Cellular Automata Music Composition: a bio-logical inspiration

Eduardo Reck Miranda,
composer, electronic music expert, artificial intelligence and music researcher

1. Introduction and motivation

Music has always been an interesting domain for the application of new scientific discoveries inviting composers to combine artistic creativity with scientific methods. "Today it is becoming increasingly common for the composer to turn to the sciences to supplement his or her compositional model" [9]. On the other hand scientists also seem to show interest in the organisational principles found in music (see [9] for a discussion).

We are particularly interested in promoting such interdisciplinary activities. Our motivation is twofold. On the one hand it is believed that scientific models carry an important component of human thought, namely formal abstraction, which can be very inspiring for music composition. On the other hand we would like to raise certain questions like: "What can be the justification for using science as a compositional tool?", or "Which aspects of science are applicable to music and how it can be done?". Obviously there are no simple answers for these and indeed we do not intend to provide any here. We believe though that each artist should be able to make her or his own judgements. As far as these questions are concerned, the work introduced in this paper is to be regarded only as a contribution for empirical experimentation.

Our research work [11][12][13][14] attempts to identify correlation among different disciplines such as biology, crystallography, and computing, in order to investigate the possibility of composing music inspired by a framework of interdisciplinary knowledge.

We have selected a class of mathematical models known as *cellular automata* (CA) to play the central role in this research due to the fact that they have been used to model a wide range of scientific phenomena. During the past three decades scientists have been investigating and developing CA [21]. Although very simple they can provide models for a wide variety of complex phenomena in physics (eg. dynamic and chaotic systems), biology (eg. genetics), and chemistry (eg. chemical reactions and crystal growth) [23].

This paper introduces an experimental system for cellular automata music composition (CAMUS, for Cellular Automata MUSic). It begins with a brief introduction to cellular automata basics and shows how we attempted to map their behaviour into musical aspects. Then it defines the underlying concepts of the system and introduces its functioning, illustrating its operation and musical aspects through an example composition. The paper ends with conclusion and further work.

Other systems have been designed in order to investigate the use of CA in music ([10][7][1][8] to cite a few). Unfortunately it is not the scope of this document to review them. However we recommend the study of these alternative approaches.

Before we begin, we would like to remind the reader that this is a speculative work. That is to say that we are aware that most of the assumptions made below in order to define what we called "cellular automata music" are pragmatic and subjective.

2. Introduction to cellular automata (CA)

CA are mathematical idealisations of systems in which space and time are discrete and quantities take on a finite set of discrete values. A cellular automaton consists of a regular array with a discrete variable at each site, referred to as a cell. The state of a cellular automaton is specified by the values of the variables at each cell. It evolves in synchrony with the tick of an imaginary clock according to an algorithm (i.e. a set of rules) which determines the value of a cell according to the value of its neighbourhood [22][21][4]. As implemented on a computer, the cells are rectangles in the screen whose states are represented by different colours.

CA were originally introduced in the sixties by von Neumann and Ulan [3] as a model of biological self-reproduction. They wanted to know if it were possible for a machine to reproduce, that is, to automatically construct a copy of itself. Their model consisted of a two dimensional grid of cells, each of which is in one of a number of states (which represented the components out of which they built the self reproducing machine). Controlled completely by the algorithm designed by its creators, the machine (i.e. a pattern of cells in the grid) would extend an arm into a virgin portion of the universe (the grid), then slowly scan it back and forth, creating a copy of itself.

Since then CA have been reintroduced over and over again and applied to a considerable variety of purposes. Various interesting algorithms were developed along these 30 years. From many different types of CA algorithms in existence nowadays, two were selected for use in this system: *Game of Life* (GL) designed by John Horton Conway (a University of Cambridge mathematician), and *Demon Cyclic Space* (DCS) designed by David Griffeath (of the University of Wisconsin at Madison) [21][4].

2.1. The *Game of Life* cellular automaton (GL)

GL is a finite two-dimensional lattice of squared cells whose states (0 or 1) are influenced by the states of neighbouring cells. Time is also discrete and from one tick of a virtual clock to the next, each cell is either *alive* (1) or *dead* (0) depending on the following algorithm:

i. if a cell is *dead* at time **t**, it comes *alive* at time **t+1** if, and only if, exactly **3** of its neighbours (i.e. fewer than **4** AND more than *2*) are *alive* at time **t**;

ii. if a cell is *alive* at time **t**, it comes *dead* at time **t+1** if, and only if, fewer than **2** OR more than **3** neighbours are *alive* at time **t**.

With this rule acting everywhere on GL's lattice, an initial configuration of live cells may either grow interminably, fall into cyclic patterns, or eventually die off.

CAMUS's implementation enables the user to design its own rules other than Conway's original rule, as we shall see later in our example composition (see §6.2.)

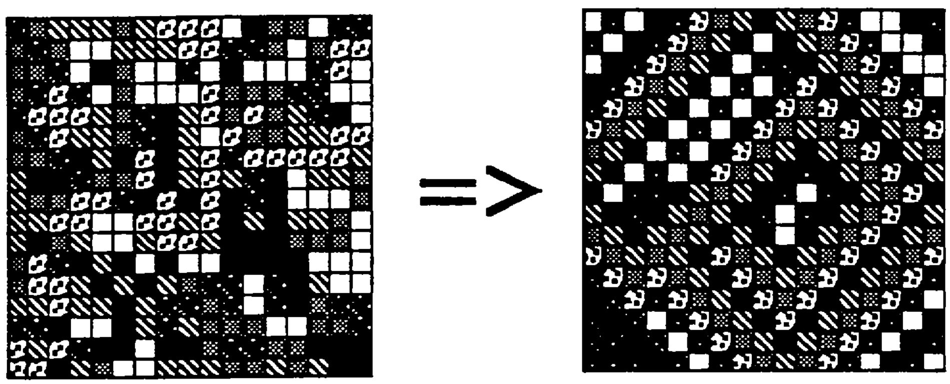
2.2. The Demon Cyclic Space cellular automaton (DCS)

The DCS algorithm designed by D. Griffeath generates a miniature universe of incredible complexity. Initialised as a random distribution of coloured cells, it always end up with stable, angular spirals reminiscent of strange crystalline growths (Fig. 1).

In order to produce this striking phenomenon of scientific interest and beauty DCS relies on a very simple algorithm. There can be any number **n** of states, each represented by a different colour, which are numbered from **0** to **n-1**. A cell that happens to be in state **k** at one tick of the clock must dominate, by the next tick, any adjacent cells that are in state **k-1**. Domination here means a change of state of the adjacent cell from **k-1** to **k**. This algorithm resembles a natural chain: a cell in state **2** can dominate a cell in state **1** even if the latter is dominating a cell in state **0**. But as it is a cyclic space the chain has no end, i.e. a cell in state **0** dominates neighbouring cells that are in state **n-1**.

Both, GL and DCS, cannot be infinitely extensible. Therefore they ought to be modelled as a squared lattice drawn onto a torus (see [11] or [13] for more details).

Fig. 1: Initialised to a random state the DCS cellular automaton always ends up with stable, angular spirals reminiscent of crystalline growths.



3. CAMUS fundamentals

Since CA produce large amounts of patterned data and music composition can be thought of as based on pattern propagation and formal manipulation of its parameters, it is not a surprise that music researchers started to suspect that CA could be translated (mapped) into a music representation in order to generate compositional material. In the following paragraphs we explain how CAMUS does this mapping.

3.1. Geometric representation of triads

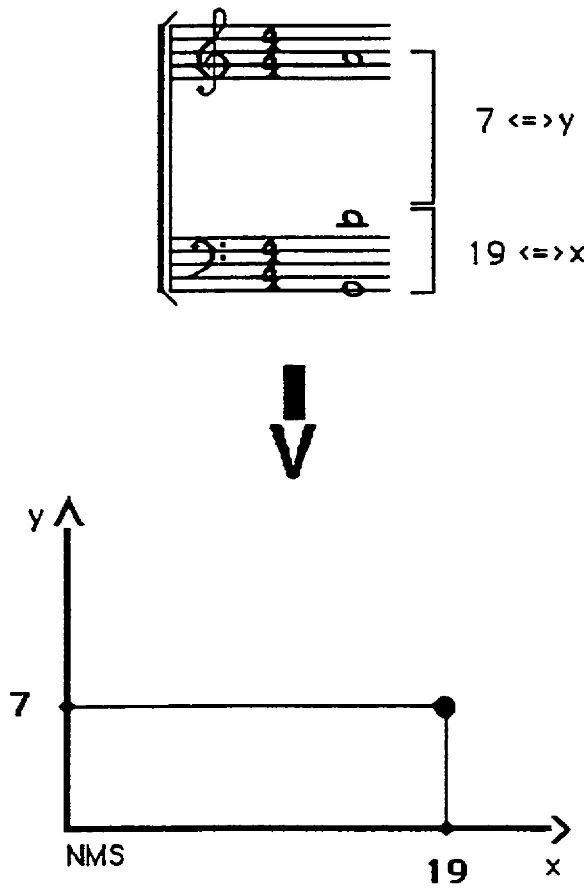
Consider a set of triads, where the term "triad" refers to a sub-set of three elements out of a tune-system set. The tune-system set may be regarded as a discrete framework around which the musical events take place. Its elements (pitches) can be identified as being on a lattice. The tune-system adopted to carry out the study in this paper is the 12 tone equal-tempered system. However other tune-systems may also be used (§7).

The set of all triads having the same interval series is regarded as a transpositionally equivalent class of triads. Any triad in a transpositionally equivalent class may be transformed into another by an operation known as "ordered transposition" (concept borrowed from Allen Forte [6]). This concept is very important for CAMUS because it suggests a geometrical model in two dimensions for representing triads. Each dimension of this model corresponds to a specific order position in the interval series, and is quantified to represent the full range of intervals that could span a pair of pitches in this system (Fig. 2). We named this model for music representation as *von Neumann Music Space* (NMS) in memory of John von Neumann. See [11] or [13] for the mathematical specification of it.

3.2. Mapping CA into a NMS

A Cartesian co-ordinate of a CA cell can be viewed as an address to a point into a NMS.

Fig. 2: The NMS horizontal co-ordinate represents the first triad's interval whereas the vertical co-ordinate represents its second interval.



Consider the following example [3][11][13]:

i. the finite automaton

(SG, sg(0), f)

where

SG = a set of control signals,
sg(0) = the ground state, and

f = local transition function;

ii. the set of control signals

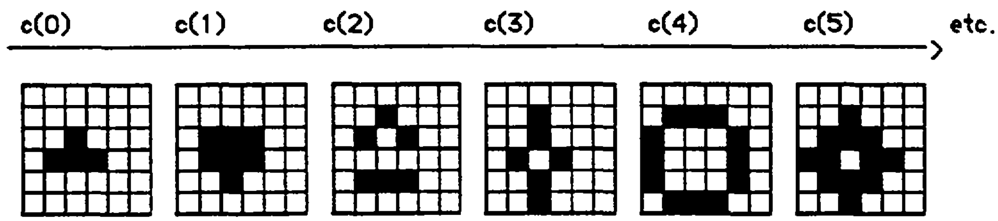
SG = {sg(0), sg(1)}

such that

sg(**0**) = **0**, and
sg(**1**) = **1**.

The above example represents a finite automaton which can assume two states: either a quiescent (0) or a non-quiescent (1) state. A configuration **c: X x Y -> SG** defines a set of cells where the elements are Cartesian co-ordinates of a two-dimensional interval space corresponding to non-quiescent state cells of **c**. A global transition function **F**, also called a *transition rule*, drives the propagation **c(0)**, **c(1)**, **c(2)**, **...**, **c(t)**. Each configuration **c** corresponds to a set of triads (Fig. 3).

Fig. 3: Propagation of patterns arising from an initial configuration **c(0)**.



In musical terms, we would say that the propagation of a configuration **c** under a certain global transition function **F** corresponds to the macro-formal organisation of the musical discourse. The micro-formal organisation in turn is defined by the internal organisation of each cell and between cells at a certain time **t**. These concepts were inspired by Xenakis' *book of screens* used as a graphical representation of sonic events in a slice of time [24].

3.3. The musical engine

CAMUS performs two different finite automata simultaneously mapped into a NMS. One is the GL cellular automaton which can assume two states: 0 (dead) or 1 (alive). It is responsible for the aforementioned micro-formal organisation of the composition. Given the fundamental pitches, GL works out a set of triads for each time step **t**. The other is the DCS cellular automaton which can assume **n** states (**n** is specified by the user). DCS is responsible for the orchestration of the composition, i.e. each state corresponds to an instrument designated to perform a cell. These two automata work in parallel (Fig. 4).

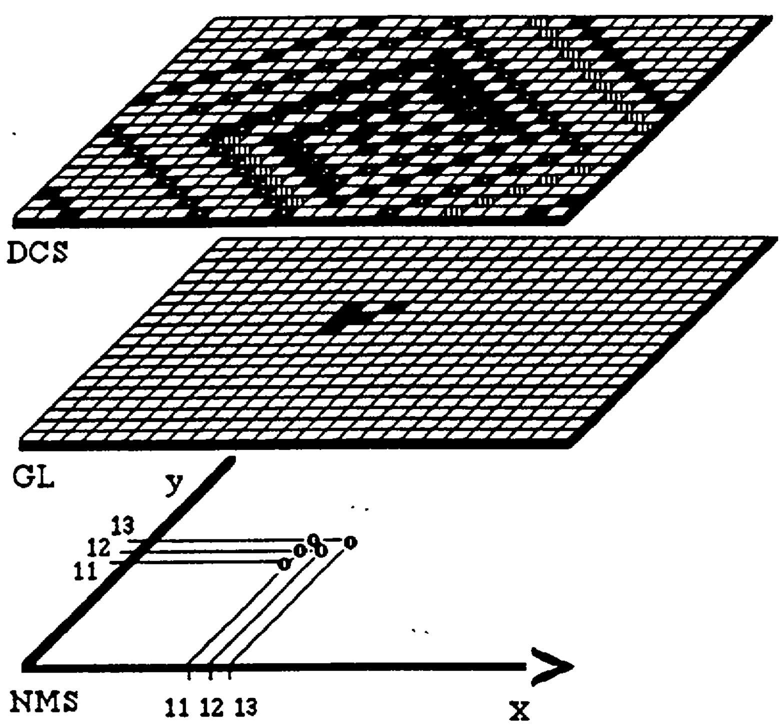
4. Cellular genetic code and temporal reasoning

Each musical cell has its own timing but the pitches within a cell can assume different durations and be triggered at different times. We say that pitches within a cell form a certain abstract shape according to a predefined CAMUS codification which is explained below.

4.1. The cellulAr geNetic coDe (AND)

Inspired by the DNA molecule in the cell's nucleus, whose symbols, drawn from an alphabet of four different chemical bases (Adenine, Guanine, Cytosine, and Thiamine), form strands coiled up into a helix, the "genetic" information of a musical cell is formed by a string (in the form **Tgg -> Dur**) whose symbols have temporal meaning. Like the nucleotide pair, the AND string (also referred to as "cellular typology" in [11] and [13]) has two components: **Tgg** (trigger component) and **Dur** (duration component). Each of these components has a certain *temporal code* which can be one out of 10 possibilities (Fig. 5).

Fig. 4: The musical engine of CAMUS consists of two different finite automata, namely GL and DCS, simultaneously mapped into a NMS.



As an example, consider the following AND: **[dna] -> a[dn]** (Fig. 6). To understand how AND has temporal meaning, imagine the above example in a time domain representation (Fig. 7). The first AND string's component, **Tgg**, stands for the trigger values, and the second, **Dur**, stands for the duration values.

Fig. 5: There are 10 different temporal codes. The combination of two forms the cellular genetic code.

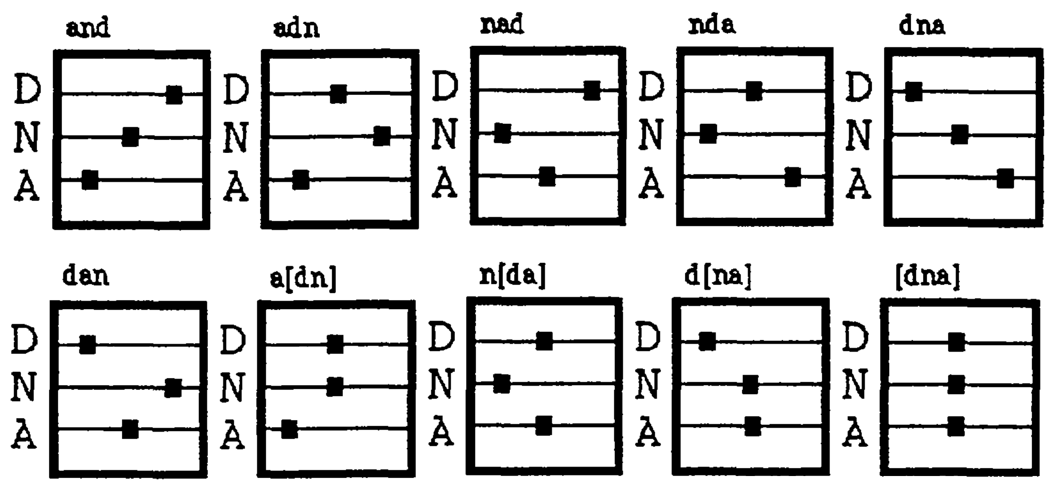


Fig. 6: Example typology [dna] -> a[dn].

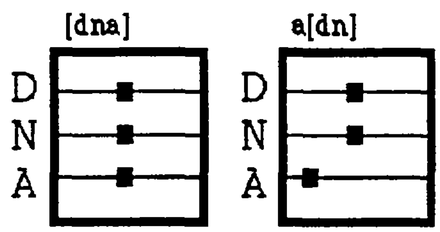
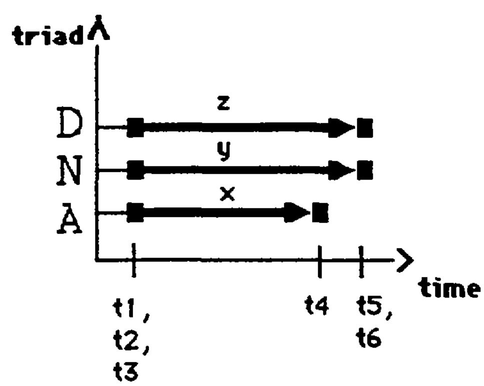


Fig. 7: Time domain representation of [dna] -> a[dn], where Tgg={t1, t2, t3} and Dur={t4, t5, t6}.



4.2. Finding out the AND

AND is deduced by the GL cellular automaton. Considering that each cell to be performed corresponds to a GL's non-quiescent cell mapped to a NMS we now propose a mechanism to find out the AND string.

Assume a 4-digit binary codification as follows:

and => 0101
adn => 0010
nad => 0111
nda => 1011
dna => 0011
dan => 0110
a[dn] => 0000
n[da] => 1111
d[na] => 1001
[dna] => 0001

Then the **Tgg** and **Dur** components are defined by:

Tgg ::= abcd | dcba
Dur ::= mnop | ponm

where **a, b, c, d, m, n, o, p** $\in$ **{ 0, 1 }.**

A cell's AND is specified by the neighbouring cells (Fig. 8). Considering the geometrical relation below, the string is calculated as follows:

a = (m, n-1)
b = (m, n+1)
c = (m+1, n)
d = (m-1, n)
m = (m-1, n-1)
n = (m+1, n+1)
o = (m+1, n-1)
p = (m-1, n+1)

Note that the AND in fact represents the abstraction of a shape. The actual numerical values for each individual pitch's trigger (t1, t2, and t3 in Fig. 7) and duration (x, y and z in Fig. 7) are calculated when the cell is being performed. CAMUS calculates these values based on a distribution formula [5] selected by the user.

Sometimes, by chance, there will be cases where one or more pitch durations may overlap the duration of a cell (Fig. 9).

Fig. 8: The neighbouring cells and their geometrical relationship.

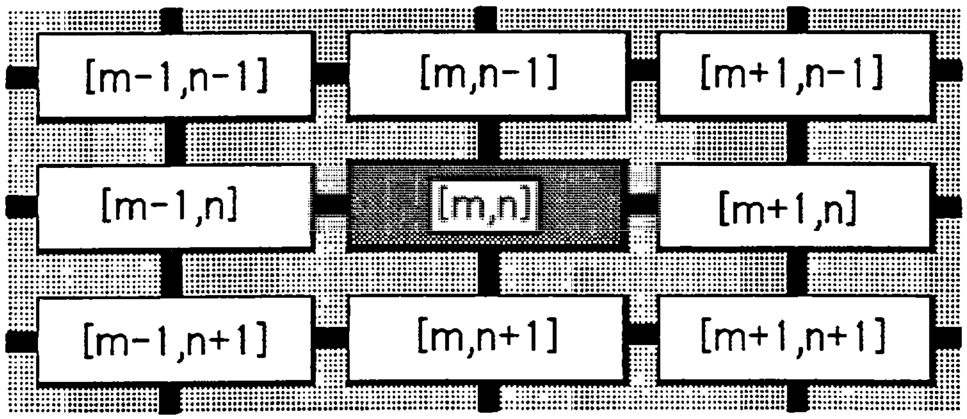
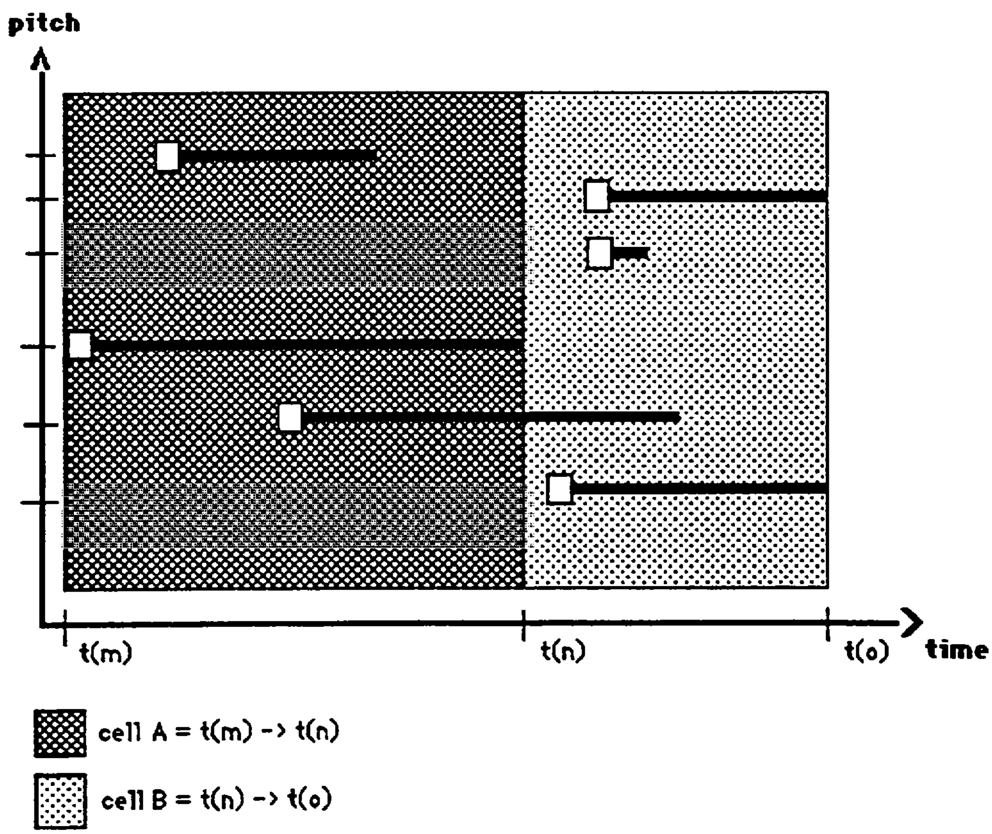


Fig. 9: Sometimes one or more pitch durations may overlap the duration of a cell.



5. The implementation

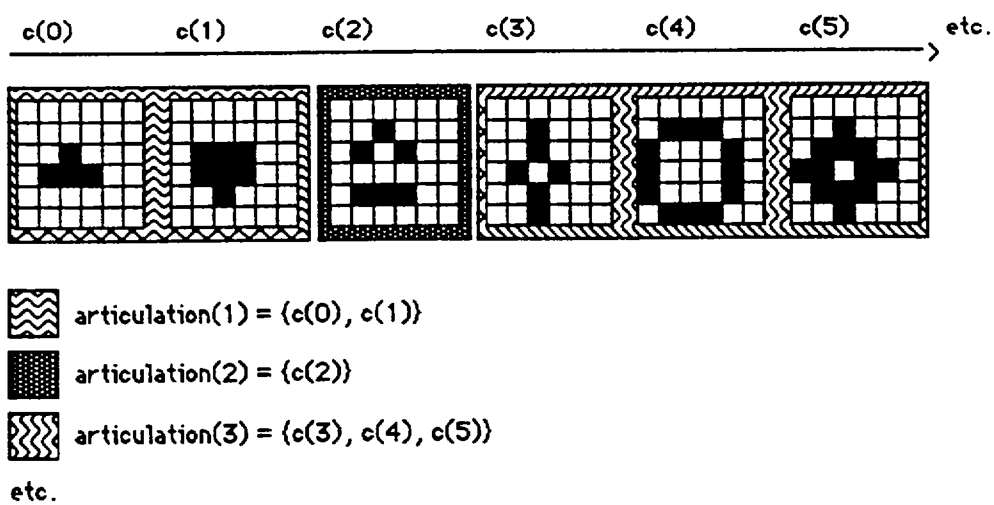
The current implementation (CAMUS V1.0) generates MIDI (Boom, 87) output from the evolution of the musical engine mentioned above.

The GL is responsible for the pitch selection. Each cell corresponds to a triad where the first pitch comes from a predefined sequence of MIDI pitch numbers defined by the user and the remaining two are determined by the corresponding cell's NMS co-ordinates (each co-ordinate corresponds to an interval from the previous pitch). The GL is also responsible for the AND deduction as we have explained before. The default GL algorithm is the one originally designed by Conway (see §2.1.). However other variants may be tried (§6.2.). The user is asked to "draw" the initial GL configuration.

The DCS, which is responsible for the orchestration, determines which MIDI channel will be used to output a cell. The orchestration value, i.e. the number of DCS states, is given by the user.

Different articulations may be specified to take place throughout a certain number of configurations **c(n)**. Articulation here means a group of configurations sharing certain common characteristics. As we mentioned before, each cell of a cellular automaton changes its state in synchrony with a tick of an imaginary clock. In turn, each tick (or step) forms a configuration of cells to be performed. After specifying the articulation loop in terms of number of steps (i.e. how many configurations will be performed within a loop), the user also specifies how many articulations will take place there (i.e. defines groups of steps). Each articulation in turn has to be specified with its own speed (i.e. how fast its elements will evolve), dynamics (i.e. how loud its objects should sound), and fundamental pitch sequence (i.e. sequence of pitches to be used as the basis of "triads") (Fig. 10). Note that bandwidth values are also specified for speed and dynamics. This is to say that actually CAMUS generates a value within these bandwidths for each cell by means of a chosen distribution formula (the same used to calculate the values of an AND, mentioned in §4.2.).

Fig. 10: Different articulations may be specified within a loop.



6. An example composition

In this section we present an example composition using CAMUS.

6.1. Musical thinking in CAMUS

CAMUS works on four dimensions of parametric control: *time* x *pitch* x *dynamics* x *orchestration*.

Time, pitch, and dynamics work according to the user defined setup for each articulation of the piece. The orchestration is defined by the number of DCS cellular automaton states (see §2.2 and §3.3.) where each state corresponds to a MIDI channel.

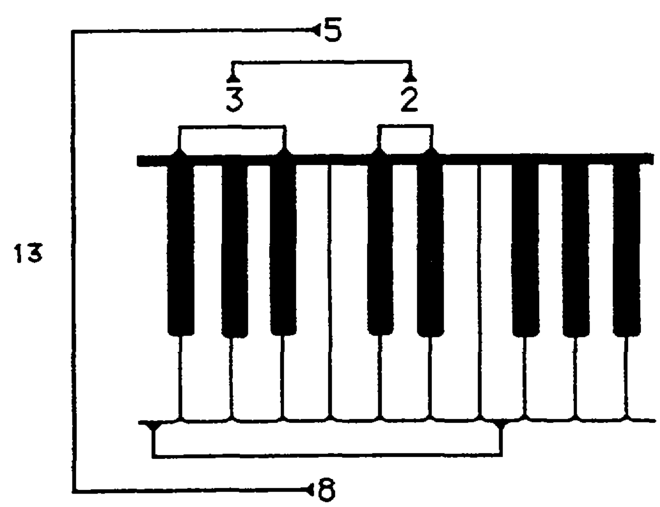
Musical form is also important here. "Form has to do with placing musical materials in time" [15]. It is possible to identify 2 levels of formal control in CAMUS. The lowest level has to do with micro-formal organisation and the highest level with macro-formal organisation. Micro-formal organisation is entirely controlled by the musical engine (explained in §3.3). However it depends on the macro-formal organisation specified by the user, that is the initial CAMUS setup, such as the GL's rules, the initial GL configuration (drawn by the user), the orchestration value, number of articulations, the distribution formula, and so forth.

6.2. *Quadratura Circuli*

Quadratura Circuli is a CAMUS aided composition for 3 instruments.

A numeric phenomenon suggested by the piano keyboard has inspired the author to figure out the parameters to initialise CAMUS. Fig. 11 illustrates a curious relationship between the piano keyboard layout and the early Fibonacci numbers: 2, 3, 5, 8, and 13. Notice that there are 13 keys within an octave: 8 white keys and 5 black keys. The black keys in turn are divided in two groups of 3 and 2 keys respectively.

Fig. 11: Relationship between the piano keyboard and the early Fibonacci numbers.

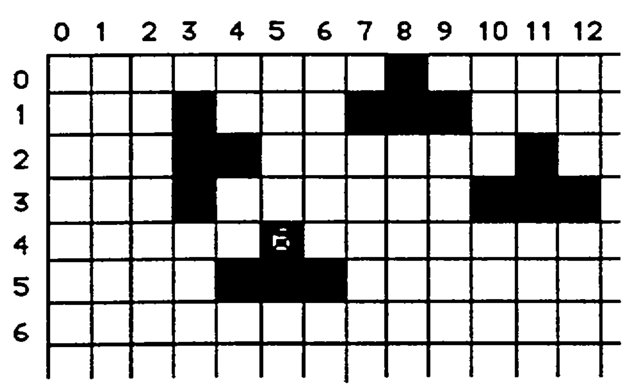


Bearing these numbers in mind CAMUS was set up for *Quadratura Circuli* as follows:

1. loop length = **5** configurations
2. no. of articulations = **2** (one of 2 configurations and other of 3 configurations)
3. life rule = fewer that **5** AND more than **2**
4. death rule = fewer than **2** OR more than **3**
5. orchestration = **3** (MIDI channels 0, 1, and 2)
6. distribution = uniform
7. articulation 1:
    - 7.1. start time = configuration **1** (i.e. c(0))
    - 7.2. end time = configuration **2** (i.e. c(1))
    - 7.3. speed = **8532**
    - 7.4. speed bandwidth = **1323**
    - 7.5. dynamics (MIDI key velocity) = **85**
    - 7.6. dynamics bandwidth = **32**
    - 7.7. fundamental pitches (MIDI numbers) = **{22, 32, 38, 52, 53, 23, 33, 28, 58, 55, 25, 35}**
8. articulation 2:
    - 8.1. start time = configuration **3**
    - 8.2. end time = configuration **5**
    - 8.3. speed = **8813**
    - 8.4. speed bandwidth = **882**
    - 8.5. dynamics (MIDI key velocity) = **82**
    - 8.6. dynamics bandwidth = **35**
    - 8.7. fundamental pitches (MIDI numbers) = **{53, 52, 55, 85, 82, 33, 32, 35, 25, 83, 23, 22}**

Fig. 12 illustrates the pattern given for the initial configuration **c(0)** of the GL cellular automaton algorithm.

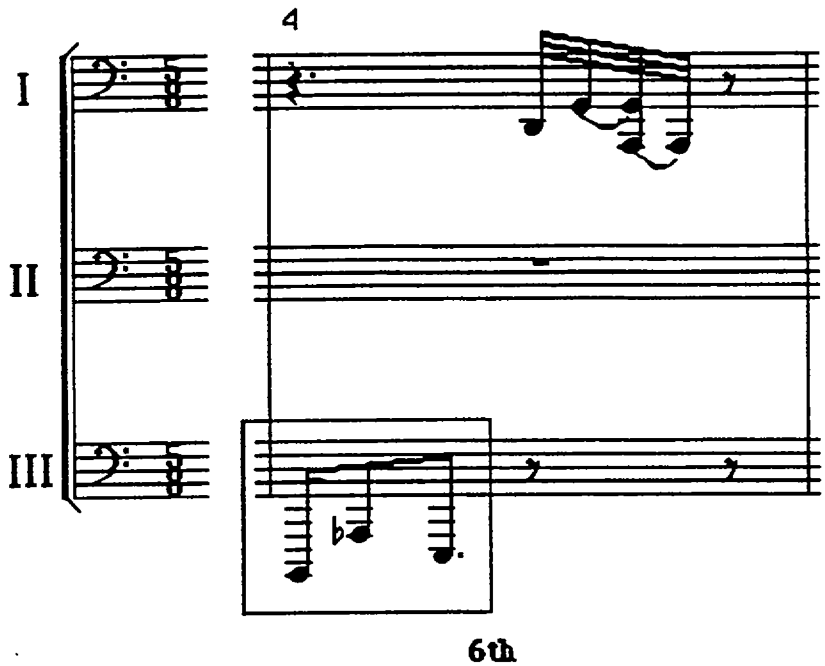
Fig. 12: The initial GL configuration of *Quadratura Circuli.*



6.3. A micro-analysis of the generated score

Fig. 13 shows the 4th bar of the generated score which has the 6th and 7th triads (or cells) of the initial configuration in it. We will examine the 6th cell in detail.

Fig. 13: The fourth bar of *Quadratura Circuli.*



CAMUS processes the cells of the lattice (Fig. 12) from top-down to left-to-right. Thus the 6th cell corresponds to the NMS co-ordinate (5,4). The fundamental pitch of this cell is B1, i.e. the 6th MIDI value of the sequence specified for articulation 1. Consequently the second pitch is E2 (i.e. five semitones above B1) and the third pitch is A2b (i.e. four semitones above E2:

fund(6) = **B1**, i.e. MIDI(23),
x(6) = **E2**, i.e. MIDI(28)<= 23+5,
y(6) = **A2b**, i.e. MIDI(32)<= 28+4.

The Tgg(6) and Dur(6) values (see §4.2.) are 0100 (**adn**) and 0101 (**and**) respectively. Therefore AND(6) = **adn** -> **and**. The **adn** code means that pitches are triggered in this order: **fund(6)**, **y(6)** , and **x(6)** (i.e. B1 -> A2b -> E2). The **and** code states that the duration of **fund(6)** is to be shorter than **x(6)**, which in turn is to be shorter than **y(6)** (i.e. B1 < E2 < A2b).

The DCS cell in the co-ordinate (5,4) is in a state which corresponds to the third instrument (i.e. MIDI channel 2).

Finally, the loudness of each of these pitches (not shown in the score of Fig. 13) oscillates between ***P*** and ***f*** (i.e. MIDI key velocity value between 177 (85+32) and 53 (85-32).

7. Conclusion and further work

This paper introduced an experimental system for music composition inspired by cellular automata. We did not intend to provide a general system nor a language for composition but an implementation of a speculative idea for empirical investigation.

Any systematisation of a technique for composition followed by its computer implementation leads to serious compositional limitations. No doubt CAMUS is limited to a narrow world of musical possibilities, yet despite its limited scope, it worked very well. The musical results sometimes are very interesting. Thus we regard it as a plausible starting point for further investigation.

The author has used CAMUS to compose both musical passages to be used in major works (eg. *The Turning of the Tide*, a live electronics piece reviewed in [18]) and pieces whose materials were entirely generated by the program (eg. *Entre l'absurde et le mystère*, for symphonic orchestra, premiered in Edinburgh in 1992). In the former the author explored other "tune-system sets" (§3.1.) than the 12 tone equal-tempered. In this case CAMUS' MIDI output was mapped to a micro tuned FM module synthesiser (the Yamaha's TX81Z). It also controlled a sampler with large scale sound events (several seconds of complex sounds). In the latter the author used the output score as a source material for a hand made orchestration where pitch classes and shapes provided by the AND were preserved.

At the moment we are expanding the scope of these ideas to sound synthesis. We are working on a version which attempts to implement a model of granular synthesis [16][17][20] by means of CA [14][12].

Acknowledgements

The author would like to express his gratitude to Peter Nelson who read a draft of this paper and made valuable suggestions.

This research is funded by a CAPES (Brazilian Research Agency) grant.

References

[1] **Beyls**, P. (1991), "Self-Organizing Control Structures using Multiple Cellular Automata", in ICMC Montreal 1991 Proceedings, pp. 254-257, Canada.

[2] **Boom**, M. (1987), "Music through MIDI", Microsoft Press, USA.

[3] **Cood**, E. F. (1968), Cellular Automata, Academic Press, UK.

[4] **Dewdney**, A. K. (1989), "A cellular universe of debris, droplets, defects and demons", in Scientific American, August/1989, pp. 88-91, USA.

[5] **Dodge**, C. , and **Jerse**, J. A. (1985), "Computer Music: synthesis, composition and performance, Shirmer Books, UK/USA.

[6] **Forte**, A. (1973), "The Structure of Atonal Music", Yale University Press, USA/UK.

[7] **Hunt**, A., **Kirk**, R., and **Orton**, R. (1991), "Musical Applications of a Cellular Automata Workstation", (unpublished paper), Music Technology Group, University of York, UK.

[8] **Katrami**, A. I., **Kirk**, R., and **Myatt**, A. (1991), "Manipulation of Cellular Automata and Fractal Landscape Mappings", in ICMC Montreal 1991 Proceedings, pp. 106-109, Canada.

[9] **Landy**, L. (1991), "What's the Matter with Today's Experimental Music?", Contemporary Music Studies Vol. 4, Harwood Academic Publishers, Switzerland.

[10] **Miller**, D. (1990), "Cellular Automata Music, in ICMC Glasgow 1990 Proceedings, pp. 314-317, UK.

[11] **Miranda**, E. R. (1991), "Cellular Automata Music investigation", Music Technology MSc dissertation, Dept. of Electronics, University of York, UK.

[12] **Miranda**, E. R. (1992), "Neural Nets, ChaOs and Formants", unpublished discussion draft, Faculty of Music, University of Edinburgh, UK.

[13] **Miranda**, E. R. (1993), "Cellular Automata Music: An Interdisciplinary Project", in Interface, Vol. 22, No. 1, pp. 3-21, Swets & Zeitlinger, The Netherlands.

[14] **Miranda**, E. R., **Nelson**, P., and **Smaill**, A. (1992), "ChaOs: a model for Granular Synthesis by means of Cellular Automata", in Annual Report and Project Directory, Edinburgh Parallel Computing Centre, UK.

[15] **Orton**, R. (1990), "Music fundamentals", unpublished Music Technology lecture notes, University of York, UK.

[16] **Roads**, C. (1978), "Granular Synthesis of Sounds", in Computer Music Journal, Vol. 2, No. 2, pp. 61-68, MIT Press, USA.

[17] **Roads**, C. (1991), "Asynchronous Granular Synthesis", in Representation of Musical Signal, De Poli, G. et al. (Editors), MIT Press, USA.

[18] **Smaill**, A. (1993), "New Music in Edinburgh" (concert review), in Computer Music Journal, Vol. 17, No. 1, pp. 66-67, MIT Press, USA.

[19] **Tang**, P. C. L. (1984), "On the Similarities between Scientific Discoveries and Musical Creativity: A Philosophical Analysis", in Leonardo, Vol. 17, No. 4, pp. 261-268, Pergamon Press, UK.

[20] **Truax**, B. (1988), "Real Time Granular Synthesis with a DSP Computer", in Computer Music Journal, Vol. 12, No. 2, pp. 14-26, MIT Press, USA.

[21] **Wilson**, G. (1988), "The life and times of cellular automata", in New Scientist, 08/October/88, pp. 44-47, UK.

[22] **Wolfram**, S. (1983), "Statistical mechanics of cellular automata", in Reviews of Modern Physics, Vol. 55, No. 3, pp. 601-623, USA.

[23] **Wolfram**, S. (1984), "Universality and Complexity in Cellular Automata", in Physics, Vol. 10, pp. 1-35, Elsevier Science Publishers, The Netherlands.

[24] **Xenakis**, I. (1971), "Formalized Music, Indiana University Press, USA.