Evolving Cellular Automata Music: From Sound Synthesis to Composition

Eduardo Reck Miranda
Sony Computer Science Laboratory Paris
6 rue Amyot, 75013 Paris, France
miranda@csl.sony.fr

Abstract

This paper focuses on issues concerning musical composition practices whereby the emergent behaviour of cellular automata is used to model generative processes for synthesised sound and musical forms. We introduce two cellular automata-based systems, Chaosynth and CAMUS, that we have designed for our investigation and discuss their performance and role in the composition of a number of professional pieces of music. Chaosynth is a granular synthesis system whose parameters are controlled by a variant of a cellular automaton that has been used to model Belousov-Zhabotinskii chemical reactions. CAMUS is a composition system that takes advantage of the pattern propagation properties of cellular automata in order to generate musical forms.

1 Introduction

In this paper we suggest that the discussion as to whether or not computers can compose music is no longer relevant: computers can compose if programmed accordingly. Perhaps the greatest achievement of Artificial Intelligence to date lies in the construction of machines that can compose music of incredibly good quality indeed; Cope's EMI system comes to mind here (Cope 1991). It should be noted, however, that these AI systems are good at mimicking well known musical styles. They are either hard-wired to compose in a certain style or are able to learn how to imitate a style by looking at patterns in a bulk of training examples. Such systems are therefore good for imitating composers of musical styles that are well-established, such as medieval, baroque or jazz. Conversely, issues such as whether computers can create new kinds of music are much harder to study, because in such cases the computer should neither be embedded with particular models at the outset nor learn from carefully selected examples. Furthermore, it is hard to judge what the computer creates in such circumstances because the results normally sound very strange to us. We are often unable to judge these computer-generated pieces because they tend to lack those cultural references that we normally hold on to when appreciating music.

One plausible approach to these problems is to program the computer with abstract models that embody our understanding of the dynamics of some compositional processes. Since the invention of the computer, many composers have tried out mathematical models which were thought to embody musical composition processes, such as combinatorial systems, stochastic models and fractals (Dodge and Jerse 1985; Xenakis 1971; Worral 1996). Some of these trials have produced interesting music and much has been learned about using mathematical formalisms and computer models in music composition. The potential of Artificial Life modelling is therefore a natural progression for computer music research. This paper introduces and discusses two systems that we have designed for our investigations: Chaosynth and CAMUS. Whereas the former employs a cyclic, self-organising cellular automata to control a sound synthesiser (Miranda, 1995; 1998), the latter uses a pattern propagation type of cellular automaton to generate musical structures (Miranda, 1993, 1994; McAlpine *et al.* 1999).

2 Background

The mathematical basis for the concepts described in this paper can be found in Cood's classic book *Cellular Automata* (Cood, 1968). In short, cellular automata are normally implemented on a computer as a regular array or matrix of variables, or cells, which normally can have one, two or three dimensions. Each cell may assume values from a finite set of integers and each value is normally associated with a colour. The functioning of such an automaton is monitored on the computer screen as a sequence of changing patterns of tiny coloured cells, according to the tick of an imaginary clock, like an animated film. At each tick of the clock, the values of all cells change simultaneously, according to a set of transition rules that takes into account the values of their neighbourhood. A wide variety of cellular automata and transition rules have been devised and adapted for a variety of modelling purposes in different areas from Physics and Biology to Chemistry and Social Sciences. Cellular automata have attracted our interest here because of their organisation principles. We are particularly interested in those cellular automata that display *cyclic behaviour*, *self-organisation* and/or *pattern propagation* properties. A further discussion on the importance of these properties for music is introduced in the author's latest book *Composing Music with Computers* (Miranda 2001).

By way of related work we cite an early attempt at a cellular automata-based sound synthesis system described by Hunt and co-workers in a paper presented at the International Computer Music Conference in 1991 (Hunt *et al.* 1991). Also refer to the work of Bilotta and Pantano presented at this workshop.

3 Controlling the evolution of complex sounds with Chaosynth

For our investigation into cellular automata sound synthesis we selected the granular synthesis technique. Granular synthesis is essentially a time-based synthesis technique (Miranda 1998): it works by generating a rapid succession of very short sound bursts called granules (e.g., 35 milliseconds long) that together form larger sound events (Figure 1). This synthesis technique can be metaphorically compared with the functioning of a motion picture in which an impression of continuous movement is produced by displaying a sequence of slightly different images at a rate above the scanning capability of the eye. Granular synthesis bears a close resemblance to neo-impressionist painting, whereby entire pictures were produced using small touches of unmixed colour, as if the intention of the artist was that colours should mingle in the spectator's eye, rather than on the painter's palette (Figure 1). The sounds from granular synthesis tend to exhibit a great sense of movement and flow.

Chaosynth is a cellular automata-based sound granules generator, whereby each granule is represented by a specific configuration of a cellular automaton. In this case, a set of transition rules F determines the evolution of the sound granules, as the propagation $c^{t+1} = F(c^t)$ evolves.

After a number of experiments, the cellular automaton ultimately chosen for Chaosynth is ChaOs (an acronym for Chemical Oscillator). ChaOs is our own variant of a cellular automaton that has been used to model the behaviour of a type of catalytic chemical reaction known as Belousov-Zhabotinskii reactions (Scott 1989). In our case, the cellular automaton models the way in which most natural sounds produced by acoustic instruments evolve: they tend to converge from a wide distribution of their partials to form oscillatory patterns.

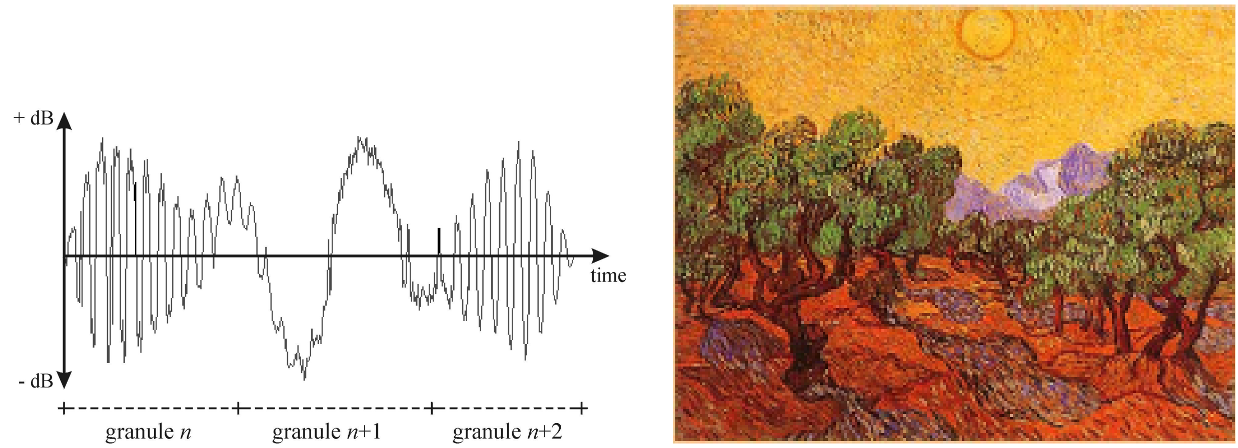

Figure 1: The unmixed rapid succession of short sounds of a granular synthesiser mingles in the ear of the listener (left) in the same way that the unmixed small touches of a neo-impressionist painting mingle in the eye of the spectator, as in Van Gogh's paiting *Olive Trees* (right).

ChaOs can be thought of as a matrix of cells containing identical simple electronic circuits, whose behaviour could be compared to that of an artificial neurone, or perceptron (Hertz *et al.* 1991). At a given moment, the cells can be in any of the following states: *polarised*, *depolarised* or *collapsed*. The model embodied by this automaton can be described as follows: a cell interacts with its neighbours (normally eight neighbours) through the flow of electric current between them. There are minimum $V_{\min}$ and maximum $V_{\max}$ threshold values which characterise the state of a cell. If its internal voltage V_i is under $V_{\min}$, then the cell is polarised. If it is between $V_{\min}$ (inclusive) and $V_{\max}$ values, then the cell is depolarised. Each cell has a potential divider, composed of two resistors R_1 and R_2, which is aimed at maintaining V_i below $V_{\min}$. But when it fails (i.e., when V_i reaches $V_{\min}$) then the cell becomes depolarised; the higher the value of V_i, the more polarised is the cell. There is an electric capacitor *k* which regulates the rate of depolarisation, but once a cell is depolarised, the tendency is to become more and more depolarised. When V_i reaches $V_{\max}$ the cell collapses. A collapsed cell at time *t* is automatically restored to a polarised state at time *t*+1. The automaton tends to evolve from an initial random distribution of states in the grid of cells towards an oscillatory cycle of patterns (Figure 2).

Assume a discrete set of *n* states φ_n, such that $n \geq 3$. The definition of the states of the cells can be formalised as follows:

- if $V_i < V_{\min}$ then φ_0
- if $V_i \geq V_{\min}$ and $V_i < V_{\max}$ then φ_p, where $0 < p < n-1$
- if $V_i \geq V_{\max}$ then φ_{n-1}

In practice, the state of the cells is represented by a number between 0 and $\varphi - 1$, where φ is the amount of different states that a cell can assume (remember that, in theory, φ can be equal to any integer higher than 3). All states in between exhibit a degree of depolarisation; the closer the cell's state value gets to $\varphi - 1$, the more depolarised the cell becomes.

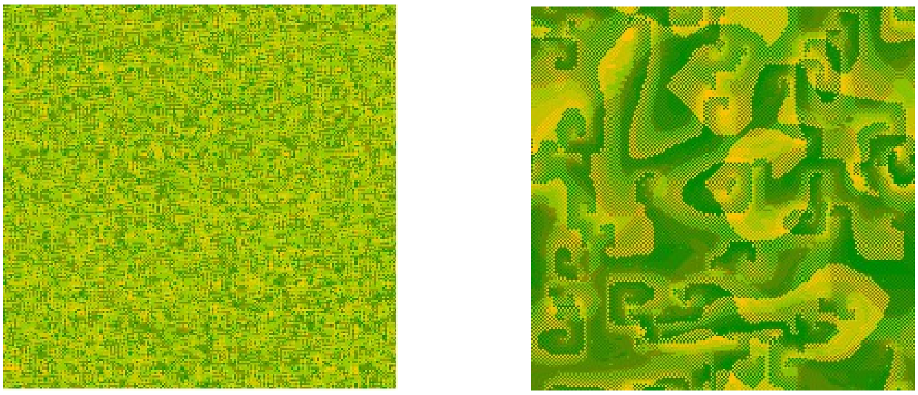

Figure 2: From an initial random distribution of states in the grid of cells (left), ChaOs usually evolves to form an oscillatory cycle of patterns (right).

The set *F* is defined by three rules, simultaneously applied to each cell, selected according to its current state, as follows:

- if polarised: The cell *a* may or may not become depolarised at time *t*+1. This depends upon the number of polarised cells *P* and the number of collapsed cells *C* in its neighbourhood (normally 8 neighbours), and the resistance of the automaton to depolarise its cells: $\varphi^{t+1}(a) = \left(\frac{P}{R_1}\right) + \left(\frac{C}{R_2}\right)$

- if depolarised: The tendency of a depolarised cell *a* is to become more depolarised as time elapses. The rate of the depolarisation depends upon the value of the capacitance *k* of the nerve cell and the degree of depolarisation of the neighbourhood: $\varphi^{t+1}(a) = \left(\frac{S}{P}\right) + k$, where S is the amount of neighbours considered.

- if collapsed: $\varphi^{t+1}(a) = \varphi_0$, that is, a collapsed cell at time *t* becomes polarised at time *t*+1.

Once implemented on a computer, the behaviour of ChaOs is determined by setting up the following parameters:

- the dimension of the grid (i.e., the total amount of cells *a*)
- the number of φ cell states
- the neighbourhood *S*
- the resistors R_1 and R_2
- the capacitor *k*

3.1 How can a configuration of cells represent a sound granule?

Each sound granule produced by Chaosynth is composed of several partials and each partial is a sinewave produced by an oscillator (in fact, we could use any other waveform here, but for the sake of simplicity for the experiments we have used only sinusoids). An oscillator o^n needs three parameters to function, namely frequency (f_z), amplitude (ω_z) and duration (d_z) of the partials. ChaOs controls these parameters as follows: the states φ of the cells are associated with frequency and amplitude values and a number of oscillators o^n are associated with sub-sets of cells (e.g., $o^1 = \{a^1, a^2, a^3, a^4\}, o^2 = \{a^5, a^6, a^7, a^8\}$, etc.). An example of a grid of 400 cells associated with 16 oscillators of 25 cells each is shown in Figure 4. Each sound granule therefore results from an additive synthesis process whereby all oscillators simultaneously produce sinewaves at each configuration $c^0, c^1, c^2, \ldots c^n$ of ChaOs cells (Figure 3). The frequencies and amplitudes of these sinewaves are determined by the arithmetic mean over the frequencies and amplitude values associated to the states of their corresponding cells (Table 1). Mathematically, the frequency F_i^n and the amplitude Amp_i^n values for each signal generator i during configuration c^n are determined by the arithmetic mean over the frequency and the amplitude values associated with the states of the cells of their corresponding sub-sets of cells during this cycle:

$$F_i^n = \frac{\left\{\sum_{a=1}^{A} f_a^n\right\}}{A} \quad \text{and} \quad Amp_i^n = \frac{\left\{\sum_{a=1}^{A} p_a^n\right\}}{A}$$

where f_a^n and p_a^n are the frequency and amplitude of cell a during configuration c^n and A is the total amount of cells in the subset. The total duration of a sound is given by the total number of configurations and the duration of the individual granules; for example, 100 configurations of 40 millisecond granule would result in a sound event of 4 seconds duration.

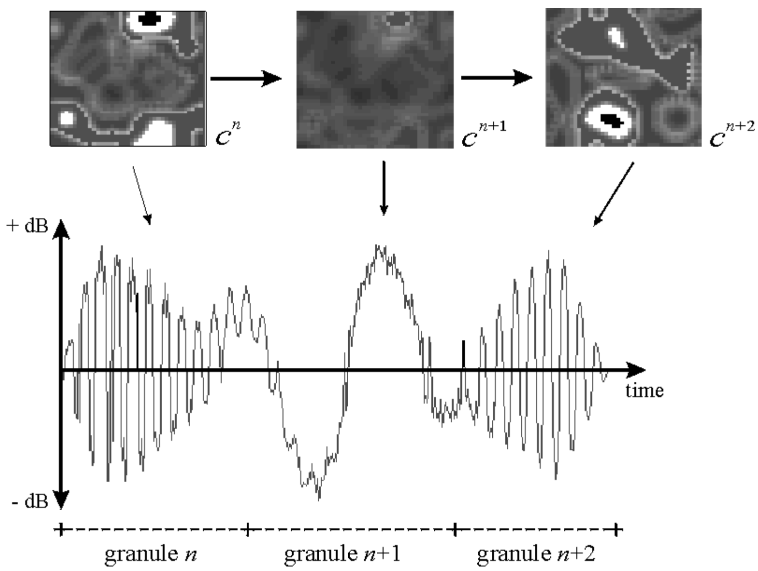

Figure 3: Each configuration of the cellular automaton generates a sound granule.

The duration of a whole sound event is determined by the total number of configurations that the automata produces and by the duration of the individual granules. For instance, 100 configurations of 30 millisecond granules each result in a sound event of 3 seconds duration. The duration of the individual granules is set beforehand and it can vary according to a linear function.

Cell state	Value	Colour	Frequency	Amplitude
m_0	0	white	110 Hz	0 dB
m_1	1	red	220 Hz	-3 dB
m_2	2	blue	330 Hz	-6 dB
…	…	…	…	…
m_n	$n-1$	$l(n)$	$f(n)$	$p(n)$

Table 1: An example of frequency and amplitude allocations to cell states.

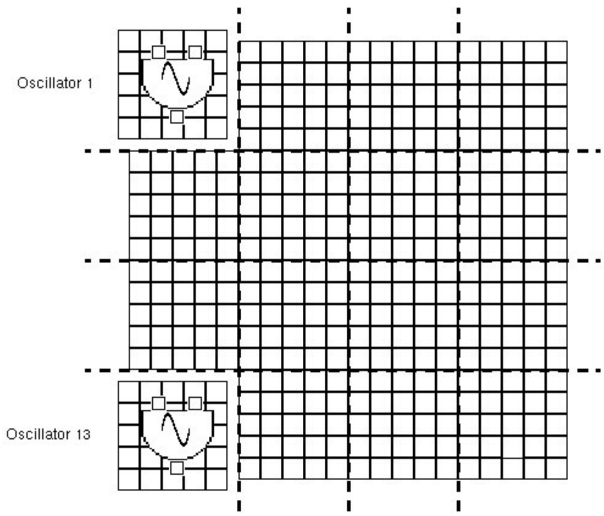

Figure 4: An example of an arrangement of oscillators associated to cells.

3.1 Commentary on the results

The cellular automaton generates sound whose morphological evolution resembles the morphological evolution of sounds produced by most acoustic instruments: their partials converge from a random distribution to oscillatory patterns. The random initialisation of the states in the grid of cells produces an initial random distribution of frequency and amplitude values, which tend to settle into an oscillatory cycle.

We have synthesised sound using up to 64 different states, $\{\varphi_1, \varphi_2, \ldots \varphi_{64}\}$, that is, up to 64 different frequency and amplitude values, and up to 64 oscillators on grids of up to 4,000,000 cells. The results tended to exhibit a great sense of natural movement and flow, and yet, most of these synthesised sounds are not readily found in the "real" acoustic world. Some results bear remarkable resemblance to the sounds of flowing water, birds, frogs and insects.

Variations in tone colour are achieved by varying the set of frequency values associated with the different cell states. For example, if the frequency set contains values lower than 200 Hz, then the result will be a sound in the lower band of the audible range. Conversely, if the set contains frequencies above 800 Hz, then the result will be a brighter sound in the higher band of the audible range.

The size of the grid and the amount of cells per oscillator are largely responsible for the degree of granularity of the spectral variation: a larger number of cells per oscillator produces finer granulation effects, whilst a smaller number produces coarser granulation effects.

The length of the individual granules also plays a key role in the overall result. The acoustic effect of the variation of the length of the individual granules can be summarised as follows: very short lengths (about 35 milliseconds) produce textures of sparkling bubble-like clouds, whereas greater lengths (about 800 milliseconds) produce sequences of sound strokes; lengths above 1 second produce sequences of musical notes of varying timbres.

Different rates of transition from random to oscillatory patterns are obtained by changing the values of the resistors R_1 and R_2, and the value of the capacitor k of the cellular automaton.

The author used Chaosynth to generate sounds to compose *Olivine Trees*, a prize winning electroacoustic composition composed at the University of Edinburgh's Parallel Computing Centre and Faculty of Music. *Olivine Trees* was recently featured on the BBC Radio 4 *Nature* programme, presented by Mark Cowardine, as an example of evolutionary music modelling (14 May 2001).

4 Propagating musical pattern sequences with CAMUS

In a previous paper we demonstrated how music can be thought of as a form of pattern propagation (McAlpine *et al.*, 1998). Music can be appreciated at many different levels. Whilst for some it is sufficient that a composition possesses a pleasant melody that can be hummed along to, others may be more concerned with the temporal development of the piece: how initial sound structures evolve from their introduction through to the work's conclusion, and so on. For example, we may view each theme or motif in a composition as a separate pattern. As the composition progresses, the patterns are subjected to certain transformations, such as transposition, inversion, augmentation, and so forth, according to the formal structure that the composer chooses for the work. This structure can be rigidly adhered to or used as a general guiding principle, but as long as certain design constructs are in place to guide the temporal development of the composition, we can say that we have a system of pattern propagation according to some predetermined constraints.

We have devised a number of experiments in order to study whether cellular automata that exhibit pattern propagation behaviour could be used or adapted to model the propagation of musical patterns. These experiments culminated in a system, named CAMUS (short for Cellular Automata MUSic). CAMUS uses two types of cellular automata: the classic *Game of Life*, devised by John Horton Conway, and Demon Cyclic Space, designed by David Griffeath (Dewdney, 1985; 1989). In this paper we will focus only on the role of the former.

The Game of Life cellular automaton consists of a matrix of cells, which can exist in two states: alive (represented by 1) or dead (represented by 0). The original transition function devised by Conway to determined the evolution of the automaton can be summarised in two simple rules that are applied simultaneously to all cells of the matrix:

- if a cell is dead at a time *t*, it comes alive at time *t*+1 if it has exactly 3 neighbours (out of a neighbourhood of 8 cells) alive

- if a cell is alive at time *t*, it becomes dead at a time *t*+1 if it is has fewer than 2 or more than 3 neighbours alive

CAMUS' implementation of the Game of Life allows for the specification of rules other than Conway's original rules. In most cases, an initial configuration of live cells may either grow infinitely, fall into a cyclic pattern sequence or eventually die of (Figure 5).

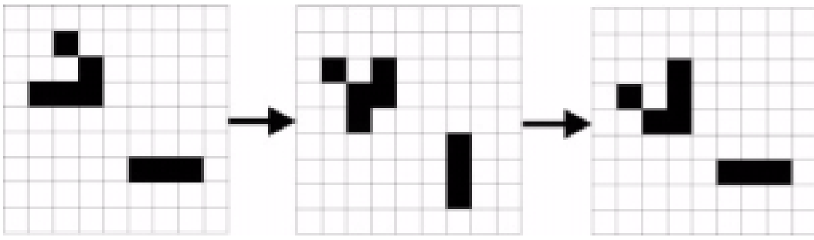

Figure 5: The Game of Life in action.

4.1 Cellular automata music modelling

CAMUS uses Cartesian space in order to represent musical triplets; a musical triplet in the context of this research is a set of three musical notes: $\{A, N, D\}$. The space has two dimensions, where the horizontal co-ordinate represents the distance between *A* and *N*, and the vertical co-ordinate represents the distance between *N* and *D*. These distances can either be expressed in terms of half-tones (i.e., semitones) or quarter-tones; in this paper we use half-tones for all examples (Figure 6).

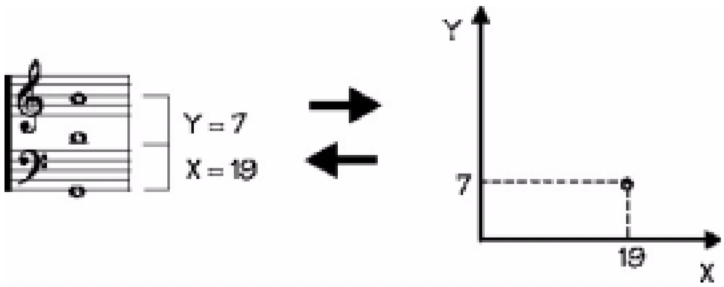

Figure 6: CAMUS represents musical triplets in Cartesian space.

The musical model is defined as follows: consider the automaton $\{M, m_0, f\}$, such that $M = \{m_0, m_1\}$, where $m_0 = 0$ and $m_1 = 1$. Let *P* and *Q* denote sets of integers such that for all $p \in P$ corresponds to the values of the horizontal co-ordinate and for all $q \in Q$ corresponds to the values of the vertical co-ordinate of the matrix of cells. A configuration $c: P \times Q \to M$ therefore defines a set of triplets in two-dimensional Cartesian space, such that $\{a \in P \times Q | m(a) \neq m_0\}$ is finite. A set of rules *F* drives the production of configurations of triplets (Figure 7). In this case, the propagation of configurations can be considered as a kind of macro organisation of musical forms in time.

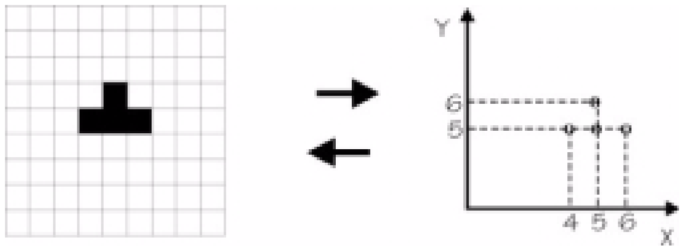

Figure 7: Each configuration of the Game of Life automaton produces a number of triplets.

To begin the compositional process, the system is set up with a starting configuration. At each time-step, the co-ordinates of each live cell are analysed and used to determine a triplet. For example, in the case of Figure 7, the cell at co-ordinates (5, 5) is alive, and thus constitutes a triplet. The co-ordinates (5, 5) describe the distances between the notes of the triplet. Assuming that a fundamental note *A* is given (it is not important here to know where this notes comes from), then the closest note *N* will be five semitones above the fundamental, and the furthest note *D* will be ten semitones above the fundamental (i.e., five semitones above *N*). By default, cells are played from left to the right and from top to bottom, but the user can change this order.

Once the notes for each cell have been determined, the states of their neighbouring cells are used to calculate the relative starting time and duration of each note of the triplets, according to a time representation scheme called *AND-code* (Figure 8).

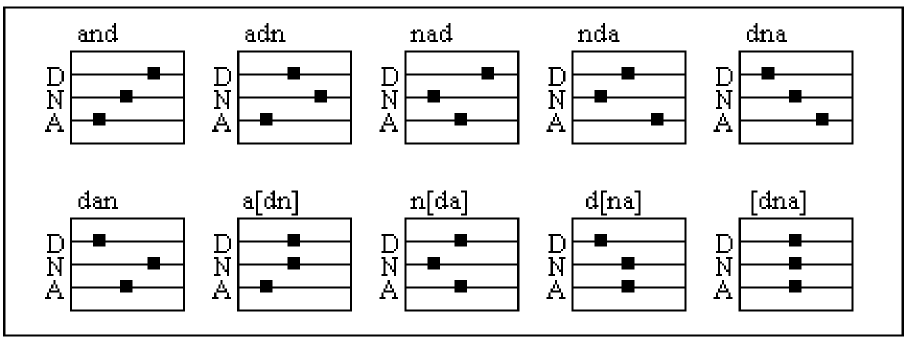

Figure 8: The time representation scheme used to shape the temporal relationship between the notes of a triplet.

In order to determine the AND-code of a cell $a(p,q)$, CAMUS firstly builds a list $S = [s_1, s_2, \ldots, s_8]$ containing the states of the 8 neighbouring cells. Next, the system establishes four 4-bit words from *S* according to the following scheme:

$$w_1 = s_1 s_2 s_3 s_4$$
$$w_2 = s_4 s_3 s_2 s_1$$
$$w_3 = s_5 s_6 s_7 s_8$$
$$w_4 = s_8 s_7 s_6 s_5$$

The format of the AND-code is $Tgg + Dur$ and it has 8 bits. The Tgg part of the code corresponds to the triggering time of the notes of the triplet and the Dur part corresponds to their duration. The system performs an 'inclusive or' operation to compute both parts of the AND-code as follows:

$$Tgg = w_1 \vee w_2$$
$$Dur = w_3 \vee w_4$$

Each half of the code is inferred by means of the following criterion:

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

where a denotes the fundamental note of the triple, n the middle note and d the upper note. The letters within square brackets indicate simultaneity of events.

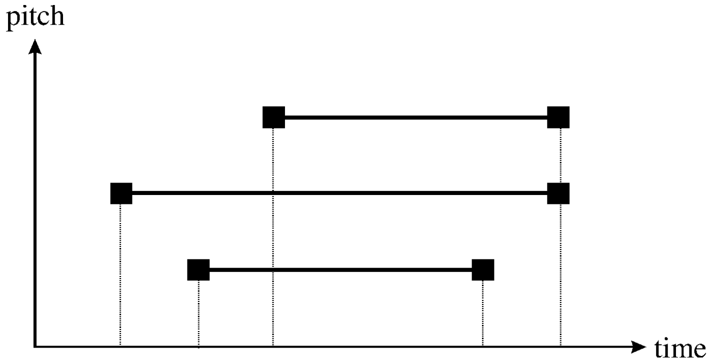

Figure 9: An example of a time-template for the spatial organisation of a triplet. The horizontal axis represents time and the vertical axis pitch.

As an example, consider again the case of cell (5, 5) in Figure 7. Firstly the system forms the list S = [0, 1, 1, 1, 0, 0, 0, 0] according to the state of its neighbourhood. Then, the system matches 0111 with w_1 for calculating Tgg and 0000 with, say, w_3 for calculating Dur. The AND-code for this cell therefore values 0111000. The abstract time-template for the internal organisation of this cell's triplet therefore is $nad \rightarrow a[dn]$ (Figure 9). The actual trigger and duration values in seconds are calculated according to a complementary procedure that is not essential to understand at this stage. An example of a musical passage that could be generated by this single cell is illustrated in Figure 10.

4.2 Commentary of CAMUS

CAMUS deals with musical forms at two levels: at the level of the internal organisation of the triplets and at the level of the external organisation of the triplets in time. Whilst the internal organisation of a triplet is largely dictated by the AND-code, the macro-organisation level is controlled by the Game of Life cellular automaton (and by a few other mechanisms of secondary importance that are not discussed in this paper).

Despite the arbitrariness of the musical representation embodied by CAMUS, we have come to conclude that cellular automata are appropriate for generating musical forms. We observed that CAMUS can produce interesting musical sequences. Indeed, a number of professional pieces have been composed using CAMUS-generated sequences, such as *Entre o Absurdo e o Mistério*, for chamber orchestra (Miranda 2001) premiered by The Chamber Group of Scotland in Edinburgh (UK) and *Grain Streams*, for piano and electroacoustics, premiered by pianist Jerôme Valet in Annecy (France). Both pieces have been performed and well-received at other public venues.

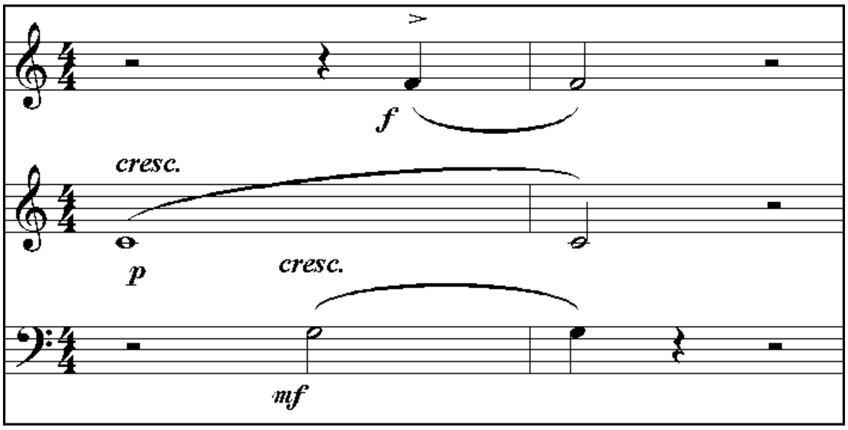

Figure 10: An example of a musical passage generated by a single cell using the template portrayed in Figure 9.

5 Results

The results of the cellular automata experiments are very encouraging. They are good evidence that both synthesised sounds and abstract musical forms can be successfully modelled using cellular automata. It should be noted however that it is not our primary aim to model existing musical styles or mimic existing acoustic musical instruments. Rather, we are looking for global principles governing both existing and *would-be* (after Casti 1997) musical forms and sound.

Chaosynth and CAMUS should be regarded as two successful case-studies and good starting points for further experimentation. Perhaps the next step would be to integrate both types of systems in order to study the relationship between abstract musical forms and the micro-organisation of their contents (in this case synthesised sounds) within the context of cellular automata.

In general, we found that Chaosynth produced more interesting results than CAMUS. We think that this might be due to the very nature of the phenomena in question. The inner structures of sounds seem more susceptible to cellular automata modelling than larger musical structures. As music is primarily a cultural phenomenon, in the latter case we need to think seriously about employing modelling techniques that take into account the dynamics of social interaction and cultural evolution; e.g. (Bull *et al.* 2001). In this case, we might need to apply modelling methods where phenomena (in our case, musical styles) emerge autonomously from interactions of a population of agents. We are currently investigating this possibility (Miranda 2000).

Chaosynth is available for Windows and Macintosh platforms, and is manufactured and distributed by Nyr Sound: <http://www.nyrsound.com>. CAMUS is freeware and there are two versions available, one of which uses three-dimensional cellular automata (not discussed in this paper). Both versions of CAMUS were implemented for Windows platforms by Kenny McAlpine, formerly of Glasgow University. Both versions of CAMUS are available on the accompanying CD-ROM of the author's latest book *Composing Music with Computers* (Miranda 2001).

References

Bull, L., Holland, O. and Blackmore, S., "On the Meme-Gene Coevolution", *Artificial Life*, Vol. 6, pp. 227-235, 2001.

Casti, J. L., "Would-be worlds: toward a theory of complext systems", *Artificial Life and Robotics*, 1(1), 1997.

Cood, E. F., *Cellular Automata*, London (UK): Academic Press, 1968.

Cope, D., *Computers and Musical Style*, Oxford (UK): Oxford University Press, 1991.

Dewdney, A. K., "Building computers in one dimension sheds light on irreducibly complicated phenomena", *Scientific American*, May 1985.

Dewdney, A. K., "A cellular universe of debris, droplets, defects and demons", *Scientific American*, August 1989.

Dodge, C., Jerse, T. A., *Computer Music: Synthesis, Composition and Performance*, New York (NY): Schirmer Books, 1985.

Hertz, J., Krogh, A., and Palmer, R. G., *Introduction to the theory of neural computation*. Redwood City (CA): Addison-Wesley Publishing Company, 1991.

Hunt, A., Kirk, R. and Orton, R., "Graphical Control of Granular Synthesis using Cellular Automata", *Proceedings of the International Computer Music Conference*, pp., 4160-418, 1991.

McAlpine, K., Miranda, E. R., Hoggar, S., "Composing Music with Algorithms: A Case Study System", *Computer Music Journal*, 23(2), 1999.

Miranda, E. R., "Cellular Automata Music: An Interdisciplinary Project", *Interface* (now called *Journal of New Music Research*), 22(1), 1993.

Miranda, E. R., "Music composition using cellular automata", *Languages of Design*, 2, 1994.

Miranda, E. R., "Granular Synthesis of Sounds by Means of Cellular Automata", *Leonardo*, 28(4), 1995.

Miranda, E. R., *Computer Sound Synthesis for the Electronic Musician*. Oxford (UK): Focal Press, 1988.

Miranda, E. R., "Sobre as Origens e a Evolução da Música" *Revista Eletrônica de Musicologia*, 5(2), <http://www.cce.ufpr.br/~rem/remi.html>, 2000.

Miranda, E. R., *Composing Music with Computers*. Oxford (UK): Focal Press, 2001.

Scott, S., "Clocks and chaos in chemistry", *New Scientist*, 02 December 1989.

Worral, D., "Studies in metamusical methods for sound image and composition", *Organised Sound*, 1(3), 1996.

Xenakis, I., *Formalized Music*, Bloomington (IN): Indiana University Press, 1971.