ;  $N = 12$ ;  $f = \{\text{a cell in state } S=0 \text{ stays move to state } S=1 \text{ iff it has 6 cells in state } S=1; \text{ a cell in state } S=1 \text{ stays in that state iff it has 3,4,5 or 6 neighbors in state } S=1\}$ . (d) Its asynchronous counterpart in a fixed-point attractor.

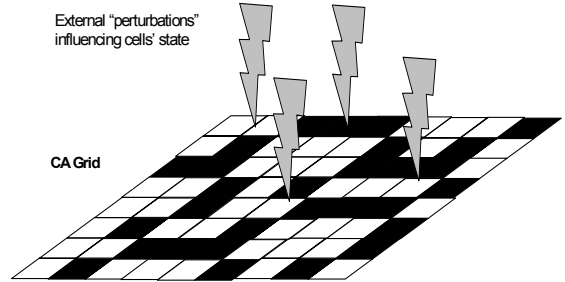
In the experiments presented in this paper, CA have an asynchronous time-driven dynamics: at each tick of the internal clock, a cell has a probability  $\lambda_a$  to wake up and update its state. The update of a cell has been implemented as atomic and mutually exclusive among neighbors to avoid concurrency problems.

In any case, we emphasize that the choice of the dynamic is substantial. In fact, asynchronous CA exhibit behaviors that are very different from the ones of their synchronous counterparts, both in terms of the dynamic and of the final attractors. Although both synchronous and asynchronous dynamics have the same fixed points [SchR99], i.e., attractors that are fixed points under synchronous dynamics are fixed points also under asynchronous dynamics and vice versa, the basins of attraction can be very different: some of the final attractors reached under asynchronous dynamics are hardly reached under synchronous one. As an example, Figure 1 compares the global states of two different CA under synchronous and asynchronous dynamics. Beside the visual differences in the perceived patterns, the synchronous regime makes both CA reach a cyclic attractor (i.e. the CA cycles through a number of states), while the asynchronous regime make both CA reach fixed-point attractors (i.e. the CA settles in a final state) that their synchronous counterparts have never been observed to be able to reach.

### 2.3 Perturbing CA Evolution

We are interested to study the evolution of CA in an open world, that is, when the internal dynamics of the CA can be influenced by the unpredictable effects of an external environment.

Such openness of the CA implies that some cells can be forced from the external to change their state, independently of their internal dynamic and transition function (See Figure 2). In a biological and social perspective, one could consider that the activities of individuals in a landscape, while being driven by internal goals, are continuously affected by the dynamics of the environment (e.g., weather conditions or influences of the society). In a computational perspective, which is that of more relevance here, one must consider that components of pervasive computing systems (agents and embedded sensors) will be typically devoted to monitor and control our physical environments, and will be influenced in their execution – other than by their reciprocal interactions – by what they sense in such environments. And, in several cases, environments possess a dynamics that is not controllable or foreseeable. For instance, the temperature and lightening condition in a room that a sensor is devoted to control may vary dynamically for a number of reasons that cannot be predicted.



**Figure 2: The basic structure of a perturbed cellular automaton**

All the above types of openness can be effectively modeled in cellular automata by having external perturbations change the internal states of the CA cells accordingly to specific dynamics. From a more formal point of view, we characterize a perturbed CA as:

- $A = (S, d, N, f)$ ;
- asynchronous time-driven dynamics (with probability  $\lambda_a$ );
- a perturbation action  $\varphi(\alpha, D)$ .

where  $A$  is the quadruple defining a CA, the dynamics is the one already discussed in Subsection 2.1, and the perturbation action  $\varphi$  is a transition function which acts concurrently with  $f$  and can change the state of any of the CA cells to a given state  $\alpha$  with some probabilistic distribution  $D$ , independently of the current state of the cells and of their neighbors. Specifically, in our experiments with  $S=\{0,1\}$ ,  $\alpha=1$  and  $D$  is

a uniform distribution of rate  $\lambda_e$ . Thus, to implement the effect of the perturbation in our simulations, we have superimposed a perturbation via a pool of processes (asynchronously triggered with probability  $\lambda_e$ ) capable of forcing the cells of the CA to change their state to  $I$ , independently of the internal cells' dynamics (Figure 2).

### 3 EMERGENT BEHAVIORS IN PERTURBED CELLULAR AUTOMATA

The behavior exhibited by perturbed CA is dramatically different from both their synchronous and their not-perturbed asynchronous counterparts. In particular, our experiments have focused on understanding the impact of the perturbation dynamics (determined by  $\lambda_e$ ) against the internal dynamics of asynchronous CA (determined by the rate of cell updates  $\lambda_a$ ).

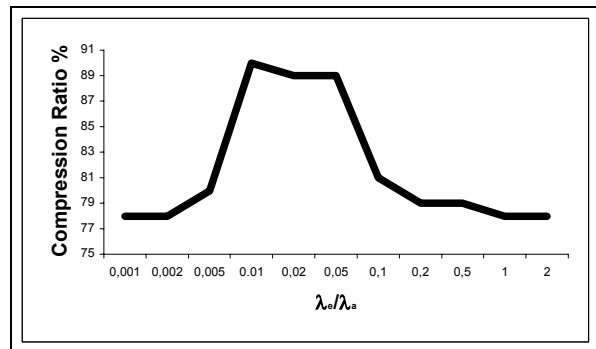
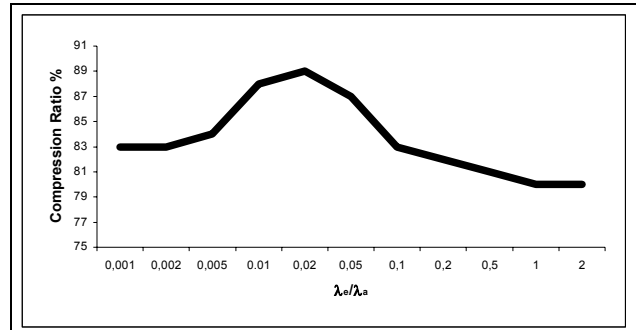
#### 3.1 The Effects of the Perturbation Dynamics

When the external perturbation is high enough to effectively perturb the internal dynamic of the CA, but it does not completely prevail over it (which happens when  $\lambda_e$  is comparable  $\lambda_a$ ), peculiar patterns emerge (see Figure 3 on the last page of this paper). In particular, we have observed that the perturbation on the cells induced by the external – while keeping the system out of equilibrium and making impossible for it to reach any equilibrium situation – tends to break localized stable sub-patterns facilitating the CA to self-organize into a large (or even global) coherent regular pattern. The interested reader can refer to the page <http://www.agentroup.unimore.it/DCA> to repeat these experiments on-line.

Figure 3 on the last page reports several examples of such phenomenon for different types of binary-state CA (e.g., with different transition rules and different structures of the neighborhood). There, for each type of CA, we have reported a sample of a typical pattern emerged under a specific value of the ratio  $\lambda_e/\lambda_a$ . This enables to clearly outline the influence of the ratio  $\lambda_e/\lambda_a$  and the generality of the observed behaviors. For all types of CA, the strictly local patterns that emerge in the absence of perturbation (or in presence of perturbations with a very low dynamic), tend to extend to a larger scale in the presence of (moderated) perturbations. In particular, in Figure 3, one can observe that, depending on the specific type of CA, these structures may sometime extend to the whole CA grid so as to make a unique global pattern emerge (e.g. Figure 3a, 3c, 3d), other times different large-scale structure, not extending to the whole grid, can coexists within it (e.g., Figure 3b, 3e).

The observed large-scale patterns are rather stable despite the continuous effects of the perturbations. However, in some cases, the emergent patterns are part of a dynamic structure that evolves due to the continuous perturbing effects, while always preserving the overall structure. For instance, the long diagonal stripes in Figure 3d (i) continuously change their micro-level shape, while maintaining the same global structure and (ii) they tend to continuously and slowly

translate horizontally in the CA lattice. Should the perturbation vanish after the patterns have already emerged, the emerged global patterns tend to suddenly freeze, but do not disappear.



**Figure 4: Regularity of the emergent patterns depending on  $\lambda_e/\lambda_a$  ratio, a measure by the degree at which the pattern can be compressed. Measures performed with the Lempel-Ziv algorithm (used in WinZip 8.0) on the binary strings representing the rows of the CA. (up): Same type of CA of Figure 3a. (bottom): Same type of CA of Figure 3d.**

The phenomenon that characterizes the behavior of perturbed CA can be intuitively described as follows. When the degree of perturbation is null or rather small, each cell reaches a local static equilibrium (or, in some of the observed cases, reaches soon a small, localized, cyclic attractor), which reflects in a global uniform equilibrium of the whole system characterized by a multiplicity of small localized stable structures. When the ratio  $\lambda_e/\lambda_a$  ratio increases, the system is kept in a substantial out-of-equilibrium situation, resulting in continuous attempts to locally re-establish equilibrium. This typically ends up with cell groups having found new equilibrium states more robust with regard to the perturbation (or compatible with it). These patterns are those that are able to continuously correct irregularities in their structure induced by the perturbation. Such stable local patterns, spread by the state transition rule, start dominating and influencing the surrounding, in a sort of positive feedback, until a large-scale (or even global-scale) stabilized situation

emerge. The large-scale stabilization typically results in visually observable large-scale patterns or, which is the same, in a measurable highly ordered configurations of the cells (Figure 4). When the degree of perturbation is high enough to avoid local stable patterns to persist for enough time, no local structure survive for enough time to enable large-scale structure to emerge, and the situation becomes somewhat “turbulent”.

### 3.2 Generality of the Observed Phenomenon

The above behavior is by no mean exhibited only by discrete cellular automata with binary states and connected over a regular lattice.

We have experienced with CA in which the state of cells of a CA can assume values over larger discrete domains (e.g. integer instead of binary state), and have observed that the same qualitative behavior is preserved whenever the state transition function exhibits non-linearity. As an example, Figure 5 shows the results obtained with a CA in which the local state of cells can assume integer values over the range 0—255, and in which a non-linear transition rule is imposed. As the figure outlines, a moderate degree of disturbances promotes the emergence of large-scale “islands” of cells which have somewhat coordinated their behavior.

We have also experienced with CA which cells are not connected in a regular lattice but in an amorphous network, i.e., in which cells are randomly placed in a 2-d space and in which the neighborhood structure is determined by the distances of the cells from each other. Also in this case, as Figure 6 shows, the presence of moderated perturbations makes large-scale patterns emerge that are not observed for a non-perturbed CA. Clearly, because of the amorphous nature of the network (compared to the regular lattice-like structure of the network in “traditional” CA), the patterns that emerge are not geometrically regular.

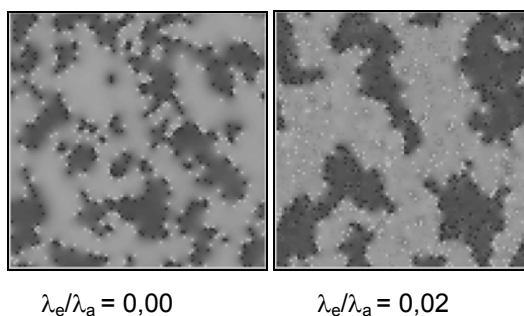
These results tend to confirm the generality of the phenomenon: the emergence of large-scale patterns from local interactions in the presence of disturbance, independently of the internal structures of cells and of the structure of the network in which they operate. An even more convincing argument of the generality of this phenomenon, however, comes from the observation that very similar phenomena occur in nature, in different classes of physical and biological systems of autonomous components, whenever properly “perturbed”.

As an example, J. Wootton [Woo01] recently explained the origin of the large-scale correlated patterns appearing in the distribution of mussels’ species in Pacific coasts. Wootton found out that the disturbances created by the ocean waves can promote the emergence of long-range correlations from local interactions. This explains the observed emergence of large-scale patterns in that species of mussels. This phenomenon can be directly assimilated to the one we have observed in our experiments.

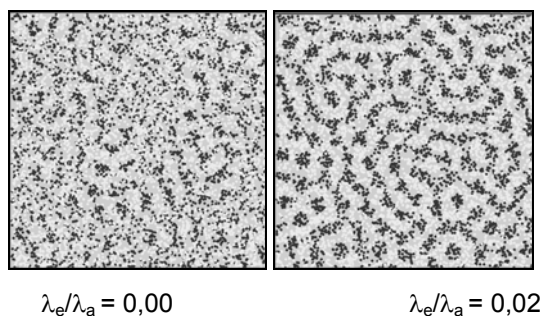
As another example, disordered waves of solitons (i.e., of

autonomous photons trapped by their own electric field) when interacting with each others (accordingly to local physical rules) and when perturbed by non-coherent wavelengths (e.g., in the presence of noise) may globally organize so as to form optical (observable) patterns with structures that, in some cases, are dramatically similar to the ones we observed in perturbed CA (e.g., the interested reader can compare the patterns of Figure 1a and 1c with those reported in [Kip00]).

Eventually, to some extent, perturbed CA can be assimilated to those open thermodynamic systems that, in the presence of energy injection, can exhibit the emergence of regular spatial structures (e.g. Bénard convection cells [NicP89]). However, in these cases, the principles underlying such phenomena are rather different from the ones underlying perturbed CA.



**Figure 5. Emergence of large-scale patterns in a perturbed nearly-continuous CA with the following characteristics:  $S=\{0(\text{white})\text{—}255(\text{black})\}$ ;  $N = 8$ ;  $f = \{\text{each cell evaluates its next state NS by comparing its current state CS and the average value A of neighboring cells: if } |CS-A| < 64 \text{ then } NS = A, \text{ if } |CS-A| > 64 \text{ and } A \geq 128 \text{ NS}=255, \text{ if } |CS-A| > 64 \text{ and } A < 128 \text{ NS}=0\}$ . Perturbations set cell state to a random value.**



**Figure 6. Emergence of large-scale patterns in an amorphous binary-state CA. Cells are randomly placed in a 2-d space, and the neighborhood of a cell is defined by all those cells that are within a specific distance (in the above test, the distance is set so that a cell is neighbor of 100 cells on average). The transition rule is as follows:  $f = \{\text{a cell in state } S=0 \text{ stays move to state } S=1 \text{ iff it has more than 30 and less than 50 neighbors in state } S=1; \text{ a cell in state } S=1 \text{ stays in that state iff it has more than 24 and less than 60 neighbors in state } S=1\}$ .**

## 4 IMPLICATIONS FOR PERVASIVE COMPUTING

As already outlined, perturbed cellular automata have characteristics that will be reflected by future pervasive computing systems: locality in interactions, autonomy of components, and situatedness in a dynamic environment. Despite the fact that CA represents an extremely limited model of pervasive computing systems, and in consideration of the generality of the observed phenomena, there are very good reasons to presume that the emergence of global spatial patterns exhibited by CA will be observed – in terms of emergent globally coordinated patterns of activities – as soon as our environments will start being populated by multitudes of dispersed computers-based systems.

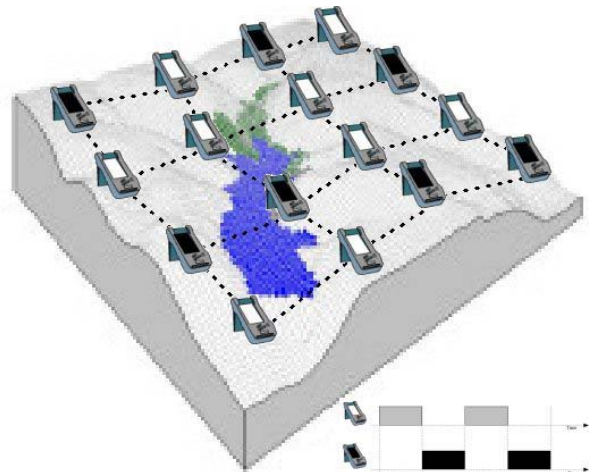
These considerations suggest that the design and development of pervasive computing systems (or of other highly decentralized software systems) should take into account these phenomena and be ready for:

- A prudential *defensive* approach, to prevent the emergence of dangerous counterproductive behaviors.
- a constructive *offensive* endeavor, to foster the emergence of specific emergence behaviors towards the achievement of otherwise difficult global application goals.

### 4.1 Defending from Emergent Behaviors

Defensively, the reported experiments alert on the possibility that a software system immersed in a dynamic environment may exhibit behaviors very different from the ones it was programmed and tested for.

To ground the discussion, let us consider a sensor network application scenario, in which a large number of computer-based devices with sensing capabilities are dispersed in an outdoor environment (see Figure 7). In general, the very goal of a sensor network is to spatially coordinate the activities of the sensors to cooperatively achieve specific sensing activities in an environment. For instance, we can consider that each sensor has a binary state variable representing the sensor duty-cycle (i.e. specifying whether the sensor is *fully-active* or *almost-sleeping*, i.e. performing only low-level tasks) and that neighbor sensors have to coordinate/synchronize their duty cycles so as to: (i) perform some collective sensing to detect events that no single sensor could detect in full by its own; (ii) save battery power by sleeping whenever possible, still ensuring that a given geographic zone is always properly monitored by a reasonable number of active sensor. Moreover, we can assume that peculiar environmental events can trigger suddenly a node duty-cycle to the *fully-operational* mode, if something happens that requires the sensor full attention. It is not difficult to recognize that, with these hypotheses, the sensor network strongly resembles asynchronous and perturbed CA.



**Figure 7. The nodes in the sensor network resemble the cells of a CA. Nodes binary state variable represents their duty-cycle.**

In this scenario, if unexpected macro-level spatial patterns in the duty-cycles of the sensor network emerge, they may cause large amount of data be left out from the network perception. In extremely unlucky situations, a sensor network could miss out a great portion of the events happening in the environment (e.g., if the event is spatially distributed like the emerged macro-level spatial pattern, but in a complementary way). Moreover, if the duty-cycles are used to control the degree of activity of a node (and thus its power consumption), the emergence of spatial patterns could produce imbalances in battery usage and cause network nodes in specific geographical regions to prematurely run out of power.

It is worth noting that these problems are of a very general nature, and similar considerations apply to a variety of other novel pervasive computing scenarios. In the case of information retrieval applications – and specifically of location-based information access – emergent coordination may cause a large amount of available information to be left out of the search, and have the remaining portion over accessed. Emergent coordination in the cells of a mobile telephony system (organized – as a CA – in regular grids) may lead to bandwidth saturation and cause denial of services malfunctioning. In mobile ad-hoc networks, emergent coordination in bandwidth utilization limits throughput and increases latency in message delivery. In computational self-assembly systems and in modular robots, whose very application goal is to coordinate the relative positioning of mobile computer-based particles, the emergence of spatial coordination patterns could simply prevent the achievement of the required application goal.

### 4.2 Exploiting Emergent Behaviors

The emergence of large-scale or global patterns in a system could also turn, in some situations, to be a useful and desirable characteristic.

In the above sensor network example, macro-level spatial patterns in the duty cycle of the sensors could enable to easily discover the presence of correlated spatial events happening in the environment. For example, if the duty-cycles in the network get arranged in a global pattern made of stripes (see figure 3a), nodes can easily detect stripe-like distributed events, by comparing their readings with neighbor ones. Specifically, the network detects a stripe-event if the nodes having a coordinated duty-cycle read similar events in the same cycle. More in general, the possibility of enforcing specific global spatial patterns of activities can be used to coordinate the sensing activities of sensor networks. While similar goals can be achieved via complex distributed coordination and synchronization algorithms, achieving them via the natural emergence could turn out to be much more efficient and simpler to be programmed.

As another application example, directly inspired from the visual appearance of the CA patterns, one could think at “intelligent paintings”. Paintings can be made up of active, radio-enabled, micro-components [MamZ04, Zam04], able to change their colors according to local transition rules, and exhibiting, when painted, nice globally ordered patterns.

More in general, the possibility of making global patterns emerge from a system relying on local interactions could be exploited so as to enforce global coordination and synchronization in a large-scale system with very low efforts. In perspective, one could think at exploiting the environmental dynamics to control and influence a decentralized distributed computing system from “outside the loop” [EstC02, Ten00], that is, without intervening directly on the system itself. In a world of continuous computations, where decentralized software systems are always running and cannot be stopped (this is already the case for Internet services and will be the case for networks of embedded sensors), changing, maintaining and updating systems by stopping and re-installing them is not feasible. Instead, given the availability of proper models and tools, one could envision the possibility of influence the system without stopping it, simply forcing specific environmental dynamics changing the global behavior of the system so as to make it exhibit the required behavior.

How such forms of control could be possibly enforced, will be discussed in the next section.

## 5 CONTROLLING CA BEHAVIOR

From the previous discussion, it turns out that it would be of fundamental importance to have the possibility of controlling emergent behaviors in CA. In fact, by considering again the relations with decentralized pervasive computing systems, having the possibility of controlling emergent behaviors may open up the doors to both defending from the emergence of undesirable behaviors and exploiting useful behaviors by making them emerge as needed.

However, due to the characteristics of such systems (large number of components, large-scale distribution and

decentralization, autonomy of components) one cannot think at controlling the global behavior of the system via a direct control on all its components. Rather, such control must be as much distributed and decentralized as possible, and should rely only on the possibility of controlling a few components of the systems, without making any assumption on the possibility of controlling all components and their dynamic interactions.

To this end, we experienced two complimentary ways of controlling emergent behaviors in perturbed CA, one of which having lead to quite successful, and rather surprising, results.

### 5.1 Generalization-based Control

The generalization-based methodology gets its inspiration from Hopfield’s work on neural networks [Hop82]. It is recognized that Hopfield’s networks can generalize a pattern from imposition of a partial pattern. Should perturbed CA exhibit such generalization property (that non-perturbed CA do not exhibit), one could exploit it to make any desired global pattern emerge.

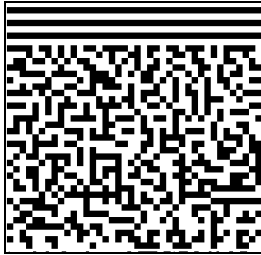
In CA, super-imposition of a partial pattern translates into initializing a localized sub-set of CA cells according to the desired patterns (see Figure 8 for an example applied to the CA of Figure 3a), and then let the CA evolve by making the global pattern spontaneously emerge from the local imposition.

In the case of a decentralized pervasive computing system, such methodology would imply the possibility of controlling the activities of a local cluster of components, and then let this locally imposed control diffuse to the whole system. This property would be very important in those cases in which only a spatially limited portion of the system is accessible for control and modification. For example, considering the presented sensor network scenario, such an approach would imply controlling some sensors close to an eventual base station and have the control diffuse to the whole geographic region.

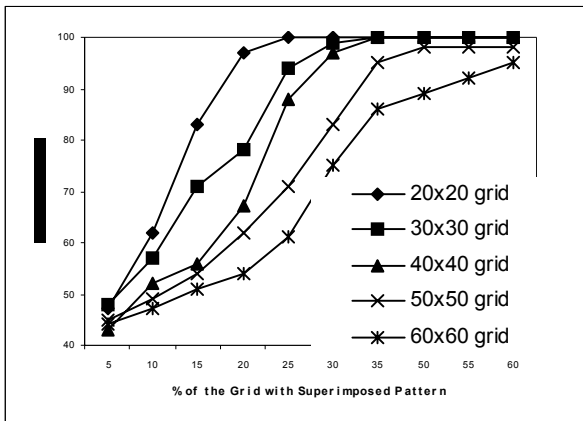
From our experiments, we found out that, for a few rules and for small grid dimensions, such methodology worked quite well (provided that the degree of perturbation dynamics was in the appropriate moderate range). For instance, to make a given pattern emerge out a 20x20 CA grid under the rule of Figures 1b and 3a, we had to initialize a local portion of about 20% of the global grid size. In the case of a pervasive computing system, this would mean that controlling the initial state of a cluster of a few dozens of component may be enough to influence the whole behavior of a system with hundreds of distributed components. Unfortunately, the generalization-based methodology appears to be neither general nor scalable. Generality is lacking because the methodology does not work with all types of CA, but only for those types of CA that exhibit, in the presence of perturbations, globally-extended patterns (e.g., the CA in Figure 3a, 3c, 3d). With regard to scalability, one can see

from figure 9 that, even if with specific rules the methodology can work quite well with small grids, the increase of the grid size requires a more than proportional increase in the size of the superimposed pattern. For instance, for a 60x60 grid, even by superimposing a pattern extending for the 60% of the grid, one still does not have reasonable guarantees to have such pattern extend to the whole grid.

As a further disadvantage, the above methodology cannot be exploited in any way to avoid a specific global configuration to emerge.



**Figure 8. Generalization-based methodology: a local pattern imposed on a portion of a CA grid (same type of CA as in Figures 1a and 3a).**



**Figure 9. Percentage of Successes of the Generalization-based Methodology, depending on the percentage of superimposed cells, and for different dimensions of the grid (same type of CA as in Figures 1a and 3a).**

## 5.2 Rule-based Control

The *rule-based* methodology starts from totally different considerations and attempts at controlling the global behavior of CA by changing – in a limited percentage of the system cells – the local rules  $f$  determining their state transitions. With respect to a decentralized pervasive computing system, such control methodology would imply having control over a limited percentage of its distributed components, and would translate in injecting in these components (e.g., via mobile code technologies [Bor02] and wireless communications) some new working parameters and procedures.

In general, to make a desired pattern emerge in the CA, the

rule has to change so as to make (some of the) cells recognize whether the cells in the neighborhood are in the right configuration (i.e., their states overall defines a local configuration that is compatible with the global one that is expected to emerge). Among a variety of possible rule modifications we have tested, we have found any good rule modification should not be too strict, i.e., it should not force a cell to continuously change its state in the attempt at locally forming a suitable configuration. For instance, consider the following modified rule: a cell should stay in its current state *if and only if* this state, together with the state of neighboring cells, overall defines a local configuration suitable to the emergence of the global one; otherwise it should change its state. For such strict rules, the whole automata end up in a chaotic state (cf. constraint-based CA in [Wol02]). Instead, we have found out that such rule modifications have to be very weak and, counter-intuitively, should enable limiting the “normal” state transitions of cells (as implied by the unmodified rule) whenever the state of neighbor cells defines a local configuration that is not in sharp contrast with the desired global one. For instance, given the generic rule for a binary-state CA:

$$f = \{ \text{a cell in state } S=0 \text{ must move to state } S=1 \text{ iff it has between } D_1 \text{ and } D_2 \text{ neighbors in state } S=1; \text{ a cell in state } S=1 \text{ remains in that state iff it has between } L_1 \text{ and } L_2 \text{ neighbors in state } S=1 \}$$

A simple modified rule  $mf$  enabling a specific pattern to emerge can be in the form:

$$mf = \{ \text{rule } f \text{ OR a cell must move to and stay in state } S=1 \text{ if the neighbor cells currently in state } S=1 \text{ defines a compatible configuration w.r.t. the desired pattern} \}$$

which can be roughly rephrased as follow: follow the normal transition rule unless the neighborhood situation is already satisfying with regard to the desired pattern.

Applying the  $mf$  rule (or, in general, a properly modified rule) to a non-perturbed CA, or to a CA perturbed with an inappropriate dynamic, gives not results at all, not even if a very large percentage of cells are modified. Either a local stable configuration is rapidly found by cells (in the case of null or low perturbation dynamics) or the situation becomes, as expected turbulent (in the case of high perturbation dynamics). However, in the presence of moderated perturbation dynamics (in the same range that let the “normal” not-modified CA exhibit global or large-scale patterns), the  $mf$  rule enables any desired pattern to emerge from any initial configuration of any CA, and to extend to a global scale, independently of the chosen  $f$  rule and independently of the dimension of the CA grid. For instance, we have been able to make the peculiar pattern of Figure 10 emerge from the CA represented in Figure 3d by applying the modified rule, a pattern that emerged only very rarely (about 0,01% of the cases) in previous experiments.

Of course, for a control methodology to be applicable to large distributed and decentralized systems, it must assume

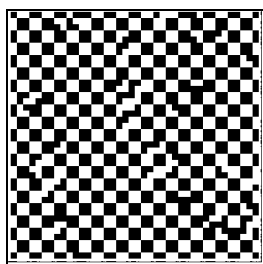


the capability of influencing the behavior of only a limited number of the components of a system. For this reason, we have tested the rule-based methodology also by modifying the rule only in a limited percentage of the CA cells. As Figure 11 shows with regard to a specific rule, such experiences have been very satisfying: rule-based control enables a desired pattern to emerge (and extend over the whole grid) even when only a very low percentage of the cells apply the modified rules. In particular, as soon as more than 25% of the cells are controlled, the desired patterns emerge in nearly 100% of the cases. These results are almost independent of the dimension of the grid, making the methodology scalable. Very similar quantitative results apply for different rules.

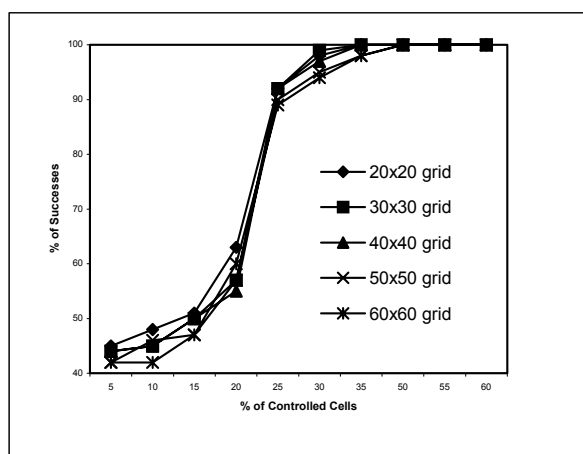
From the point of view of avoiding a global pattern to emerge, the method is even more effective. In that case, unlike the previous one, one must apply to a portion of the cells a rule explicitly contrasting the emergence of the undesired pattern, i.e., a rule forcing a cell to change its state whenever the local configuration would be compatible with the global pattern to be avoided:

$mf = \{rule\ f\ AND\ a\ cell\ must\ change\ its\ state\ as\ soon\ as\ it\ recognizes\ in\ the\ neighborhood\ a\ local\ configuration\ which\ is\ compatible\ to\ the\ global\ one\ to\ be\ avoided\}$

By applying such a rule even to a very low percentage of cells (about 5%) one can avoid emergent patterns to extend to a global scale, although some of them will always be enabled to extend to a rather large-scale. By increasing the percentage of modified cells, emergent patterns are more and more constrained in their possibility to extend to a large scale, until (when the percentage of modified cells grows up to 20%) the effect of perturbations in making large scale patterns emerge is almost annihilated.



**Figure 10. Rule-based methodology: a rare pattern whose emergence can be controlled (same type of CA as in Figures 1c and 3d).**



**Figure 11. Percentage of Successes of the Rule-based Methodology, depending on the percentage of controlled cells, and for different dimensions of the grid. ( same type of CA as in Figures 1c and 3d)**

## 6 RELATED WORK

In this section we briefly discuss related work in the areas of cellular automata, multiagent systems, and pervasive computing systems.

### 6.1 Cellular Automata

CA has been extensively studied in the scientific literature, with a specific interest on the very properties of CA, or on their application as computational systems [Wol94, Wol02], or in the exploitation of CA for simulation purposes [Bar97]. Strictly related, studies in the so called cellular programming area aim at exploiting the emergent behaviors of non-uniform CA in which each cell can have its own local transition function for applications to image recognition, combinatorial optimization problems and evolvable hardware [Sip99]. In the area of simulation [AdaPL04, Bar97, Ban98, BCT02], CA have been and are still widely exploited for the simulation of biophysical processes and socio-economical phenomena.

In most of the above studies, CA are considered as synchronous and closed systems, for the sake of achieving determinism and predictability in CA's behavior (and, thus, in the performed experiments). Some works recognize the peculiar and interesting computational behaviors exhibited by asynchronous model [IngB84, LumN94, SchR99, WeiYW03]. Nevertheless, they still miss in identifying the strong influences that the openness of the system and the perturbation of the environment can have on the behavior of the CA. Possibly, our investigations on perturbed CA may lead to further useful application of cellular-based computational approaches.

Strictly related to works on CA are researches on boolean networks [Kau93]. From a broad perspective, boolean networks can be considered as sorts of non-uniform CA with a topology of interconnection that can be described as a

directed graph, typically not regular. These types of networks have found several interesting applications in modeling biological systems (e.g., genetic networks). Still, as in the case of CA, the analysis of the influence of a perturbing environment on network dynamics is missing (the effects of an external perturbation are only considered referring to the robustness of attractors).

Studies on stochastic cellular automata [Bar97] (i.e., CA in which transition rules are probabilistic) have shown that non-determinism in cell update may lead to global scale spatial patterns emerge. Non-determinism can be considered as a sort of openness, therefore stochastic automata strongly resemble perturbed CA in their global dynamics. Nevertheless, this model still lacks of explicit identification of non-determinism as a sort of openness to the environmental dynamics and of its relations to agent-based computing. To the best of our knowledge, the only work that identifies the potential relations between stochastic automata and distributed computing systems is described in [BarD01]. As discussed in Section 4, the authors identifies that global scale behavior emerging in stochastic CA can potentially be used as a tool for globally coordinating the behavior of multiagent systems. Still, the authors experience with mono-dimensional CA, thus missing the spatial expressiveness that instead emerges in our two-dimensional experiments. Moreover, the lack of modeling environmental dynamics let the authors miss the potential dangers of such emergent behaviors.

## 6.2 Multi-agent systems

In the computer science community, the specific area in which the problems of emergent behaviors have been received more attention is the area of multiagent systems. This is mainly due to the fact that multiagent systems are made up of autonomous and situated entities, and thus are prone to exhibit unexpected emergent behaviors.

Since the origins of distributed artificial intelligence and of multiagent systems researches, a large amount of studies have shown that systems in which autonomous components interact with each other in a network, and change their status accordingly to the outcomes of these interactions, can make peculiar global behaviors emerge [GasB92, HubH93]. Recent examples of these studies may be found in the area of computational markets [KepHG00, Tes02] and of computational ecosystems [Huh01]. However, most of these studies focused on the internal dynamics of the system, without taking into account the perturbation of the environment.

Studies in the area of artificial social laws [MosT95] show that global rules constraining the behavior of all the agents in a group can notably influence the dynamic behavior of the group. Analogously, studies adopting an organizational metaphor for the design of multiagent systems [JenB03, CabLZ03, ZamJW03], show that the definition of global environmental rules to which all agents must obey is very useful toward the effective control of the global multiagent

system behavior. For all the above approaches, the basic intuition is that agents, for the very fact of living in an environment (i.e., a society or an organization) are not fully autonomous but, instead, their actions can be constrained and influenced by the environment, the same as the state of the cells in CA can be changed by the perturbation function. However, the above studies exploit such kind of environmental abstractions constructively during the design process, and assume having full control over the environment behavior. Thus, they miss in identifying that agents may live in dynamic environment, where the rules governing their execution and their interactions can change during the evolution of the multiagent systems and can influence their behavior in unpredictable (or simply uncontrollable) way.

The importance of the environmental abstraction and of its dynamic in the global behavior of the system is properly attributed in the study and implementation of field-based [BanMV04] and ant-based multiagent systems [Par97, BonDT99, ParB01]. In these systems, very simple agents can indirectly interact with each other in a local way, by spreading synthetic fields or pheromones in the environment and by locally sensing their concentration. Such models can achieve difficult goals, such as: finding shortest paths, clustering data, etc., by exploiting self-organization and emergent phenomena. The similarities between these multiagent systems and perturbed CA are strong: they both exploit asynchronous components affected in their execution by the environmental dynamics, and both evolve to create global coordinated activity patterns. However, till now, such researches have focused on the possibility of “designing” the environment and of controlling its dynamics to constructively exploit it. Few researchers explicitly focused on the perturbing effects that uncontrollable environmental dynamics can have on the global behavior of a system [ParBS01], and on the possible way to control it.

## 6.3 Decentralized Pervasive Computing

Inspired by recent advances in communication and microelectronics technologies, a vast amount of recent researches propose the exploitation of clouds of small-scale computer-based systems, to be embedded in our everyday objects and to be dispersed in our everyday environments, so as to enrich them with smart sensing and actuating capabilities. The very goal of the applications proposed in this area is somehow related to make the distributed components (nodes, sensors, etc.) coordinate with each others in spatial patterns of activities that are suitable for the achievement of global application-specific tasks.

In mobile ad-hoc and sensor networks, one of the mainstream research focus concerns the design of distributed sensing and activity patterns on the basis of the nodes physical locations. For example, some algorithms to let nodes self-organize a shared coordinate system have already been proposed [BacN03]. By using these algorithms the single components become aware of their and of neighbors’ location within the network and can use this information to send messages to

physically-bounded regions in the network. Other important algorithms enable nodes to differentiate their activities on the basis of their location. For instance, such algorithms could allow nodes distant from any relevant source of events to differentiate their activities to sleep saving battery power. In any case, in all these works, the main focus is to devise algorithms to obtain specific functionalities *by design* (i.e. the algorithm explicitly and directly addresses the desired functionalities). The possibility that the same functionalities can be obtained, in a simpler and more efficient way by adopting an *emergent* approach, similar to the one we propose for DCA (i.e. relying on a bottom-up, trial-and-error design), is mostly neglected.

Other interesting areas of research, addressing the problem of globally coordinating the activities of a large number of distributed components, are modular robots and self-assembly of computational particles [StoN04]. In these researches, the idea is to have a collection of simple autonomous robots (or computational mobile particles) running a distributed algorithm to assume a global shape or a global coherent motion pattern. One of the most successful approaches in this scenario adopts data structures resembling fields and morphogen gradients. The particles constituting the modular robot have the basic capabilities of propagating sorts of abstract computational fields in the network, and to sense and react to such fields. In particular, particles can transfer an activity state towards directions described by fields' gradients, so as to make coordinated global patterns of activities emerge in the system [StoN04]. Although some of the proposed results are indeed impressive, they are achieved via complex distributed algorithms, and the possibility that emergent behaviors could be exploited to achieve similar results in a simpler way is disregarded.

To the best of our knowledge, none of the above research areas seriously take into consideration the possibility that not-designed, counterproductive and dangerous behaviors, can emerge in the system. On the contrary, the analysis and the experiments reported in this paper alert that this possibility is indeed concrete.

In summary, by considering the strict similarities between the above scenarios and the one of perturbed cellular automata, we expect that analysis of emergent behaviors and tools to control them, possibly along the lines we have sketched in this paper, will soon be required in the domain of decentralized pervasive computing.

## 7 CONCLUSIONS AND FUTURE WORK

This paper has reported the outcomes of a set of experiments that we have performed on a large set of asynchronous CA by super-imposing an external perturbation that dynamically influences the state transitions of cells. The experiments have shown that the perturbation makes large-scale spatial structures emerge, which have not observed under synchronous and unperturbed regime. Starting from that observation, the paper has argued that, since future decentralized and pervasive computing systems will exhibit

all of the characteristics of perturbed CA (local interactions, autonomy of components, and influence by a dynamic environment), they will be likely to exhibit very similar behaviors. This calls for appropriate models, methodologies and tools, explicitly taking into this emergent phenomenon and possibly exploiting it offensively, as a way to effectively achieve global-scale coordination in a system, or defensively, to prevent and control the emergence of undesired large-scale coordination patterns of activity. In this context, a further set of experiments have shown that it is possible to control the emergence of a pattern in a perturbed CA by means of a fully decentralized mechanism. We argue that a similar sort of control can be applied to control the behavior of pervasive and decentralized computing systems.

Our results, although applied to a very minimal model of pervasive computing system, motivate further work and experiments. In particular:

- more experiments are needed to evaluate the behavior of CA under a variety of different perturbation regimes, other than the simple ones we have discussed in this paper;
- more extensive experiments are needed to evaluate the behavior of more complex CA, i.e., CA with more complex transition rules and in which cells may have differentiated behaviors and may be connected according to a variety of irregular patterns (e.g., amorphous CA and small-world CA);
- strictly related, we have to evaluate the applicability of the proposed control methodologies (discussed in this paper with regard to simple and regular CA) in the context of more complex and amorphous CA.

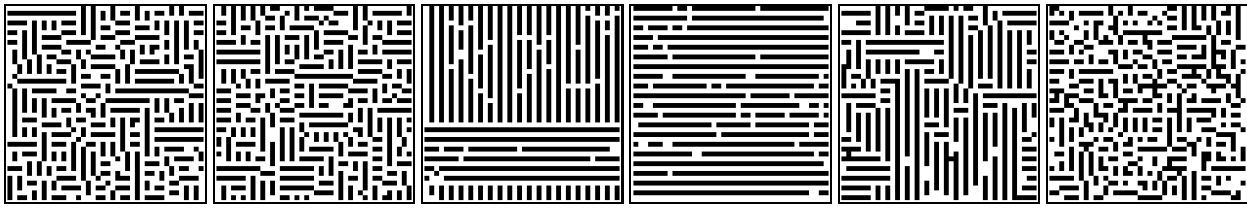
The main objective is to make our experiments more and more approximate the actual characteristics of real-world pervasive computing scenarios and, eventually, to end up with a powerful simulation environment and with effective general-purpose tools to control the global behavior of large scale decentralized pervasive computing systems.

## REFERENCES

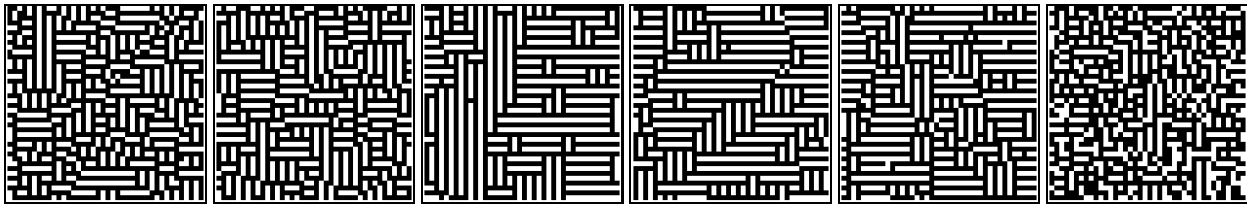
- [AdaPL04] S. Adachi, F. Peper, J. Lee, "Computation by Asynchronously Updating Cellular Automata", *Journal of Statistical Physics*, 114(1), Jan 2004.
- [AlbJB99] R. Albert, H. Jeong, A. Barabasi, "Diameter of the World Wide Web", *Nature*, 401:130-131, 9 Sept. 1999.
- [AlbJB00] R. Albert, H. Jeong, A. Barabasi, "Error and Attack Tolerance of Complex Networks", *Nature*, 406:378-382, 27 July 2000.
- [Ban98] S. Bandini, R. Serra, F. Suggi Liverani (Eds.). *Proceedings of the 3rd Conference on Cellular Automata for Research and Industry*. Springer-Verlag, 1998.
- [BacN03] J. Bachrach, R. Nagpal, M. Salib, H. Shrobe, "Experimental Results and Theoretical Analysis of a Self-Organizing Global Coordinate System for Ad Hoc Sensor Networks", *Telecommunications Systems Journal, Special Issue on Wireless System Networks*, Kluwer Academic Publishing, 2003.

- [BanMV04] S. Bandini, S. Manzoni, G. Vizzari, "Towards a Specification and Execution Environment for Simulations based on MMass: Managing at-a-distance Interaction", AT2AI-4, April, 2004, Vienna, Austria.
- [Bar97] Y. Bar-Yam, "Dynamics of Complex systems", Addison-Wesley, 1997.
- [BarD01] T. D. Barfoot, G. M. T. D'Eleuterio, "Multiagent Coordination by Stochastic Cellular Automata", Proceeding of the Joint International Conference on Artificial Intelligence, Seattle (WA), Aug. 2001.
- [BCT02] S. Bandini, B. Chopard, M. Tomassini. Proceedings of the 5rd Conference on Cellular Automata for Research and Industry. Springer-Verlag, 2002.
- [BonDT99] E. Bonabeau, M. Dorigo, G. Theraulaz. "Swarm Intelligence. From Natural to Artificial Systems". Oxford University Press, 1999.
- [Bor02] C. Borca, et al., "Cooperative Computing for Distributed Embedded Systems", 22th International Conference on Distributed Computing Systems, Vienna (A), IEEE CS Press July 2002.
- [CabLZ03] G. Cabri, L. Leonardi, M. Mamei, F. Zambonelli, "Location-dependent Services for Mobile Users", IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems And Humans, Vol. 33, No. 6, pp. 667-681, November 2003.
- [EstC02] D. Estrin, D. Culler, K. Pister, G. Sukjatme, "Connecting the Physical World with Pervasive Networks", IEEE Pervasive Computing, 1(1):59-69, Jan. 2002.
- [GasB92] L. Gasser, J. P. Briot, "Object-based Concurrent Programming and Distributed Artificial Intelligence", in Distributed Artificial Intelligence: Theory and Practice, Kluwer Academic, pp. 81-107, 1992.
- [Hop82] J.J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", Proceedings of the National Academy of Science USA, 79:2554-2558, 1982.
- [HubH93] B. A. Huberman, T. Hogg, "The Emergence of Computational Ecosystems", in SFI Studies in the Science of Complexity, Vol. V, Addison-Wesley, 1993.
- [Huh01] M. Huhns, "Interaction-Oriented Programming", 1st International Workshop on Agent-Oriented Software Engineering, LNCS No. 1957, Jan. 2001.
- [IngB84] T. E. Ingerson, R. L. Buvel, "Structure in Asynchronous Cellular Automata", Physica D, 10:59-68, 1984.
- [JenB03] N. Jennings, S. Bussmann, "Agent-based control systems", IEEE Control Systems Magazine 23 (3) 61-74, 2003.
- [Kau93] S. A. Kauffman. The origins of order. Oxford University Press, New York, 1993.
- [KepC03] J. Kephart, D. M. Chess, "The Vision of Autonomic Computing", IEEE Computer, 36(1):41-50, Jan. 2003.
- [KepHG00] J. O. Kephart, J. E. Hanson, A. R. Greewald, "Dynamic Pricing by Software Agents", Computer Networks, 32(6): 731-752, May 2000.
- [Kip00] Kip, D., Soljacic, M., Segev, M., Eugenieva, E. & Christodoulies, D. N. Modulation Instability and Pattern Formation in Spatially Incoherent Light Beams, Science, 290:495-498 (2000).
- [LumN94] E. D. Lumer, G. Nicolis, "Synchronous Versus Asynchronous Dynamics in Spatially Distributed Systems", Physica D, 71:440-452, 1994.
- [MamZ04] M. Mamei, F. Zambonelli, "Programming Pervasive and Mobile Computing Applications with the TOTA Middleware", IEEE Conference on Pervasive Computing and Communications (Percom), Orlando (FL), USA, March, 2004.
- [MosT95] Y. Moses, M. Tenneholtz, "Artificial Social Systems", Computers and Artificial Intelligence, 14(3):533-562, 1995.
- [NicP89] G. Nicolis, I. Prigogine, Exploring Complexity: an Introduction, W. H. Freeman (NY), 1989.
- [ParB01] V. Parunak, S. Brueckner, "Entropy and Self-Organization in Agent Systems", 5th International Conference on Autonomous Agents, Montreal (CA), May 2001.
- [ParBS01] V. Parunak, S. Brueckner, J. Sauter, "ERIM's Approach to Fine-Grained Agents", NASA/JPL Workshop on Radical Agent Concepts, Greenbelt (MD), Sept. 2001.
- [Par97] V. Parunak, "Go to the Ant: Engineering Principles from Natural Agent Systems", Annals of Operations Research, 75:69-101, 1997.
- [Pis01] K. Pister, B. Warneke, M. Last, B. Leibowitz, "Smart Dust: Communicating with a Cubic-Millimeter Computer", IEEE Computer, 34(1):44-51, Jan. 2001.
- [RipIF02] M. Ripeani, A. Iamnitchi, I. Foster, "Mapping the Gnutella Network", IEEE Internet Computing, 6(1):50-57, Jan.-Feb. 2002.
- [SchR99] B. Schönfisch, A. De Roos, "Synchronous and Asynchronous Updating in Cellular Automata", BioSystems, 51(3):123-143, 1999.
- [Sip99] M. Sipser. "The Emergence of Cellular Computing". IEEE Computer, 37(7):18-26, July 1999.
- [StoN04] K. Stoy, R. Nagpal, "Self-Reconfiguration Using Directed Growth", 7th International Symposium on Distributed Autonomous Robotic Systems (DARS 2004), Toulouse, France, June 23-25, 2004.
- [Ten00] D. Tennenhouse, "Proactive Computing", Communications of the ACM, May 2000.
- [Tes02] L. Tesfatsion, "Agent-Based Computational Economics: Growing Economies from the Bottom Up", Artificial Life, Vol. 8(1), pp. 55-82, MIT Press, 2002.
- [WeiYW03] Y. Wei, S. Ying, Y. Fan, B. Wang, "The Cellular Automaton Model of Investment Behavior in the Stock Market", Physica A, 325(3-4), 2003.
- [Wol94] S. Wolfram. Cellular Automata and Complexity. Addison-Wesley, 1994.
- [Wol02] S. Wolfram, A New Kind of Science, Wolfram Media, Inc., 2002.
- [Woo01] J. T. Wootton, "Local Interactions Predict Large-scale Patterns in Empirically Derived Cellular Automata", Nature, 413: 841-844, 25 Oct. 2001.
- [ZamJW03] F. Zambonelli, N. R. Jennings, M. J. Wooldridge, "Developing Multiagent Systems: the Gaia Methodology", ACM Transactions on Software Engineering and Methodologies, 12(3):417-470, July 2003.
- [Zam04] F. Zambonelli, M-P. Gleizes, M. Mamei, R. Tolksdorf, "Spray Computers: Frontiers of Self-Organization for Pervasive Computing", 14th IEEE Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, IEEE CS Press, Modena (I), June 2004.

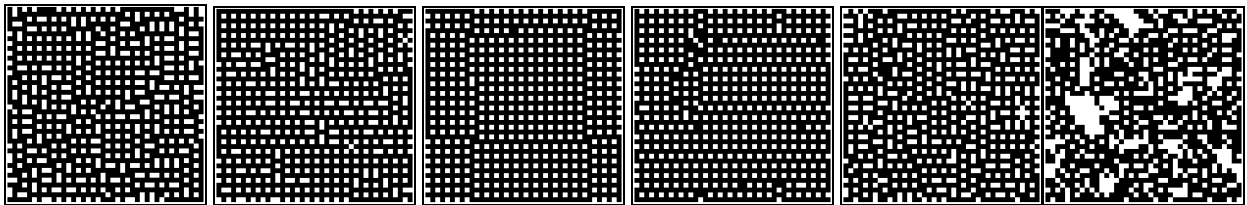
(a)  $S=\{0\text{-white},1\text{-black}\}$ ;  $N = 8$ ;  $f = \{ \text{a cell in state } S=0 \text{ move to state } S=1 \text{ iff it has 2 neighbors in state } S=1; \text{ a cell in state } S=1 \text{ stays in that state iff it has 1 or 2 neighbors in state } S=1 \}$



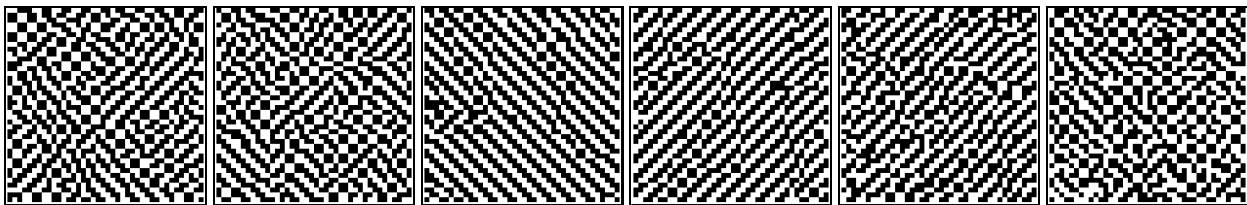
(b)  $S=\{0,1\}$ ;  $N = 8$ ;  $f = \{ \text{a cell in state } S=0 \text{ move to state } S=1 \text{ iff it has 3, 4, or 5 neighbors in state } S=1; \text{ a cell in state } S=1 \text{ stays in that state iff it has 2, 3, 4 or 5 neighbors in state } S=1 \}$



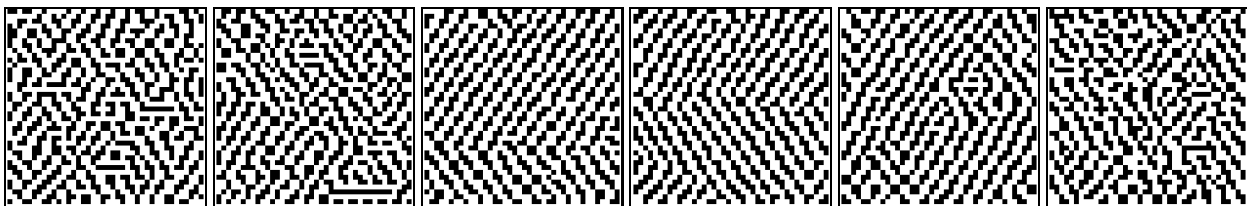
(c)  $S=\{0,1\}$ ;  $N = 8$ ;  $f = \{ \text{a cell in state } S=0 \text{ move to state } S=1 \text{ iff it has 4, 5, or 6 neighbors in state } S=1; \text{ a cell in state } S=1 \text{ stays in that state iff it has 3, 4, 5 or 6 neighbors in state } S=1 \}$



(d)  $S=\{0,1\}$ ;  $N = 12$ ;  $f = \{ \text{a cell in state } S=0 \text{ move to state } S=1 \text{ iff it has 6 neighbors in state } S=1; \text{ a cell in state } S=1 \text{ stays in that state iff it has 3, 4, 5 or 6 neighbors in state } S=1 \}$



(e)  $S=\{0,1\}$ ;  $N = 20$ ;  $f = \{ \text{a cell in state } S=0 \text{ move to state } S=1 \text{ iff it has 6 neighbors in state } S=1; \text{ a cell in state } S=1 \text{ stays in that state iff it has 3, 4, 5 or 6 neighbors in state } S=1 \}$



$\lambda_e/\lambda_a$  0,00      0,01      0,02      0,05      0,1      0,2

Figure 3: The behavior of different CA depending on the  $\lambda_e/\lambda_a$  ratio.