ICMC 1989, The Ohio State University, Columbus, OH

The Musical Universe Of Cellular Automata

Peter Beyls

Artificial Intelligence Lab,
Brussels University,
Pleinlaan, 2
1050 Brussels Belgium
email: peter%arti1.vub.ac.be

ABSTRACT

First, the general framework of complex dynamics, as a fundamentally new computational paradigm is briefly introduced. Within this context, we elaborate on empirical work in the domain of cellular automata (CA), discrete computational devices which allow for the simulation of complex phenomena, including wave propagation, natural growth and self-organization. A CA is a regular assembly of cells whose global behaviour is completely specified in terms of local operations. To move on from one generation to the next, a single rule is evaluated for all cells simultaneously. Very complex and surprisingly intricate overall behaviour results from the application of amazingly simple rules. Many types of rules are explored and a number of extensions on the basic scheme are suggested: multi-level automata, history tracking, the use of 2D rules in linear automata, the introduction of feedback etc. Procedural mapping for real-time composition and performance is discussed in detail. The universal computational capabilities of CA may be appreceated from the many illustrations documenting the present paper.

1. Introduction.

The idea to think of intellectual activity as a distributed process in which a large number of elements interact, is by no means a new one. Examples include the actor paradigm where problem solving is seen as the creation of specialised actors which perform a collection of tasks and which communicate amongst eachother through message passing (Hewitt 77). Another more recent example are neural networks where a typically very large number of elements are linked together. These networks evolve in time by adjusting the strength of mutal connections between elements (Hopfield 82). Still another example is Minsky's "society of mind". The idea is that mental activity can be seen as a process where many simple agents interact localy and assemble themselves in hierarchies. From this simple activity on the microscopic level, complex overall behaviour may result on the surface. Many aspects of human intelligence can be modelled this way (Minsky 86).
A most recent paradigm introduces a major swing from symbolic computing. The dynamics paradigm, introduced by Steels, proposes a fundamentally novel approach to the machine modelling of cognitive activity. This work is inspired by concepts in physical chemistry, biology and genetics and speculates on the potential of recent advances in hardware: the creation of massively parallel computer architectures (Hillis 86). This paradigm views computation as a complex dynamical (CD) system because it incorporates a very large number of interacting particles evolving in parallel over time (Steels 88). It goes without saying that CD redefines our inclination towards computers completely. I am convinced that it will also change the way we look at computer based music production in the future.

2. Complex dynamics.

2.1 Definition.

A system is complex if:
- it has a large number of similar, simple elements
- all elements evolve in parallel over time
- the same external rule applies to all elements simultaneously
- any element performs local interactions only
- the system exhibits emergent global properties

Nature offers a wealth of examples of complex systems whose overall behaviour is beyond simple analysis, yet whose basic components are extremely simple. For instance, spectacular examples of self-organisation are seen in biology, chemistry and physics. Self-organisation in ant collonies is a classic example and has been studied by (Deneubourg 83). Consider the spontanous generation of oscillating patterns in the well known Belousov-Zhabotinsky reaction: intricate behaviour suddenly appears from the cooperation of billions of molecules without any overall guidance (Babloyantz 81).
In these processes there is no central supervisor, the only activity being cooperation between neighboring elements. In this light, consider also the striking example of genetic evolution through natural selection (Dawkins 86).

2.2 How to describe dynamical systems?

Traditionally, differential equations are used as descriptors but at higher levels of complexity they are no longer useful because of limitations of representational bandwidth and because of being too difficult to solve. Geometrical descriptions received wide spread use as means to visualize dynamical systems. The idea is to represent the system's behaviour in a phase portrait: to think of the system's characteristics going through trajectories in space. The pendulum with friction is an example of a simple dynamical system with a single attractor, i.e. no mather what the starting position, the pendulum will always evolve to the same position. Chaotic systems exhibit many different limit points. i.e. points of focal attraction. Incidentally, the nonlinearity in a system may increase to unpredictable levels as the number of limit points builds up leading to the formation of a strange attractor.
Cellular automata constitute an alternative and pragmatic means to attack the problem of describing, generating and analysing complex systems. As computational descriptions, they are instrumental to a better understanding of the mechanisms which are responsible for the synthesis of complexity. Using CA it is perfectly possible to simulate systems with any degree of periodicity, including systems with strange attractors. We will not go into any further details here, for an excellent introduction to complex dynamics and its significance for artificial intelligence, please refer to (Steels 88).

2.3 Musical motivation.

As a composer I am interested in models of evolution and growth rather than in theories for structural design. The parameter "change" is the primary concern. CA are devices to create an alternative universe with a complex yet controllable behaviour. CA are models of discrete dynamical systems with interesting evolutionary characteristics. Many automata exhibit attractors because, whatever the initial configuration, their behaviour becomes cyclic: complex periodic behaviour emerges as a form of self-organisation. This means that structure is created from an originally totally random configuration. This may happen after any number of generations. The whole plethora of limit points, limit cycles and strange attractors may be observed from the extremely simple mechanism of a CA. CA may be seen as complexity amplifiers. Their experimental design is coherent with an exploratory attitude in musical composition. Also, CA are visually attractive because "what you see is what you conceive". Their universal character stimulates the mind towards discovery of new and musically interesting rules.

2.4 Related work in the arts.

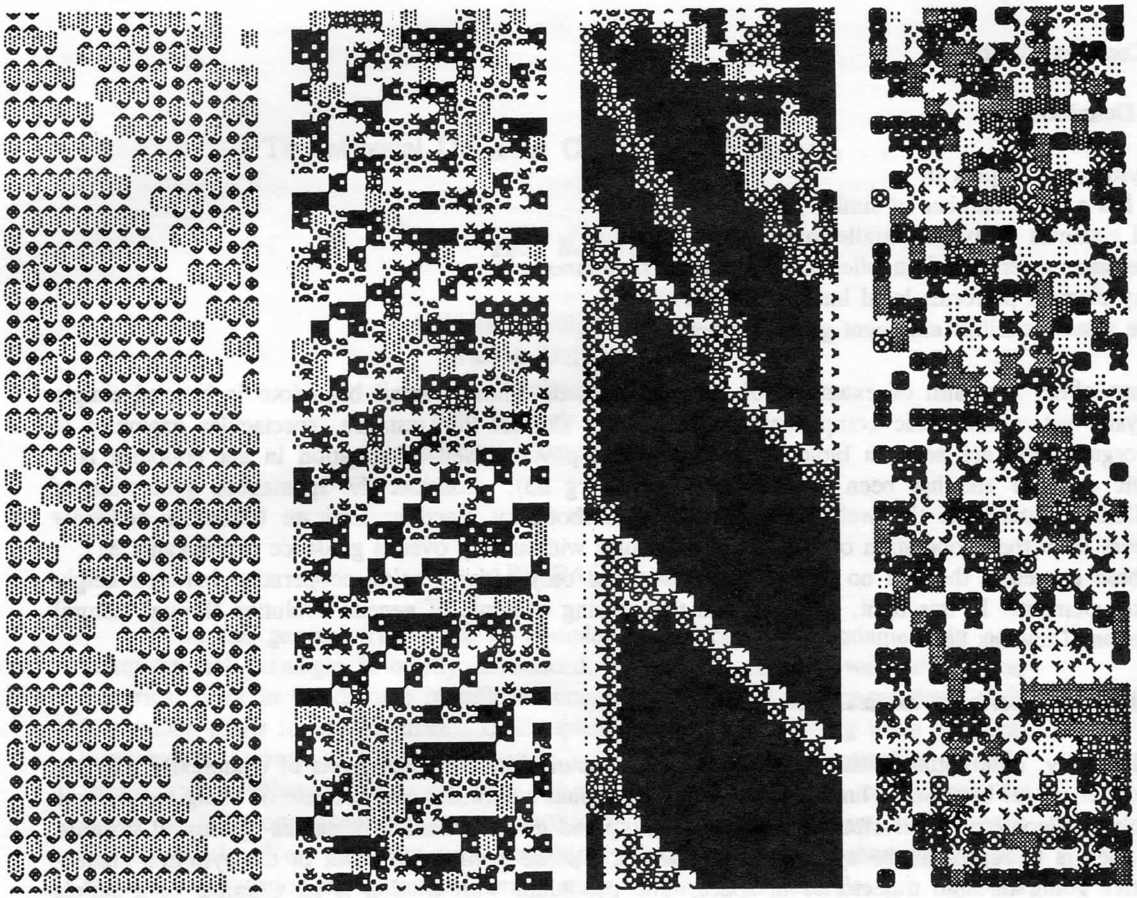

Figure 1.
Examples of continuous automata.

Figure 2.
Examples of linear automata with 2-dimentional rules.

Cellular automata have been applied extensively in the graphic arts. Sometimes as building blocks to design generators of exceptional visual complexities. Sometimes as filters for digital image processing: typically, the same rule is evaluated for all pixels constituting a picture. CA offer an elegant and economic means to simulate natural phenomena (such as cloud formation or wave propagation on a water surface) in pictures. Sometimes, the artist is fascinated by the act of inventing some complex behaviour through the use of CA -- the aim is to simulate some type of artificial life. Consequently, resulting works are "snapshots" taken at regular intervals (Sullivan 78, Beyls 80). Even special purpose graphic design languages based on CA were developped. The legendary EXPLOR system created at Bell Labs (Knowlton 79) is a classic examples. CA have a less strong tradition in music. However, CA were studied and applied in LINA, an installation piece by Joel Ryan (Ryan 89); it is an example of CA used on the level of musical events (notes) as opposed to application on audio events (samples). LINA also includes real-time video processing of the automata patterns, controlled through MIDI. It is important to take note of the size of the "time cells". Music-time means we are dealing with MIDI events, audio-time means we are interested in the manipulation of audio samples. New synthesis algorithms using CA are investigated in (Beyls 89). The introduction of the Karplus-Strong algorithm for synthesis of complex time variant spectra has generated a wide spread interest and pushed research on local computation in audio-time toward new dimentions. Many variations and extensions of the basic KS-algorithm are now being investigated.

3. Cellular automata.

3.1. Background.

The idea of cellular automata was first introduced by Von Neuman in the fifties in a wish to build a system capable to reproduce itself in a way analogous to biological reproduction in living organisms. The best known two-dimentiaonal automaton -- also capable to demonstrate principles of self-reproduction -- is the game of life (Conway 70). This game clearly illustrates that intricate overall behaviour may result from simple rules operating on the microscopic level. Complex patterns emerge "spontaneously" from local interactions between neighboring cells. The recent book by [Toffoli and Margolus 87] was also an extermely stimulating source of ideas. In this paper we will mainly study one-dimentional automata: our universe consists of a single string of contiguous cells.

3.2. Definition.

We define a CA as exhibiting the following minimum requirements:
- there are a finite number of identical cells
- the same rule is applied to all sites simultaneously
- the quantity K in every cell is chosen from a finite set
- time, space as well as the cell contents are discrete values
- the cells are arranged in some regular topology (e.g. array, vector)
- the evaluation neighborhood size R remains constant

In this paper we will discuss a number of extensions on these minimum requirements. These include the accumulation of all past behaviour in a cell (the introduction of feedback "from the past") and multi-level automata. These are discussed in further sections.

A CA evolves in time by evaluating the value of all cells within the context of recent history. That is: the next value of a cell A(i) at position i is a function of: its previous value, AND the previous values of cells within its immediate neighborhood R, AND the nature of the rule. The neighborhood of a cell normally consists of the cell itself plus a number of immediately adjacent cells.

$$A[i](t) \rightarrow \text{rule } \{A[i-r](t-1), A[i-r+1](t-1), .., A[i](t-1), .., A[i+r](t-1)\}$$

Typically, binary automata are used (R=1) with a neighborhood K=2. The pioneering work of Stephen Wolfram has been of extreme value since it led to the identification of interesting rules and, most of all, to the classification of families of automata (Wolfram 83). Extensive empirical studies give the impression that all cellular automata appear to belong to four possible families. Roughly sketched, the 4 classes have the folowing behavioural qualities:

class 1: The automaton disappears completely after (a typically small) number of generations. The initial random structure is "eaten up" by the system.
class 2: Evolution leads to simple, fixed or pulsating, highly periodic structures. The changes occur within finite limits.
class 3: Evolution leads to chaotic patterns, which appear to contain very special instances of organized structures. The degree of structural fluctuation increases at a regular pace.
class 4: This family of automata feature strange attractors: structures appear to propagate irregularly. Structural contraction and expansion is completely irregular and unpredictable.

We explored the behaviour of automata from all four classes (refer to illustrations). In the following sections, we will focus on a number of extensions to the minimum requirements of automata definition. These include, amongst other things: history tracking, the application of 2D rules on 1D automata, systems with feedback and systems with more than one rule level.

4. Examples

4.0. Continuous automata.

At the start, every cell of a one dimentional vector holds a 8-bit random number. To move from the current generation to the next, the following algorithm is applied simultaneously to every cell:
• the AND rule: the local sum – given the current neighborhood – is computed and logically ANDend with the value in the center cell, if the result is true, the center cell value is increased else decreased.
• the RULE$ rule: A convenient rule string must first be constructed to accomodate all possible evaluations of the algorithm. The rule size is computed as follows:
rule size = nr_of_different_values -1 * neighborhood_size + 1
The rule size corresponds to the maximum value of any local sum. This sum becomes a pointer in the rule string, it selects a new value to update the center cell. A visualization of the global change – showing the degree of periodicity from one generation to the next – is given below [Fig.1].

4.1. Linear automata with 2-d rules.

An unusual scheme was devised to be able to apply 2-d rules in a linear environment. The initial configuration consists of 3 vectors, not just one. Instead of only using left and right neighbors in neighborhood evaluation, we extend spatial sensivity into the recent history of the automaton and into the next generation. The previous generation provides the cell NORTH of the center cell in the Von Neuman neighborhood to be applied. The cell SOUTH of the center cell is borrowed from the generation to follow. Updating proceeds in steps of 2 generations at once, as seen in the illustrations [Fig. 2].

4.2. Wave propagation.

The basic approach here was to create structural evolution from the interaction of moving particles. The overall effect is the emergence of fading waves in a 2-dimentional field. The idea is to "pick up" the complex fluctuations of local energy at a certain location in this wave field by pointing to it with a mouse. At the start, all cells participate with maximum energy, and initially move in any of 8 possible directions. We may then stirr the cells, imposing a preference direction on some of them. What to do if any two live cells meet is determinded by a rule table. In other words, the transition table defines the nature of the waves to be produced. The Moore neighborhood is used to compute the sum of all cells surrounding the center cell in question. Additional control issues from the way in which summing takes place; for instance:
• the sum may be increased if a neighbor is not zero, or
• the sum is computed by adding all gradients in a local neighborhood

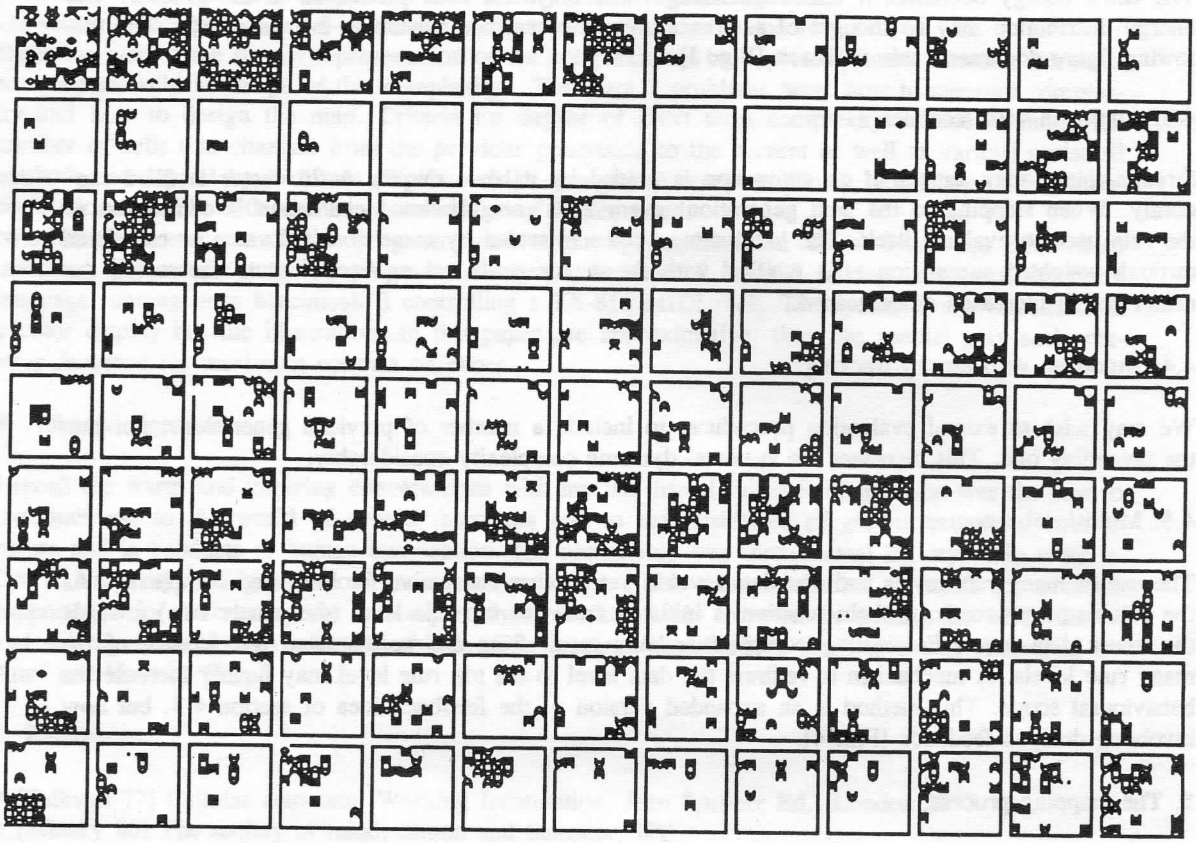

Figure 3.
Wave propagation example.

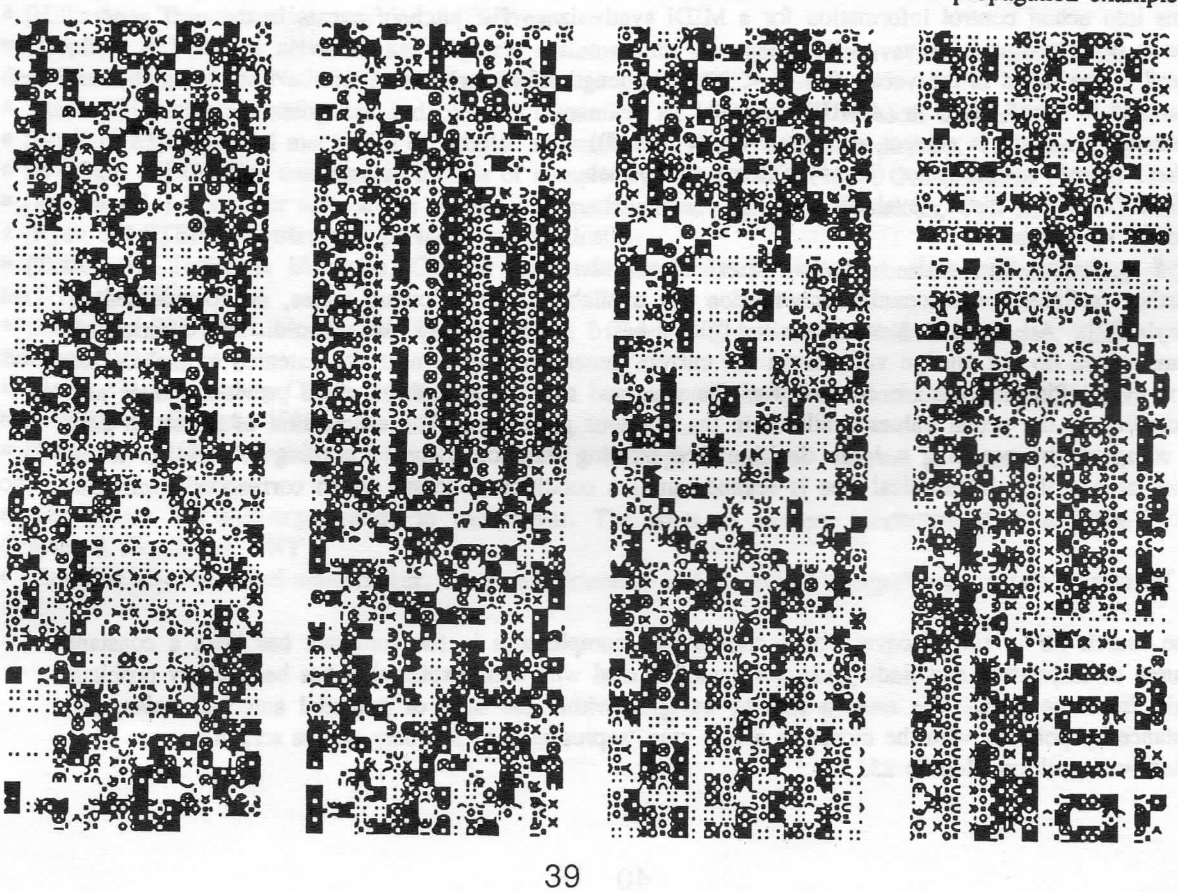

Figure 4. Automaton with one rule level and 3 data levels

The cell's energy decreases if it doesn't change, and only live cells participate in the process. The energy distribution may be thought of as a measure of the surface tension in the wave field. The following figure documents this approach [Fig. 3].

4.3. Automata with feedback.

Strange things may happen if an automaton is guided by its own output. A 2d vector is filled randomly. When stepping to the next generation, every cells' neighborhood evaluation is considered as the rule used to evaluate itself. The local sum – often re-scaled by some constant value or calculated through weighted summation – is ANDed with the center cell's value. According to the result, the center cell is increased or decreased.

4.4. Automata with history tracking.

We may wish to extend evaluation procedures to include a number of previous generations, not just the preceding one. This increases the systems' dynamic complexity considerably.

4.5. Multi-level automata.

The use of many rule levels and one data level is yet another alternative for increased complexity. As the automaton proceeds, a chain reaction is initiated from the top rule level (the master rule) towards the lower data level. Fascinating complexities may result from the accumulated interference of the many rule levels. A mechanism to redirect the data level to the top rule level may further increase the behavioural scope. This method is an expanded version of the feedback idea of section 4.3. but now involving delayed feedback [Fig. 4].

5. The mapping process.

Our basic intention is to explore the behavioural wealth of CA for real-time composition and performance. Therefore, we need a flexible mapping device to transpose the dynamics of these virtual systems into actual control information for a MIDI synthesizer. The pitch of events in the next generation is computed as follows:
item# = the index in the vector (i.e. 0 < item# < length-of-vector)
channel# = item# MOD nr-of-MIDI-channels
element-from-scale = current-scale (new-vector (item#))
offset1 (item#) = old-vector (item#) if item# else 0 then
offset2 (item#) = history evaluation
root = <constant>
key# = element-from-scale + root + offset1 + offset2
Further facilities for automatic orchestration are available through lookup tables, as documented in [Beyls 87]. All items that exist (not = 0) are heard simultaneously, as a chord. The duration of items which do not change value from the current generation to the next are extented to include the next event. Pitch offsets are computed as the weighted sum of the indices of all previous values and according to the actual values available in the previous generation. The articulation of events in time is computed by scanning a large decision tree offering duration values according to whether items exist or not. The hierarchical tree is scanned until a condition is found which corresponds with the actual contents of the vector.

7. Interaction and implementation.

The search for – and discovery of – tremendous complexities in the automata has been a constant source of inspiration and finding elegant means to deal with their exploration has been a key problem right from the start. We need a tool to navigate within the field of potential automata rules, for instance by pointing with the mouse in a rule map (represented as a square on the screen):
rule# = x-position Modulo 255

old-vector = compute one bit configuration from y-position for binary vector
old-vector = compute many 8-bit values from y-position for continuous vector
Since we want some sort of representation of the complexity to be expected from a tentative selection, we need to build an image of this complexity. There are 2 problems here: how to measure complexity and how to design the map. Criteria for degree of short term complexity include counting the number of cells that changed from the previous generation to the current as well as various statistical methods developped by Wolfram (Wolfram 84). Long term complexity may be analysed using autocorrelation or FFT, both methods currently being implemented.
Non-real-time versions of the program were originally implemented on a Symbolics 3600 Lisp machine. The current automata programs are written in Mach 2, a multi tasking version of the Forth language running on a Macintosh II controlling a TX-816 MIDI rack. The dynamics are visualized on a color display but the illustrations in this paper are approximative: they use special gray scale patterns designed for maximum contrast on paper.

8. Acknowledgement.

I recall the warm and inspiring conversations with my late friend Julian Sullivan, who was the first to introduce me to the world of cellular automata and to the intricacies of growth processes both in nature and in products of human imagination. His partnership was instrumental for my early work in CA at Univerity College London, some ten years ago. I also want to thank Luc Steels for strategic support and for introducing me to complex dynamics and Ludo Cuypers for writing the MIDI driver and solving many system related problems. The research described in this paper was sponsored by Atari and Yamaha Corp.

9. References.

- [Sullivan 77] Cellular automata. Working Information, Jean Spencer Ed., London 1977
- [Minsky 86] The society of mind, Simon and Schuster, NY
- [Hewitt 77] Viewing control structures as patterns of passing messages. Artificial Intelligence 8, 3 North Holland Pub. Co.
- [Hillis 86] The connection machine. MIT Press
- [Hopfield 82] Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences USA 1982.
- [Steels 88] Artificial Intelligence and complex dynamics. AI Memo 88-2, Brussels Univerity
- [Wofram 83] Statistical mechanics of cellular automata. Reviews of modern physics, Vol. 55, nr. 3
- [Wolfram 84] Cellular automata as models of complexity. Nature Vol. 311, October 1984
- [Wolfram 84] Computer software in science and mathematics. Scientific American, Vol. 251, nr. 3
- [Ryan 89] STEIM, Amsterdam, personal communication
- [Toffoli 87] Toffoli & Margolus, Cellular Automata Machines, a new environment for modelling, MIT Press 1987
- [Knowlton 79] EXPLOR manual, Bell Labs, NJ - [Beyls 80] Action, Exhibition catalogue, Kindt Editions, Belgium
- [Beyls 87] Introducing Oscar, Proceedings of the International Computer Music Conference, Feedback Editions, Cologne 1987
- [Deneubourg 86] Probabilistic behaviour in ants: a strategy of errors? Journal of theoretical biology, 105.
- [Babloyantz 88] Self-organization in biosystems. The roots of modern biochemistry, Walter De Gruyter & Co. Berlin - NY
- [Dawkins 86] The blind watchmaker. Longman Scientific and Technical Pub., Harlow