Cellular Automata in MIDI based Computer Music

Dave Burraston[*], Ernest Edmonds[*], Dan Livingstone[†], and Eduardo Reck Miranda[†]

[*] Creativity and Cognition Studios, Faculty of Information Technology, University of Technology, Sydney,
Australia
dave@noyzelab.com, ernest@ernestedmonds.com
[†]Computer Music Research, School of Computing, Communications and Electronics, University of
Plymouth, Plymouth, UK
{d.livingstone, eduardo.miranda}@plymouth.ac.uk

Abstract

This paper will present a survey of electronic music and sonic art applications of Cellular Automata in the MIDI domain, and of the implications of that work for future developments. Algorithmic and computational processes have been of interest to artists in the audio domain for many years, creating an emerging culture of algorithmic generative electronic art. There have been several approaches at applying Cellular Automata in the production of electronic music and sonic art. Creative domains of application exist in the fields of overall structural composition and MIDI sequencing. Cellular Automata utilised by these applications include 1D elementary, 2D Life and reaction diffusion systems. Applications from academic, independent and commercial sectors will be surveyed in an artistic, historical and technical context. This will provide the artist and scientist with a balanced view of this emerging field in generative electronic music and sonic art.

1 Introduction

Algorithmic and computational processes are an important tool for the technology based creative artist producing generative art systems. (Dorin 2001) (Candy and Edmonds 2002) (Edmonds 2003) (McCormack 2003) (Miranda 2003) Formal processes and algorithms have been utilised for centuries within the creative activity of music, the tools and technique of their application known as algorithmic composition. (Roads 1996) CA have been of interest to artists in the audio domain for many years, assisting an emerging culture of artificial aesthetics and algorithmic generative electronic art. Creating patterns and sequences is necessary for the creative artist working spatially and temporally within a chosen medium. CA are capable of a wide variety of emergent behaviours and represent an important generative tool for the artist.

CA were conceived by Stanislaw Ulam and John von Neumann in an effort to study the process of reproduction and growths of form (Burks 1970). This work was decades ahead of its time and the full impact of this work is just beginning to emerge, influence and impact on the cutting edge of the technology industry. CA are dynamic systems in which time and space are discrete. They may have a number of dimensions, single linear arrays or two dimensional arrays of cells being the most common forms. The CA algorithm is a parallel process operating on this array of cells. Each cell can have one of a number of possible states, sometimes expressed as **k**. The simultaneous change of state of each cell is specified by a local transition rule. The local transition rule is applied to a specified neighbourhood around each cell, sometimes expressed as **r**. CA are usually, but not always, infinite in length. Cells are commonly wrapped around at the edge of the array during the neighbourhood rule computation, to achieve a conceptual infinite array. In this case the array is finite but unbounded. The wiring for the edge cells and cell 4 of an 8 cell 1D CA is shown in Figure 1a. Here we can see two time steps of the system from **t0** to **t1**. The transition rule is specified by an 8 bit binary number between 0 and 255 and an example for rule 110 is shown in Figure 1b. The 8 entries in a rule transition table are defined as **T7** to **T0** left to right.

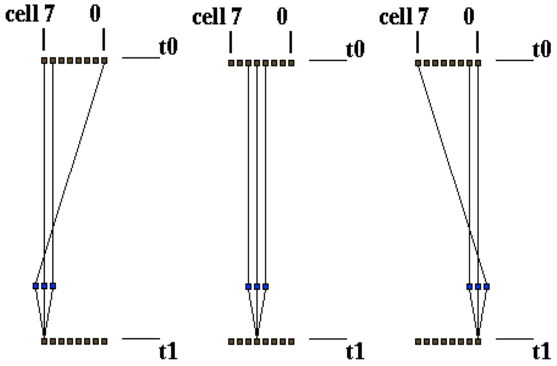

Figure 1a: Wiring of periodic boundary cells (left and right) and cell 4 (centre) of an 8 cell 1D CA.

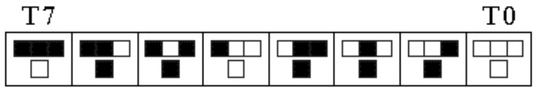

Figure 1b: Transition rule table for rule 110.

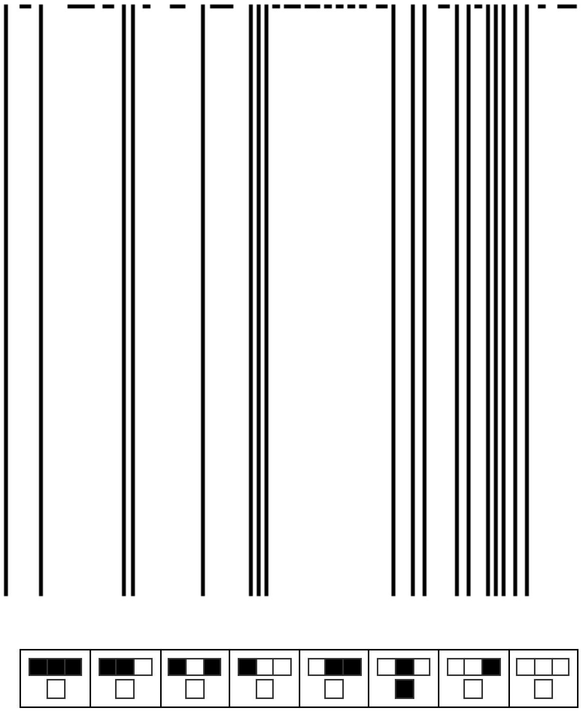

Figure 2: Class 1 Rule 4 – fixed behaviour.

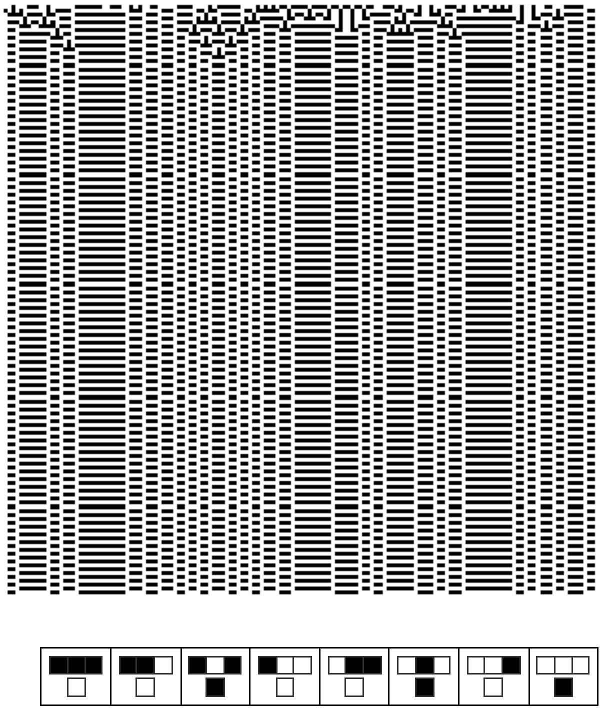

Figure 3: Class 2 Rule 37 - cyclic behaviour.

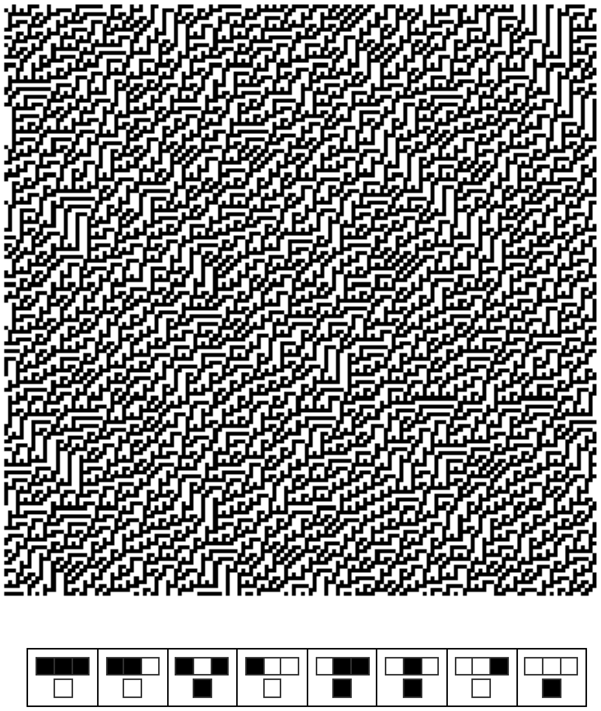

Figure 4: Class 3 Rule 45 – random/chaotic behaviour.

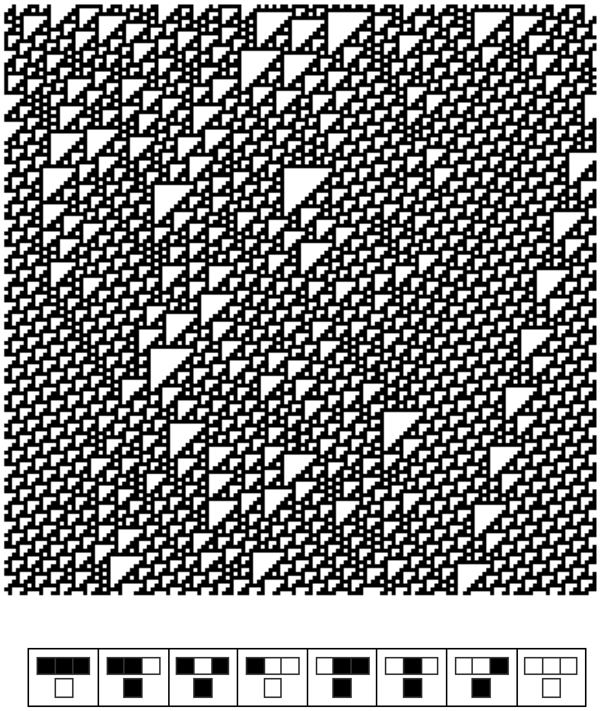

Figure 5: Class 4 Rule 110 – complex behaviour.

From a simple computational algorithm, such as one dimensional CA, it is possible to generate behaviour of intricate complexity. This parallels the art of both nature and evolution, intimating the generation of form and structure of incredible beauty. A prime example of their use is modelling pigmentation patterns in the shells of molluscs (Meinhardt 2003). Stephen Wolfram's work states that the 256 "elementary" one-dimensional binary CA (k=2, r=1) can be classed by one of four behaviours (Wolfram 2002).

Class 1: Patterns disappear with time or become fixed, they evolve to a homogenous state.

Class 2: Patterns evolve to a fixed size forming structures that repeat indefinitely, periodic structures cycling through a fixed number of states.

Class 3: Patterns become chaotic and never repeat, forming aperiodic and random states.

Class 4: Patterns grow into complex forms, exhibiting localized structures moving both spatially and temporally.

Figures 2 to 5 shows typical evolutions for each 1D class (Adamatzky 2001) (Wolfram 2002) (Griffeath and Moore 2003). The top row is the seed and the time evolves vertically downwards. Below each evolution is the neighbourhood rule table, showing the 8 possible configurations and rule for the next timestep.

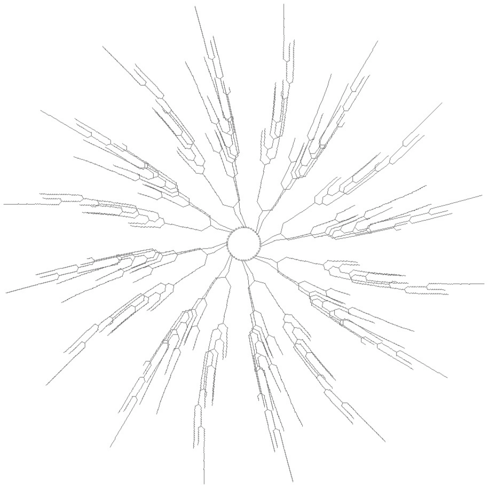

Figure 6: Class 3 Rule 30 attractor basin.

Further contributions to this classification were introduced by Chris Langton with the introduction of the Lambda parameter (Langton 1986). This is a kind of virtual tuning knob through Wolfram's classes of the CA rule space within a scalar range. Although the Lambda parameter appears useful, it should be used with care. Langton points out that it does have weaknesses and will not always be able to work correctly. Langton also supported and promoted work on the global dynamics of CA (Wuensche and Lesser 1992) which offers a new perspective based on the topology of attractor basins. In this work an atlas of these basins is presented for a variety of small CA sizes. Here one can compare the basin topologies between rules and useful insight into rule behaviours can be obtained. An example attractor basin for a 16 cell chaotic rule 30 is shown in Figure 6.

CA have been of interest to artists in the audio domain for many years, assisting an emerging culture of artificial aesthetics and algorithmic generative electronic art. Creating patterns and sequences is necessary for the creative artist working spatially and temporally within a chosen medium. CA are capable of a wide variety of emergent behaviours and represent an important generative tool for the artist. An interesting area is the crossover of CA into hardware, which will have a bearing on future creative media technology. These developments include both a 1D hardware CA (Sipper 1997) and the Bio Wall, an interactive 2D self-replicating CA display (Stauffer and Sipper 2002). CA have a substantial and wide ranging body of research (Toffoli and Margolus 1985) (Adamatzky 1994) (Sipper 1998) (Meinhardt 2003) which will continue to influence many artists in the future.

2 CA MIDI Systems

CA have been utilised in a number of novel applications in the MIDI domain. Predominantly these applications have been in the area of MIDI sequencing, using a multiplicity of CA types and mapping methodologies. Two early MIDI experimenters of CA were Peter Beyls and Dale Millen, working independently in academia on slightly different systems.

Beyls (1980) (1989) (1990), one of the first examples of a CA music system, is interesting both technically and also from a music industry standpoint as it was later commercially sponsored by Atari and Yamaha. The types of CA investigated were varied and novel, drawing from a mixture of 1 and 2 Dimensions. A further interesting avenue was the use of time dependent rules, where the rule itself changes during the CA evolution. 2D rules were also applied to a 1D CA, the North and South neighbours extracted from previous and future generations respectively. Further experiments included feedback from cell neighbourhood evaluation into the rule set itself and history tracking to include selected previous generations into the computation. A 2D wave propagation CA, based on the interaction of moving particles, was also investigated using a mouse to select areas of the wave field. Beyls wanted a flexible MIDI mapping process for real-time composition and performance, after his earlier work in non real-time. This mapping drew from the CA history evaluation, a user defined root and the current cell value. Selection of MIDI channel was by cell index number, using the modulus function to map to the available number of MIDI channels. Durational values were computed by matching the CA cell value with a condition in a large decision tree.

Beyls (1991) further expanded this work by investigating a small network of interconnected 2D CA and the use of Langton's Lambda parameter within a real-time system. Three 2D CA of 8x8 cells were used with the first CA accepting physical input gestures. The second CA is influenced by its own transition rules and the output of the first CA. The transition rules of the second CA may be tuned with the Lambda parameter by the user and also by a "subtle feedback mechanism". The third CA is used to generate MIDI messages on up to 16 MIDI channels by a process similar to activation/inhibition and is fed from the output of the second CA. An active cell in the second CA causes a non-linear increase in its corresponding cell in the third CA, also with the effect of fading some of its neighbours. Beyls (1997) (1998) (2000) continues with his work in the field of multiple CA and has extended this to include selections based on Genetic Algorithms and further use of the Lambda parameter with his CA Explorer program. (Beyls 2003) The CA are viewed as genotypes and are subject to mutation and cross-over operations.

Cellular Automata Music (CAM) was created by Dale Millen at the University of Arkansas. Music is created from 1D (k=2, r=2), 2D Game of Life and 3D Game of Life CA by mapping the results to pitch and duration values. (Millen 1990) The 1D and 2D lattice sizes are scalable up to 100. The 1D (k=2, r=2) rule is completely definable as a 32 bit pattern by on/off switches, thus allowing any of the 2^32 possible rules to be selected. Pitches are entered as a set of values and durations are applied by the program automatically, except in the 3D case where the calculated duration is also scalable. The seeding of the CA can be achieved by a number of methods, in the case of 2D Game of Life this can be entered graphically with the mouse or by automatic seeding of Game of Life forms. Millen (1992) later investigated the generation of formal musical structures with CAM, using a cyclic 1D (k=2, r=2) CA rule. This involved the arbitrary mapping of musically related pitch values to the CA rule.

CAM runs on Macintosh and is available on the Internet at http://comp.uark.edu/~dmillen/cam.html (Last visited 9/11/2003).

The Music Technology Group at the University of York developed a one dimensional "Cellular Automata Workstation", which attempted to allow the composer to interact with CA process in real time. (Hunt, Kirk and Orton 1991) The composer could adjust parameters such as rule number, neighbourhood size and number of cells. The composer could also zoom in on an evolution to map particular areas of interest to musical parameters. The basic mapping could consist of an active cell controlling pitch, the composer being able to move through CA generations to test mappings or perform live. Cells were allowed to be muted by using a pitch mask feature, allowing the simple extraction of a subset from the CA evolution. An important outcome of this work was the identification that CA output could be used to construct data streams for the parametric control of electroacoustic and MIDI instruments. These data streams were under the control of the composer, constructed by the arbitrary partitioning of the CA into blocks. Data stream mappings could take the form of MIDI controllers or system exclusive data. This work went on to describe high level control of a composition, using the output of the CA to control the playback of discrete musical passages.

In CAMUS and CAMUS 3D Eduardo Miranda investigated whether CA that exhibit pattern propagation behaviour could be utilised to model musical pattern propagation. (Miranda 1993) (McAlpine, Miranda and Hoggar 1999) (Miranda 2001) The chosen CA were the Game of Life and Demon Cyclic Space (Dewdney 1989), both occupying fixed grid sizes, CAMUS using 2D CA and CAMUS 3D extending the concept to 3D CA. In CAMUS, the Game of Life is used to determine the intervals of a triad based on the x/y locations of active cells on a column by column basis .

The Game of Life is a two-dimensional CA that attempts to model a colony of simple virtual organisms. In theory, the automaton is defined on an infinite square lattice. For practical purposes, however, it is normally defined as consisting of a finite $\mathbf{m} \times \mathbf{n}$ array of cells, each of which can be in one of two possible states: alive represented by the number one, or dead represented by the number zero. The state of the cells as time progresses is determined by the state of the eight nearest neighbouring cells. There are essentially four rules that determine the fate of the cells at the next tick of the clock:

a) Birth: A cell that is dead at time **t** becomes alive at time **t** + 1 if exactly three of its neighbours are alive at time **t**.

b) Death by overcrowding: A cell that is alive at time **t** will die at time **t** + 1 if four or more of its neighbours are alive at time **t**.

c) Death by exposure: A cell that is alive at time **t** will die at time **t** + 1 if it has one or none live neighbours at time **t**.

d) Survival: A cell that is alive at time **t** will remain alive at time **t** + 1 only if it has either two or three live neighbours at time **t**.

Whilst the environment, represented as **E**, is defined as the number of living neighbours that surround a particular live cell, a fertility coefficient, represented as **F**, is defined as the number of living neighbours that surround a particular dead cell. Note that both the environment and fertility vary from cell to cell and indeed from time to time as the automaton evolves. In this case, the life of a currently living cell is preserved whenever $2 \leq \mathbf{E} \leq 3$ and a currently dead cell will be reborn whenever $3 \leq \mathbf{F} \leq 3$. Clearly, a number of alternative rules can be set. The general form for such rules is ($\mathbf{E}_{\mathbf{min}}$, $\mathbf{E}_{\mathbf{max}}$, $\mathbf{F}_{\mathbf{min}}$ and $\mathbf{F}_{\mathbf{max}}$) where $\mathbf{E}_{\mathbf{min}} \leq \mathbf{E} \leq \mathbf{E}_{\mathbf{max}}$ and $\mathbf{F}_{\mathbf{min}} \leq \mathbf{F} \leq \mathbf{F}_{\mathbf{max}}$. The CAMUS implementation of the Game of

Life algorithm enables the user to design rules beyond Conway's original rule. However rules other than (2, 3, 3, 3) may exist, but not all of them produce interesting emergent behaviour.

CAMUS uses a Cartesian model in order to represent a triple of notes. In this context, a triple is an ordered set of three notes that may or may not sound simultaneously. These three notes are defined in terms of the distances between them, or intervals in music jargon. The horizontal co-ordinate of the model represents the first interval of the triple and the vertical co-ordinate represents its second interval as shown in Figure 7.

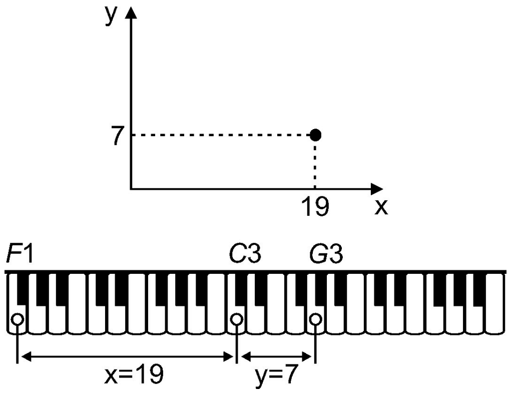

Figure 7: The CAMUS system attempts to go beyond simplistic mapping of CA onto musical notes by adopting a two-dimension spatial representation whereby the co-ordinates of a cell in the space correspond to the distances between the notes of an ordered set of three musical notes.

To begin the musical generation process, the CA is set up with an initial random configuration and set to run. When the Game of Life automaton arrives at a live cell, its co-ordinates are taken to estimate the triple from a given lowest reference note. For example, the cell at the position (5, 5) is alive and will thus generate a triple of notes. The co-ordinates (5, 5) describe the intervals of the triple: a fundamental pitch is given, then the next note will be at five semitones above the fundamental and the last note ten semitones above the fundamental. Although the cell updates occur at each time step in parallel, CAMUS plays the live cells column by column, from top to bottom. Each of these musical cells has its own timing, but the notes within a cell can be of different lengths and can be triggered at different times. Once the triple of notes for each cell has been determined, the states of the neighbouring cells in the Game of Life are used to calculate a timing template, according to a set of temporal codes as shown in Figure 8.

CAMUS 3D uses the additional **z** coordinate to create a four note grouping. Whereas CAMUS applies temporal coding to these intervals based on the neighbouring cells. CAMUS 3D uses a first order Markov Chain. In both programs the corresponding cell of Demon Cyclic Space determines the orchestration of the output to a MIDI channel corresponding to its current state. Both programs allow for a good deal of interaction and manipulation of musical parameters, and some adjustment of the CA rule space. CAMUS and CAMUS 3D are both available on CDROM in Miranda (2001).

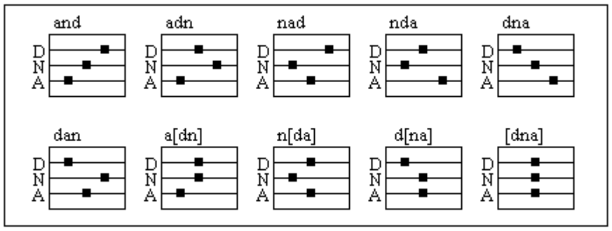

Figure 8: The codification scheme for abstract timing shapes.

Andrew Martin investigated the application of Reaction-Diffusion (R-D) systems, (Meinhardt 2003) (Turing 1952) (Turk 1991) to produce MIDI based compositions at the Australian Centre for the Arts and Technology (ACAT). (Martin 1994) (Martin 1996) The R-D application is an interesting approach, differing from the pure computational logic approach. Martin used a 24 □ 24 grid of R-D cells, scanning rows from left to right and top to bottom, giving the effect of “self-modifying repetition”. Each cell produced a number of values, the concentration of two morphogens and their second derivatives. These cell values were assigned to control tempo, note durations, note onset, note velocity and the note value itself. The note value was selected from a discrete set within a specified limit. A cell’s morphogen value could also be assigned to an event mask, based on the dominance of one morphogen over another, in order to further impart the algorithm on the note stream. Multiple instrument mappings were possible by assigning them to a subset of cell parameters. Martin implemented the system in Forth, but later produced a Max based version specifically as a drum machine.

FractMus 2000 is an algorithmic composition system for Windows with the ability to perform 1D binary CA based pitch sequencing. The system will allow for up to sixteen CA mapped to individual MIDI channels. Each MIDI channel is based on an event structure. This allows for CA parameters to change over time and for much experimentation with rule, size and initial conditions and its effect on pitch. The number of cells is restricted to values between 128 and 512. Experiments by the authors have produced interesting pitch sequences, but the program does not allow investigation into any other types of CA mapping. FractMus 2000 is available on the Internet at http://www.geocities.com/SiliconValley/Haven/4386/ (Last visited 9/11/2003).

Harmony Seeker, an interactive application developed at the University of Calabria, Italy, represents an interesting hybrid approach and uses Genetic Algorithms to breed/select multiple types of 1D CA, with k=2 or higher, and renders successful fitness matches as a batch of MIDI files. (Bilotta, Pantano and Talarico 2000) (Bilotta and Pantano 2001) (Bilotta and Pantano 2002) A mapping is achieved through the use of "musification codes", of which three types have been identified, Local, Global and Mixed. Local codes view the CA as a piano roll and the presence of an active cell causes a note event to occur. Global codes which view the CA as a whole and extract musical passages based on measures taken from the input-entropy and the evolution over time. Mixed codes extract sections from the CA and maps these to note and tempo parameters. The software is in an early beta stage and can generate a large number of MIDI sequences, although there is no capacity to map CA to loudness dynamics. These codes are based on a fitness test for musical consonance. While this remains an interesting approach to utilising CA for the production of contemporary music, it also limits itself by retaining the boundaries set by musical consonance. The approach also retains many of the barriers to contemporary practice by assuming this "musical consonance" for its breeding and selection of CA rules, and does not encourage newer mapping methodologies or alternative sound domains, especially those using non-pitched sounds.

Softstep is a commercial modular algorithmic development application for the PC by Algorithmic Arts, designed for the construction of MIDI sequencers. (Dunn 2002) CA modules included among the many other esoteric modules are Life, HiLife, and 1D Wolfram binary CA. These CA are implemented as options within Softstep's Matrix modules. Tests can be made for static or frozen evolutions to be automatically reseeded randomly or by a user specified seed, and other re-seeding options based on the current state. The application of the CA output to the MIDI domain is in the hands of the programmer.

3 Discussion of CA Systems Reviewed and Conclusion

Presented in Table 1 is a general comparison of the CA music systems reviewed, based on the following features: dimension, amount of cells, number of states, types of rules, number of CA, seeding type. CA music systems are notated in the left column and features are identified in the top row. We are now able to compare differences and similarities between work in this field. The first four features relate to the architectures of the CA system used. We can easily see differences in the architectures based on numbers of dimensions, cells and states, and of the rule types implemented. Following this are two further columns identifying the number of CA within each system, and their seeding mechanisms.

The rule types used show a reasonable degree of diversity. The one dimensional CA is a popular choice for both domains and, perhaps not surprisingly due to its wider popularity, 2D/3D Game of Life has also influenced researchers. A wide range of choice in the number of cells used is apparent, SoftStep modules being quite small, at up to 32 cells, with FractMus 2000 allowing up to 512 cells on a sufficiently powerful machine. The number of CA chosen is quite well balanced between single and multiple, and in the case of Andrew Martin's work have been created as networked compositional structures. SoftStep theoretically allows this as the CA are specified as modules within a larger programming environment. Seeding mechanisms for CA are quite generic, and random and user specified seeds are the norm for logic based systems. In 2D/3D Game of Life many starting combinations have been documented and these are often referred to as Lifeforms. In Reaction-Diffusion systems the starting conditions are usually at an equilibrium value.

	Dim	Cells	States	Rules	CA	Seed
Beyls CA Explorer 2.0	1	12	2 - 8	3,5 or 7 neighbour	9	Random, user
CAM (Millen)	1,2 and 3	1 and 2D up to 100	2	K2r2, Life, 3D Life	1	Random, Life forms, user
CAMUS / CAMUS 3D	2 and 3	40x40 12x12x12	2 (Life) 2 - 16 (DCS)	Life,Demon Cyclic Space, 3D Life	2	Random, user
CAW (York)	1	User	2	K2r1, user	1	Random, user
FractMus 2000	1	128 to 512	2	K2r1	Up to 16	User
Harmony Seeker	1	50 (practical limit)	4 (practical limit)	User	Multiple as batch	Random
Martin	1 and 2	24x24	NA	R-D	Multiple network	Equilibrium
SoftStep	1 and 2	Up to 32	2	Life, HiLife and k2r1	Multiple free assigned	Random, user

Table 1: CA music research comparison.

The diversity of CA and mappings used in these applications, though interesting, makes direct comparison somewhat difficult. From Table 2 we can see that note and duration parameters have been the most widely investigated in the MIDI domain. Softstep offers the capability to explore all parameters, but these must be constructed by the user and, unlike the others does, not represent an implemented system. In the domain of note specification three systems choose from a specified pitch set, whereas the remainder are generating either single or multiple note values from the CA output. Important musical parameters, such as loudness in the form of note velocity, tempo and timing has a marked lack of implementation.

Although much work has been done with CA in music and sonic art, the diversity and growth of the field will require further research with MIDI and sound. The majority of work so far has been conducted in the domain of note sequencing alone and further investigations are still imperative. The domain of MIDI loudness dynamics, and other parameters, has received less attention to date. More attention needs to be paid both on the architectures used, the application domains and the interface presented to the user. The creation of scalable CA architectures targeted at creative media applications is of prime importance. The sonic artist and musician must be prepared to investigate the technical and theoretical background of CA in order to successfully employ this vast behaviour space within their compositional strategy.

	Note	Velocity	Duration	Timing / Tempo	Controllers	SysEx
Beyls CA Explorer 2.0	Note & Chord		X			
CAM (Millen)	User pitch set		X			
CAMUS / CAMUS 3D	Chord		X			
CAW (York)	User pitch set				X	X
FractMus 2000	X					
Harmony Seeker	Single & Multiple Note		X	X		
Martin	User pitch set	X	X	X		
SoftStep	X	X	X	X	X	X

Table 2: CA MIDI systems comparison.

By way of conclusion, it is clear that CA have been very useful in musical practices. The use of CA in music is perhaps the most successful example of the application of evolutionary computation techniques in the arts. One criticism that comes to mind, however, is that none of the works reviewed in this paper went beyond the pragmatic utilitarian use of CA for building a compositional system. As with the fields of Acoustics, Psychoacoustics and Artificial Intelligence, which have greatly contributed to our understanding of Music, the field of Evolutionary Computation has the potential to reveal new aspects of music theory that are just waiting to be unveiled. CA is a powerful evolutionary modeling technique with great potential for Musicology. For example, we believe that CA could aid the study of the circumstances and mechanisms whereby music might originate and evolve in artificially designed worlds inhabited by virtual communities of musicians and listeners. In this case, music could be studied in the context of the origins and evolution of cultural conventions that may emerge under a number of constraints. It is up to musicologists to embrace the challenge.

4 Acknowledgements

The authors would like to thank Andrew Wuensche for allowing us to present the basin of attraction image in Figure 5 and CA wiring images in Figure 1a made at CCS with DDLab available from (www.ddlab.com). Figures 1b, 2, 3, 4 and 5 were made at CCS with Mathematica and NKS Explorer.

References

Adamatzky, A. (1994) *Identification of Cellular Automata*. Taylor & Francis.

Adamatzky, A. (2001) *Computing in Nonlinear Media and Automata Collectives*. Institute of Physics Publishing.

Beyls, P. (1980) “Action.” Exhibition catalogue, Kindt Editions, Belgium.

Beyls, P. (1989) “The Musical Universe of Cellular Automata.” In T. Wells & D. Butler, Eds., *Proceedings of the 1989 International Computer Music Conference*, pp. 34-41. International Computer Music Association.

Beyls, P. (1990) “Musical Morphologies from Self-organising Systems.” *Interface, Journal of New Music Research,* 19(2-3), 205-218.

Beyls, P. (1991) “Self-Organising Control Structures using Multiple Cellular Automata.” In *Proceedings of the 1991 International Computer Music Conference,* pp. 254-257. Montreal, Canada: International Computer Music Association.

Beyls, P. (1997) “Aesthetic Navigation.” *Proceedings of the JIM Conference,* Lyon, France.

Beyls, P. (1998) “Interactive Cellular Automata.” *Evolution 2.0 CDROM,* Liverpool Art School and Merseyside On-Line Ltd.

Beyls, P. (2000) “Synthetic Creatures in Context.” *Intersens et Nouvelles Technologies,* MIM (Laboratoire Musique et Informatique de Marseille).

Beyls, P. (2003) “Selectionist musical automata : Integrating explicit instruction and evolutionary algorithms.” *IX Brazilian Symposium on Computer Music*. Brazilian Computing Society.

Bilotta, E., Pantano, P. and Talarico, V. (2000) “Music Generation through Cellular Automata: How to Give Life to Strange Creatures.” *Generative Art GA2000,* Milano, Italia.

Bilotta, E. and Pantano, P. (2001) “Artificial Life Music Tells of Complexity.” *Proc. of Artificial Life Models for Musical Applications (ECAL 2001 Workshop),* Prague, Czech Republic.

Bilotta, E. and Pantano, P. (2002) “Synthetic Harmonies: Recent results.” *Leonardo* Vol. 35, No. 2, pp. 35-42, MIT Press.

Burks, A. (Ed) (1970) *Essays on Cellular Automata*. Univ. of Illinois Press.

Candy, L. and Edmonds, E. (2002) *Explorations in Art and Technology*. Springer.

Dewdney, A. K. (1989) “A cellular universe of debris, droplets, defects, and demons.” *Scientific American,* August, pp. 88-91.

Dorin, A. (2001) “Generative Processes and the Electronic Arts.” *Organised Sound* 6(1):47-53, Cambridge University Press.

Dunn, J. (2002) *SoftStep V3.1 Manual.* http://ww.geneticmusic.com (Last visited 9/11/2003).

Edmonds, E. (2003) “Logics for constructing generative art systems.” *Digital Creativity,* Vol. 14, No. 1, pp. 23-28.

Griffeath, D. and Moore. C. (2003) *New Directions in Cellular Automata*. Oxford University Press.

Hunt, A., Kirk, R. and Orton, R. (1991) "Musical Applications of a Cellular Automata Workstation." In *Proceedings of the 1991 International Computer Music Conference*, pp. 165-168. Montreal, Canada: ICMA.

Langton, C. (1986) Studying artificial life with cellular automata. *Physica D* 22: 120-149.

Martin, A. (1994) "Two Dimensional Reaction-Diffusion System for MIDI Composition." *Synaesthetica '94 Proceedings,* Australian Centre for the Arts and Technology (ACAT), Australian National University.

Martin, A. (1996) *The Application of Reaction-Diffusion Systems to Computer Music*. Master of Arts (Electronic Art) Sub-thesis, Australian Centre for the Arts and Technology (ACAT), Australian National University.

McAlpine, K., Miranda, E. R. and Hoggar, S. (1999) "Making Music with Algorithms: A Case-Study System." *Computer Music Journal*, Vol. 23, No. 2.

McCormack, J. (2003) "Art and the mirror of nature." *Digital Creativity,* Vol. 14, No. 1, 3-22.

Meinhardt, H. (2003) *The Algorithmic Beauty of Sea Shells*. Springer.

Millen, D. (1990) "Cellular Automata Music." In S. Arnold & D. Hair, Eds., *Proceedings of the 1990 International Computer Music Conference,* pp. 314-316. San Francisco: ICMA.

Millen, D. (1992). "Generations of formal patterns for music composition by means of cellular automata." In A. Strange Ed., *Proceedings of the 1992 International Computer Music Conference,* pp. 398-399. San Francisco: ICMA.

Miranda, E. R. (2003). "On the evolution of music in a society of self-taught digital creatures." *Digital Creativity*, Vol. 14, No. 1, 29-42.

Miranda, E. R. (1993) "Cellular Automata Music : An Interdisciplinary Project." *Interface*, Vol 22, No. 1, pp. 3-21.

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

Roads, C. (1996) *The Computer Music Tutorial*. MIT Press.

Stauffer, A. and Sipper, M. (2002) An interactive self-replicator implemented in hardware. *Artificial Life* 8(2): 175-183.

Sipper, M. (1998) Fifty years of research on self-replication: An overview. *Artificial Life* 4:237-257.

Sipper, M. (1997) *Evolution of Parallel Cellular Machines : The Cellular Programming Approach*. Springer Verlag.

Toffoli, T. and Margolus, N. (1985) *Cellular Automata Machines: a new environment for modelling*. MIT Press.

Turing, A. (1952) "The Chemical Basis of Morphogenesis." *Philosophical Transactions of the Royal Society*.

Turk, G. (1991) "Generating textures on arbitrary surfaces using reaction-diffusion." *Computer Graphics,* Vol. 25. No. 4, 289-298.

Wolfram, S. (2002*) A New Kind of Science*. Wolfram Media.

Wuensche, A. and Lesser, M. (1992) *The Global Dynamics of Cellular Automata : An Atlas of Basin of Attraction Fields of One-Dimensional Cellular Automata*. Addison-Wesley.