Digital Creativity
2005, Vol. 16, No. 3, pp. 165–185

Cellular automata in generative electronic music and sonic art: a historical and technical review

Dave Burraston[1] and Ernest Edmonds

University of Technology, Sydney, Australia

dave@noyzelab.com

survey

Abstract

This paper will review electronic music and sonic art applications of Cellular automata (CA) in a historical and technical context. Algorithmic and computational processes have been of interest to artists for many years, creating an emerging culture of 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.

The sonic artist and musician must be prepared to investigate the theoretical background of CA in order to successfully employ their vast behaviour space within compositional strategy. There is an extensive amount of mathematical and scientific literature relating to CA, however much of this is esoteric or difficult to understand. Important and accessible CA concepts are presented concisely in a non mathematical context to give sufficient background for the review.

There have been several approaches at applying CA in the production of electronic music and sonic art. Examples exist in the fields of overall structural composition, MIDI sequencing and sound synthesis/modification techniques. Applications from academic, independent and commercial sectors will be critically reviewed in an artistic, historical and technical context. This will provide the artist and scientist with a balanced view of this emerging field.

Keywords: algorithmic composition, cellular automata, electronic music, generative music, sound synthesis

1 Introduction

This paper will review electronic music and sonic art applications of Cellular automata (CA) in a historical and technical context. Following the main review is a comparative table highlighting the similarities and differences of the different approaches (Table 2). 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 and Miranda 2003). Complex systems such as Cellular automata (CA) produce global behaviour based on the interactions of simple units (cells). Their wide variety of ordered, complex and chaotic behaviour represents an important generative tool for the artist. CA have been of interest to musicians for many years, assisting an emerging culture of generative electronic art. An example visualisation of ordered, complex and chaotic CA behaviour converted to music is shown in Figure 1. Example CA evolutions are shown to the right, the horizontal dimension represents the cell space and time is evolving horizontally downwards.

The enabling technologies available to the electronic musician have taken the form of interconnection and sound generation. Musical Instrument Digital Interface (MIDI) was the first manufacturer wide standard enabling the interconnection of electronic music instruments to each other and to computers (Rumsey 1994 and Penfold 1995). This made possible to some degree the connecting of electronic musical devices for performance

1462-6268/05/1603-0165$16.00 Routledge Taylor & Francis Group

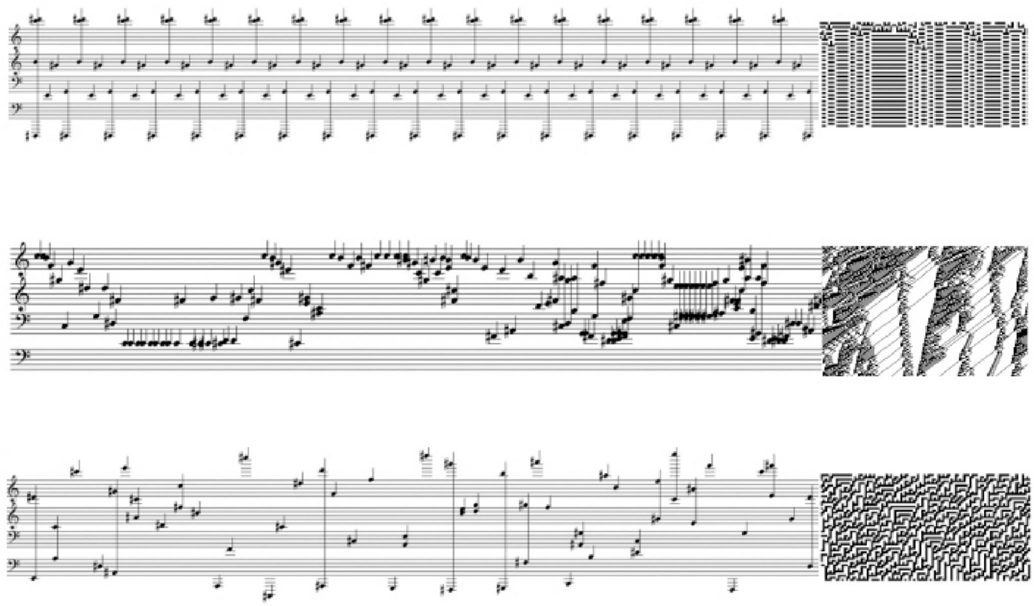

Figure 1. Musical visualisations of ordered, complex and chaotic CA behaviour.

and control, irrespective of manufacturer. Sound synthesis and processing techniques cover a vast area and is a constantly evolving field. A solid grounding of the principles and techniques in contemporary application is described in Roads (1996), Russ (1997) and Miranda (2002).

2 Generative art and algorithmics

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 1986). The evolution of the electronic computer transferred algorithmic composition to the digital domain, bringing a wealth of new possibilities and allowing the machine to be a part of the process. Notable among early experiments in algorithmic composition was the Olson-Belar composing machine (Olson and Belar 1961, Hiller 1970). Built around 1951, this machine allowed the assignment of weighted probabilities to electronically produced random numbers. The results of this process controlled pitch and rhythmic event generation. Lejaren Hiller and Leonard Isaacson began using information theory and statistical mechanics to automate musical composition in 1955, and by 1957 had created *Illiac Suite for String Quartet*, the first major algorithmic composition produced with a computer (Hiller and Isaacson 1959, Chadabe 1997). This provided a common ground for later collaborative work with John Cage on *HPSCHD*, completed in 1969 and consisting of up to seven keyboard parts, fifty-one tapes and over 5,000 slide projections. Iannis Xenakis initially composed with stochastic formulas by hand, notably *Metastasis* in 1955, using the laws of probability and large numbers. Xenakis later programmed a computer to aid the compositional process and in 1962 completed the *Stochastic Music Program* (Xenakis 1992).

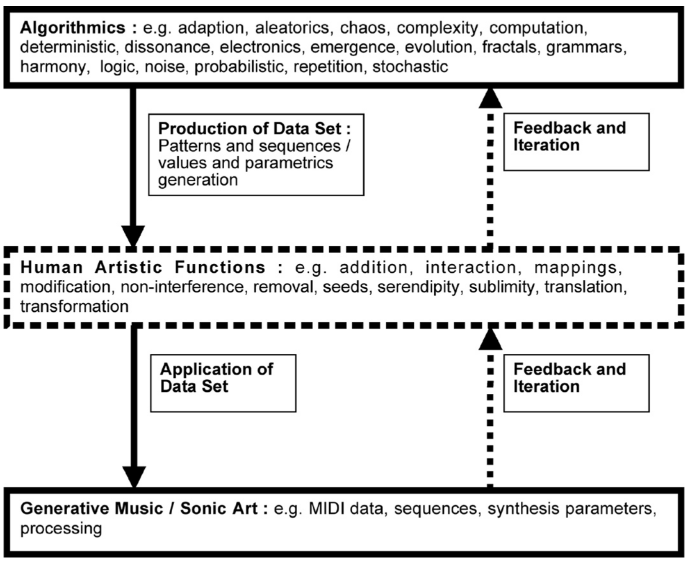

Figure 2. Algorithmic composition and the generative art process.

Symbiosis of interaction and performance with algorithmic techniques can be seen in the work of Raymond Scott and Salvatore Martirano. Scott founded Manhatten Research Inc. in 1946 and produced a host of esoteric machines to sequence and synthesize sound. His *Electronomium* was a composition and performance console utilising electronic relays. The system's musical output could be modified indirectly by the composer, guiding the system to produce a satisfactory composition. The *Electronomium* was used in advertising jingles, audio logos and film soundtracks, as well as by Scott for his own electronic compositions (Freff 1989). Martirano's *SalMar Construction* of 1972, was a realtime performance instrument utilising manually and logically driven touch sensitive switches. A total of 291 switches gave a performer interaction with four parallel programs controlling a modular analogue synthesizer. A zoom feature enabled movement and change within micro and macro structures. Martirano of his performances states, "Control was an illusion. But I was in the loop." (Chadabe 1997).

The generative art process suggested in McCormack (2003) represents genotype as the formal specification and the resultant artwork experience as phenotype. Algorithmic composition can be placed in context with the generative art process as detailed in Figure 2. Algorithmics allows the production of data sets to be mapped to parameters for the generation of musical or sonic outcome. In the centre, the artist may also perform functions upon this data set before application. Feedback and iteration may be factored into this process at all stages.

Complexity theory demonstrates that complex systems of simple units, such as the cells in a CA, produce a variety of behaviours. Complex systems produce global behaviour based on the interactions of these simple units. Excellent introductory accounts of the area of complexity theory, CA and Artificial Life are described in Levy (1992) and Coveney and Highfield (1995). In their account of complexity theory Coveney and Highfield define an emergent property as :

> *A global property of a complex system that consists of many interacting subunits.*

A more technical, but also readable, introduction to the ideas underlying complexity theory and adaption are described in Flake (1999). Adaption with genetic algorithms (GA), and complexity theory with CA, offer the use of emergent computation and behaviours as compositional aids to the generative process. Biological adaption is a natural creative process of evolution and GA attempt to model this behaviour, achieving this through the use of fitness criteria, and processes such as breeding and mutation. Approaches to algorithmic composition and generative art through the utilisation of such artificial models, draws much inspiration from natural and biological processes.

Digital Creativity, Vol. 16, No. 3

3 Background on cellular automata (CA)

Complex systems such as CA produce global behaviour based on the interactions of simple units. 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). The concept of the universe as a type of CA computer was introduced by Konrad Zuse and termed 'calculating space' (Zuse 1969). Here Zuse poses the controversial question : "Is nature digital, analog or hybrid?". Ed Fredkin, a long standing CA scientist, is convinced that the universe is digital (grainy) and has developed his own digital philosophy termed 'finite nature' (Fredkin 1992). Fredkin believes that the digital mechanics of the universe is much like a CA, deterministic in nature but computed with unknowable determinism. Space and time in this view are discrete quantities, everything is assumed to be grainy.

Some important CA concepts are now presented to give a sufficient background in the area for the purpose of this review. CA are dynamic systems in which space and time 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. 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. CA were qualitatively classed by Stephen Wolfram with one of four behaviours (Wolfram 1984).

Class 1 : Patterns disappear with time or become fixed.

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.

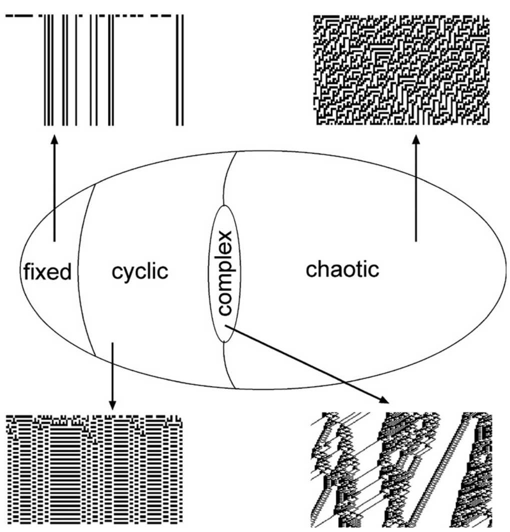

Figure 3. Langton's schematic of CA rule space and example space-time plots of their behaviour.

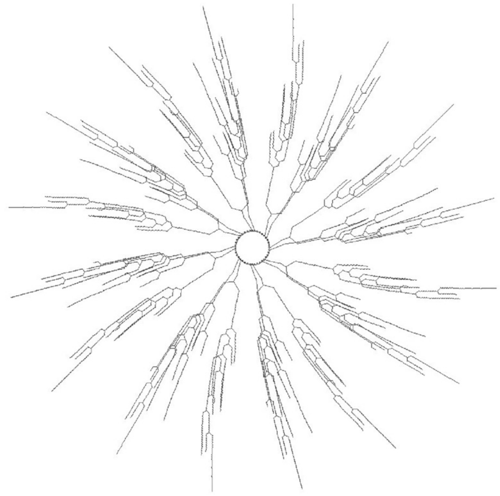

Figure 4. Example of an attractor basin for a chaotic rule.

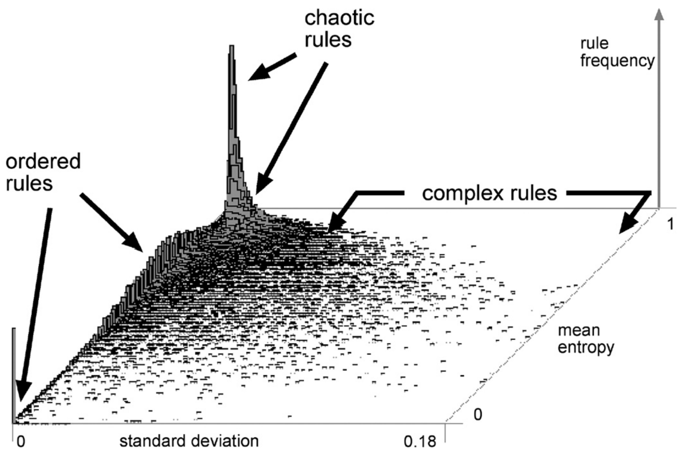

Figure 5. Example of Wuensche's automatic CA rule space classification using a random sampling of >10,000 5 neighbour 1D binary CA (v2k5).

Other methods of behaviour classification have been devised, an example of six categories is given in Li, Packard and Langton (1990). It is known that it is undecidable to assign a CA to a Wolfram class (McIntosh 1990) and there are several behaviour prediction parameters, five of which have been surveyed in (Oliveira, Oliveira and Omar 2001). The relatively rare complex behaviour of class 4 was shown to occur between ordered (class 1 and 2) and chaotic (class 3), termed the 'edge of chaos' (Langton 1990, 1991). Langton introduced the £f parameter, a kind of virtual tuning knob through the classes of CA rule space. Although the λ parameter appears quite useful it should be used with care. Langton points out that it does have weaknesses and will not always be able to work correctly. The λ parameter is seen as a useful tool when dealing with CA of higher number of states and neighbourhood size. Langton produced an example schematic illustrating his view of rule space shown in Figure 3. We have included spacetime plots of example 1D CA evolutions for each class. The evolutions show space (cells) in the horizontal and time runs vertically downwards. It is important to note on this diagram that the boundary between order and chaos contains complex behaviour within it. This implies that the transition from order to chaos as λ increases, may occur in a discontinuous jump to chaos or may pass through a region of complex behaviour.

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, rule symmetry categories and rule clustering. In their work an atlas of these basins is presented for a variety of small CA sizes up to about 15 cells depending on the particular rule. Here one can compare basin topologies and measures between rules to gain insight into different rule behaviours. Attractor basin topology reflects the dynamics of a CA rule and can be used as a method of identifying ordered, complex and chaotic

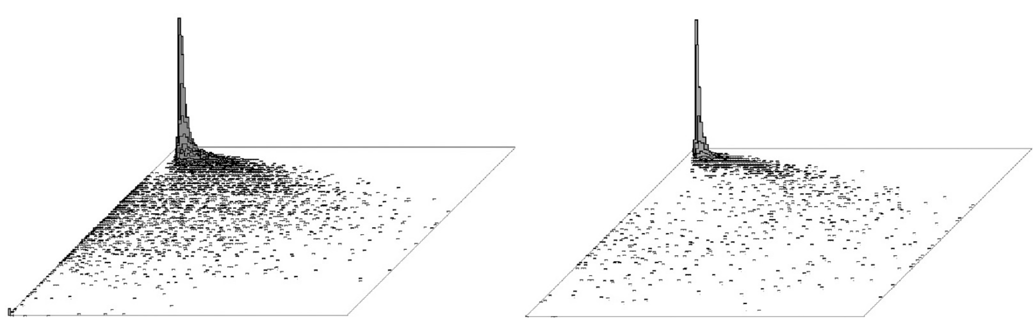

Figure 6. Example of increase in chaotic rules demonstrated by Wuensche's automatic CA rule space classification using a random sampling of >10,000 v2k6 (left) and v2k7 (right).

behaviour. Wuensche's *Discrete Dynamics Lab* (*DDLab*) software allows for the exploration of global dynamics (Wuensche 2005), as well as many other important aspects of CA. An example attractor basin for a chaotic rule is shown in Figure 4. In *DDLab* the number of state values is represented by v and the number of cells in the neighbourhood by k. This differs slightly from Wolfram's numbering scheme of k as the number of states and r as the neighbourhood radius. Wuensche's numbering scheme allows for even numbers of cells within a neighbourhood. Most of the CA music literature reviewed in the following sections adopt Wolfram's terminology. We will use Wuensche's numbering terminology from *DDLab* in the following brief outline of his work on automatic rule space classification and until the end of this section.

CA behaviour is often described based on a subjective assessment of its spacetime images. A thorough account of behaviour characteristics, its automatic classification by various parameters and example rules is given in (Wuensche 1997). Complex behaviour emerges (self-organises) in spacetime from random initial conditions to form gliders, particles or self-sustaining patterns existing over a uniform or periodic background. A glider can be seen as a dislocation or defect of this background. At least 150 cells, and more cells for larger neighbourhood sizes, are required before complex behaviour can emerge.

DDLab can automatically classify rule space from random rule samples based on mean entropy (a measure of amount of disorder) against its standard deviation (the variation of the entropy over time). Each type of behaviour has its own 'signature'. Ordered rules will tend to decrease entropy and standard deviation, chaotic rules increase entropy with a low standard deviation, and complex rules tend to have higher standard deviations and entropy values away from the extremes. An example of this method can be seen in Figure 5 where a random sample of over 10,000 rules was measured. The rules here are the 5 neighbourhood 1D binary rules, v2k5 in Wuensche's terminology or k2r2 in Wolfram's. Wuensche has importantly shown with this method that as the neighbourhood size is increased, the proportion of chaotic rules rises very sharply. This can be seen clearly in Figure 6 where similar plots are shown for 6 and 7 neighbour rules showing a chaotic tower as the main feature in the top left of the plots. The implication of this is that if a rule is generated at random it is highly likely to be a chaotic rule, which has implications for the serendipitous use of these systems in artistic endeavour.

The magnitude of the numbers of rules is extremely large, Wentian Li (1989) has commented on the 5 neighbour rules:

> *Even if we can produce a spatial-temporal pattern from each rule in 1 second, it is going to take about 138 years to run through all the rules. Considering the redundancy due to equivalence between rules upon 0-to-1 transformations, which cut the time by half, it still requires a solid 69 years.*

Rule type (vnkn)	Total number of rules
v2k3	2^(2^3) = 256 (the elementary rules)
v2k4	2^(2^4) = 65536
v2k5	2^(2^5) = 4294967296
v2k6	2^(2^6) = 1.844674407370955e+19
v2k7	2^(2^7) = 3.402823669209385e+38

Table 1. Total number of CA rules for v2k3 to v2k7 of one dimension.

The total number of CA rules is a function of the number of states and the size of the neighbourhood. A two state rule with three neighbourhood cells is termed a v2k3 rule and contains a total of 256 rules. Table 1 shows a summary of the total number of rules for 1D binary CA with neighbourhoods of 3 to 7 cells. As the neighbourhood is increased there is a dramatic increase in the total number of rules. The 1D v2k3 rules can be grouped into 88 equivalence

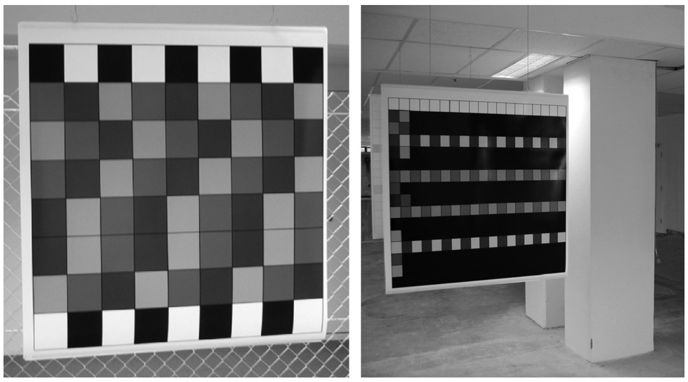

Figure 7. Noyzelab's digital prints of CA compositional data.

rule classes, with a maximum of four rules per equivalence class by the application of simple rule table transformations (Walker and Aadryan 1971). Global behaviour of rules within an equivalence class is identical and the spacetime patterns are simply changed by being negative, mirror image or both. The 88 equivalence classes can be further grouped into a total of 48 rule clusters by another simple rule table transformation (Wuensche and Lesser 1992). In this case the equivalence classes become grouped in clusters of related global dynamic properties, although their spacetime patterns are often different. Another popular method of reducing the number of rules for a chosen neighbourhood size is to simply take the sum of these cells (Wolfram 1983). Rules computed in this manner are termed 'totalistic' and are a very small subset of each vnkn rule type. There are only 64 totalistic v2k5 rules, reducing further to 20 rule clusters (Wuensche and Lesser 1992). A more manageable number, but also a large reduction of the possible behaviour. The v2k3 rules are also termed the 'elementary' rules and only one equivalence class (rule 110 and its 3 equivalents) is said to produce complex behaviour (Wolfram 2002).

Another interesting area is the research into hardware, which will have a bearing on future application of CA 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). In addition to the references already citied CA have a substantial and wide ranging body of research (Toffoli and Margolus 1985, Adamatzky 1994, Sipper 1998, Adamatzky 2001, Wolfram 2002, Griffeath and Moore 2003 and Meinhardt 2003) which will continue to influence many artists in the future. An extensive bibliography of CA can be found in Nisho (1975).

4 Generative electronic music and sonic art with CA

There have been several approaches at applying CA in the production of electronic music and sonic art. Iannis Xenakis used CA calculated with his 'pocket computer' to determine the succession of chords within a rational, perceptible structure for his composition *Horos* in 1986. Xenakis was attracted by the simplicity of the CA process to produce a rich output (Varga 1996). David Weinstein's graphical music score for his *Illuminated Man* (Davies 1986), a 20ft square textile dye on canvas, is a 2D arrangement of black and white squares which suggests a CA influence. Joel Ryan, in collaboration with the artist Ray Edgar, created *LINA*, an installation piece (Edgar and Ryan 1986, Beyls 1989). *LINA* utilised a single CA, with mappings applied to the MIDI and visual domains. Sound was produced by the use of a pair of Yamaha TX7 MIDI synthesizers and video by a Fairlight video synthesizer. Isle Ex exhibits on the internet philosophical transmusical renderings based on 2D CA. A variety of 2D rules have undergone transformation to MIDI by 'musification maps' and these may be viewed online (IsleEx 2005).

Bill Vorn has utilised 2D Life (Gardner 1970) for his noise art installation *EVIL/LIVE* (Vorn 2005). Vorn used an 8x8 matrix of halogen lights suspended from the ceiling controlled by both 2D Life and a matrix of sensors within the same room. The physical space becoming an influential input, by human location, creating a parallel interaction with the cellular space. Sounds samples are triggered by the cell activity to produce rhythmic patterns and structures. Vorn also created a library of objects for Cycling74's *Max* in 1996 called *Life Tools*, containing CA objects for 1D binary, 2D Life and 3D Life. These form part of the IRCAM library of *Max* objects and is available from IRCAM's ftp server (IRCAM 2005). Delicate Ear have created *Life Filter* using the *Life* object, where cell movement is mapped to control frequency of a 1024 band filter available. This is also the basis for the Cycling74 *Pluggo Harmonic Filter*. Another Delicate Ear CA application is *Cellsound*, in which up to 25 sine waves are controlled by CA. *Life Filter* and *Cellsound* are available for free at (Delicate Ear 2005).

Another installation work connecting CA and gallery space is *GalSim* (Woolf and Thompson 2002). *GalSim* uses a 2D lattice gas CA to generate the model of an artificial gallery space, designed to mimic the movement and reactions of people in a real gallery. The dimensions of the artificial gallery are adjustable to simulate different spaces. *GalSim* was used to provide a GA with automatically generated fitness data for *The Sound Gallery*, an interactive A-Life artwork. This enabled the testing of software before installation without having to recruit volunteers to interact at the design stage. Alan Dorin created an interactive installation piece called *Liquiprism* to produce polyrythmic patterns by the arrangement of six 2D CA grids (Dorin 2002). These CA were interconnected and presented as surfaces of a cube. CA activity may be altered by the user in order to influence the sonic outcome to some degree. Each surface of the cube is mapped to a different MIDI channel. Cells are assigned to pitches dependent on their position upon these surfaces. Dorin also makes the software

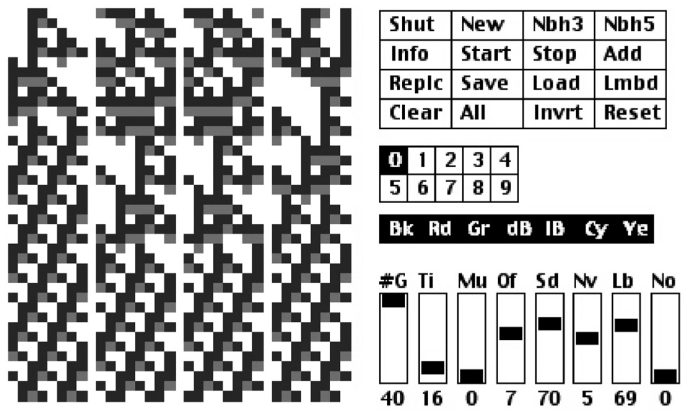

Figure 8. Peter Beyls' *Interactive cellular automata*.

available for free download (Dorin 2005).

Noyzelab exhibits a number of musical recordings based on 1D and 2D CA, and other algorithmic processes. These compositions were recorded using interactive and pre-processed algorithmic techniques. Here CA have been used in many forms , sometimes controlling other algorithms and juxtaposed with manual keyboard techniques. Some alternative methods of visualising CA compositional data have also been exhibited as digital prints in a gallery context, an example is shown in Fig 7. Experimental compositions and more data visualisations are presented in Burraston and Edmonds (2004), Burraston (2005a, 2005b, 2005c).

Cellular Grid Machine is a generative software artwork based on the notion of throw away music (Robot Software 2005). The interface allows a semi interactive composition process in which the software and the computer are also considered to be the medium. The system is a celebration of machine aesthetics and human-machine symbiosis. Grid based sequence patterns are created by a variety of mutating and interacting processes. A mutation system based on 2D Life or user defined rules may be induced to adjust the workings of the grid patterns.

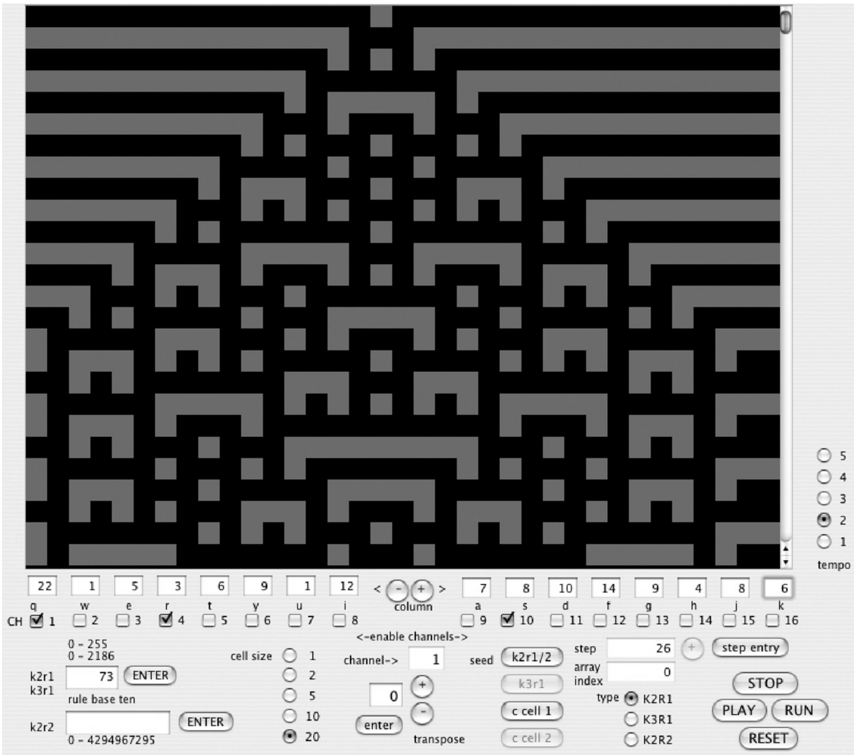

Figure 9. Dale Millen's *Cellular Automata Music.*

5 CA systems in the MIDI domain

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. A review of CA in the MIDI domain is presented in (Burraston, Edmonds, Livingstone and Miranda 2004). Two early MIDI experimenters of CA were Peter Beyls and Dale Millen, working independently in academia on slightly different systems.

Beyls' early work (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 (Beyls 89). 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 further expanded this work by investigating a small network of interconnected 2D CA and the use of Langton's Lambda parameter within a realtime system (Beyls 1991). 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

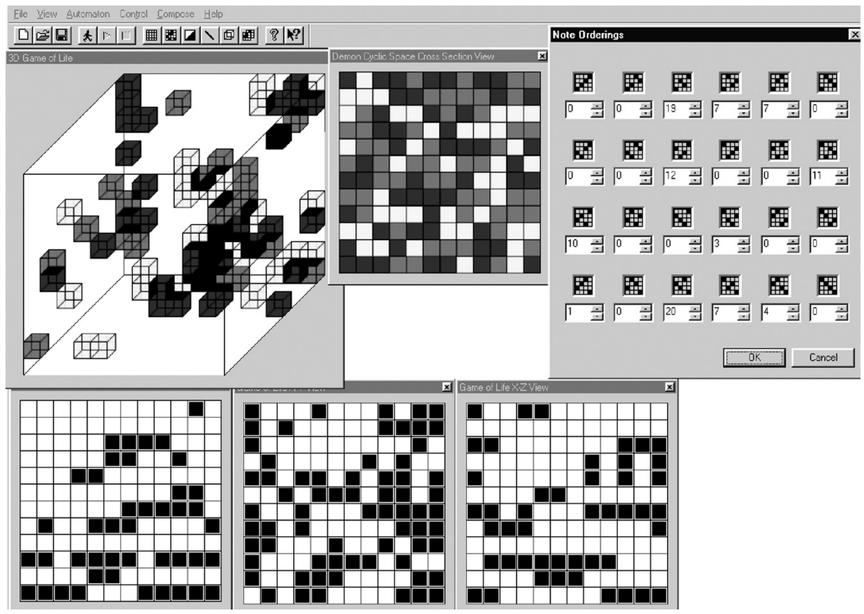

Figure 10. Eduardo Miranda's *CAMUS 3D.*

the effect of fading some of its neighbours. Beyls continues with his work in the field of multiple CA (Beyls 1997, 1998, 2000) and a screen shot of his *Interactive Cellular Automata* program is shown in Fig 8. Beyls has extended his work 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, the latest incarnation consists of two CA with a voting rule (Beyls 2004).

Cellular Automata Music (CAM) was created by Dale Millen at the University of Arkansas. Music is created from 1D k2r2, 2D Life and 3D 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 k2r2 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 Life this can be entered graphically with the mouse or by automatic seeding of Life forms. Millen later investigated the generation of formal musical structures with CAM, using a cyclic 1D k2r2 CA rule (Millen 1992). This involved the arbitrary mapping of musically related pitch values to the CA rule. CAM runs on Macintosh and a new version for Macintosh OSX (Millen 2004), shown in Fig 9. is available for free download (Millen 2005). The OSX version utilises three different types of 1D CA and the number of cells is preset in five sizes up to a maximum of 700 cells.

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 Orton1991). 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 2003). The chosen CA were 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, shown in Figure 10. In *CAMUS*, 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. *CAMUS 3D* uses the additional z coordinate to create a four note grouping. *CAMUS* applies temporal coding to these intervals based on the neighbouring cells and in *CAMUS 3D* by using 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 CD-ROM (Miranda 2001).

Andrew Martin investigated the application of Reaction-Diffusion (R-D) systems (Meinhardt 2003, Turing 1952 and Turk 1991), to produce MIDI based algorithmic compositions (Martin 1994, 1996a and 1996b) at the Australian Centre for the Arts and Technology (ACAT). The R-D application is an interesting approach, differing from the pure computational logic approach. Martin used a 24x24 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 cells morphogen value could also be assigned to an event mask, based on the dominance of one morphogen over an 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 (FractMus 2005).

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 et al. 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 map 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' (Bilotta and Pantano 2000). 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. A beta version of *Harmony Seeker* is available on CD-ROM (Miranda 2001).

Softstep is a commercial modular algorithmic development application for the PC by Algorithmic Arts, designed for the construction of MIDI sequencers (Dunn 2005). 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 reseeding based on the current state. The application of the CA output to the MIDI domain is in the hands of the programmer.

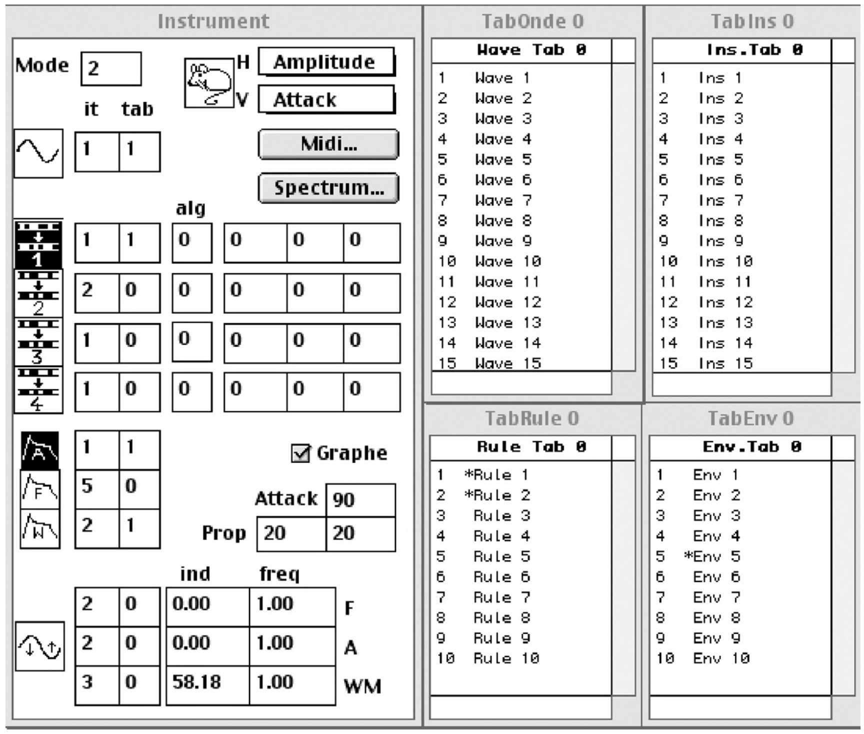

Figure 11. Jacques Chareyron's *Linear Automata Synthesis (LASy)*.

6 CA systems in the synthesis and processing domains

The application of CA in the audio domain has been investigated by a variety of approaches. This field has naturally evolved at a slower pace, mostly due to the higher computational cost of digital signal processing. Initial experiments combining CA and sound synthesis utilised a technique known as granular synthesis, based on Denis Gabor's quantum approach to sound (Gabor 1947), Xenakis being the first to develop a compositional theory for grains of sound (Xenakis 1992). Curtis Roads developed the first computer implementations of granular synthesis (Roads 1978, 1996) and Barry Truax further suggested that self-organising systems may provide useful models for this synthesis method (Truax 1988). Following this suggestion Peter Bowcott proposed a compositional control strategy utilising 2D Life. Bowcott's method extracted time stamped x/y coordinates of live cells to generate score data for Csound granular synthesis instruments (Bowcott 1989). Bowcott was also inspired by Xenakis' 'book of screens' (Xenakis 1992), whereas most previous granular synthesis methods used Roads' concept of tendency masks for the high level organisation of sonic grains. Bowcott importantly identified that the musical mapping of the algorithm should parallel its behaviour and not merely be "a representation of the visual output".

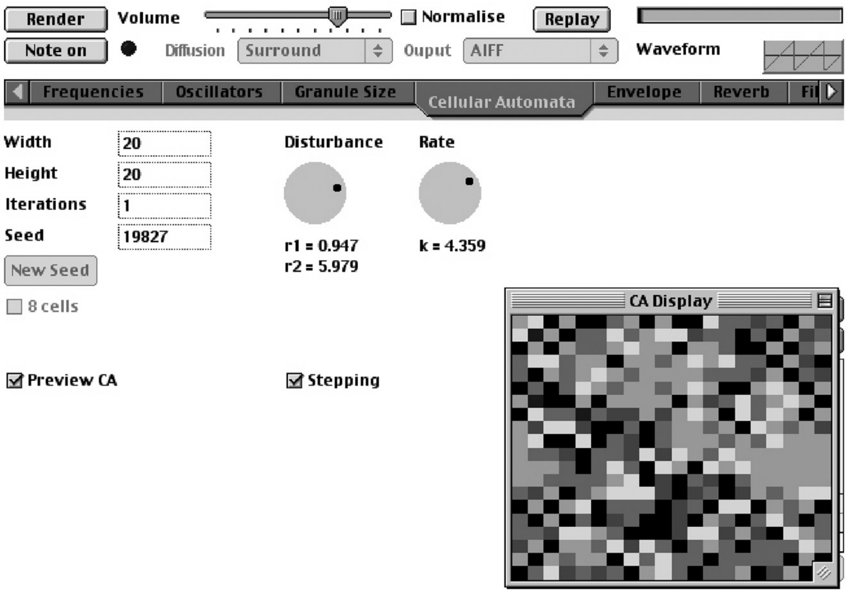

Figure 12. NYR Sound's *Chaosynth*.

The University of York's *Cellular Automata Workstation* (Hunt, Kirk and Orton 1991), described in the previous section, was used to map the CA output to the tendency masks of a granular synthesizer (Orton, Hunt and Kirk 1991) to assist in the process of generating a large amount of control data. York produced another graphical 1D CA tool (Katrami, Kirk and Myatt 1991) which allowed timbral manipulation of a phase vocoder (PV) (Flanagan and Golden 1966, Roads 1996). Timbral manipulation was achieved by using the CA output in two approaches, as frequency filters for the PV analysis data and also by applying the CA directly to the PV resynthesis process. The system used a fixed mapping of 512 pixels, the CA being scaled if not of this size.

LASy (*Linear Automata Synthesis*) was an inspired attempt by Jacques Chareyron at forging an original direction for CA synthesis, using a 1D CA to create a model of sound evolution (Chareyron 1988 and 1990). *LASy* creates a broad range of timbres from simple to complex and a screenshot is shown in Figure 11. The 1D CA is viewed as a wavetable and the cell values equate to sample values. Each generation of the CA thus becomes a time slice of the resultant sound stream, creating a self-modifying waveform based on the CA transition rules. Chareyron demonstrated that this process can be viewed as the operation of a digital filter on a time-limited signal. By using the sum of two neighbouring cells he went on to show that the basic Karplus-Strong plucked string algorithm (Karplus and Strong 1983) could be simulated. Interaction is via computer keyboard, mouse and MIDI in order to play and modify sounds. Sounds are organised as banks of instruments specifying the CA rule, an initial waveform and envelope settings. *LASy* runs on the Macintosh platform and is part of the University of Milan's computer music workbench called *Intelligent Music Workstation* (IMW) and is also available on CD-ROM(Miranda 2002).

Chaosynth produces sound by using a 2D CA to drive a granular synthesizer, originally implemented on a Cray supercomputer at Edinburgh Parallel Computing Centre (Miranda 2003). The CA is modelled as a matrix of cells, where each cell is an identical electronic circuit existing in a state of polarisation, depolarisation or collapse. The program allows some interaction on the part of the user for adjusting CA circuit parameters, shown in Figure 12. These being two resistors within a potential divider and a capacitor regulating the rate of depolarisation. Equal sections of the CA grid are chosen from a 'frequency pool' to control an oscillator, creating transitions from initial configurations into oscillatory patterns. *Chaosynth* also allows post processing of the CA output by multi-mode filters, ring modulators, envelope generators, low frequency oscillators, sample and holds, and a modulation matrix. This post CA section is very similar in architecture and use to a traditional analogue synthesizer. *Chaosynth* is available as a commercial

	Dimensions	Cells	States	Rules	CA	Seeding	MIDI	Audio
Beyls CA Explorer 2.0	1	12	2 - 8	3,5 or 7 neighbour	9	Random, user	X	
Ca (CCM)	1	32	2	K2r1	1	Random		X
CAM (Millen)	1,2 and 3	1 and 2D up to 100	2	K2r2, Life, 3D Life	1	Random, Life forms, user	X	
CAM OSX (Millen)	1	35 to 700 (in 5 fixed steps)	2 - 3	K2r1, k2r2, totalistic k3r1	1	Random, user, single seed	X	
CAMUS / CAMUS 3D	2 and 3	40x40 12x12x12	2 (Life) 2 - 16 (DCS)	Life,Demon Cyclic Space, 3D Life	2	Random, user	X	
CAW (York)	1	User specified	2	K2r1, user	1	Random, user	X	X
Chaosynth	2	Up to 999x999	>=3	Chemical Oscillator	1	Random		X
FractMus 2000	1	128 to 512	2	K2r1	Up to 16	User	X	
Harmony Seeker	1	50 (practical limit)	4 (practical limit)	User	Multiple as batch	Random	X	
LASy	1	512 per wave	4096	Function / Time based	Multiple	Waveform		X
Martin	1 and 2	24x24	NA	R-D	Multiple networked	Equilibrium	X	X
SoftStep	1 and 2	Up to 32	2	Life, HiLife and k2r1	Multiple free assigned	Random, user	X	
Virtual Waves	1	32	2	4 cell addition	1	Random, user		X

Table 2. CA music systems comparison.

application for Windows and an older free Mac OS9 version by NYR Sound (NYR 2005).

The MIDI R-D systems investigated by Andrew Martin at ACAT, described in the previous section, were also applied to the synthesis of sound (Martin 1995, 1996a and 1996b). These investigations were primarily with 1D R-D systems of both 24 and 32 cells. The output of the R-D systems created scores for an additive/FM CSound system, the number of FM instruments being equivalent to the number of R-D cells. The temporal nature of timbre was emphasized in this study, with instruments having R-D control mappings for peak amplitude, FM rate, FM depth, attack and decay characteristics. Martin further expanded this work to include multiple linked R-D systems of one and two dimensions with an interactive graphic interface and also utilising the Karplus-Strong algorithm (Karplus and Strong 1983). Parameters were extended to include panning position/speed and pluck index. The R-D systems were connected in a tree branching structure in order to control the overall parameters of a sonic composition. Later work at ACAT by Tim Kreger investigated the use of 1D binary CA to control a Fourier based spectral filter (Kreger 1997 and 1999). Here the CA was used to create an evolving band-pass filter, each cell determines if the respective band in the Fourier analysis will be resynthesised.

The Centre for Computer Music at the University of Cincinnati have produced a real-time granular processing tool called *ca* (Vaidhyanathan et al. 1999) for the SGI platform. A sound sample is selected and decomposed into grains, which are then passed through a bank of filters. A 32 cell binary 1D CA evolution controls the parameters of this bank of filters, transforming the harmonic structure of the grains.

Virtual Waves was an early commercial Windows application by Synoptic for building modular synthesis systems and rendering the result as a sound file. The CA module consists of a fixed 1D binary 32 cell CA additive synthesizer based on a user defined fundamental frequency. The frequency mapping is selectable as linear or logarithmic. The neighbourhood look up table uses 16 user definable sets of 4 cells, which is an unusual choice. The majority of published 1D CA research often has a symmetrical neighbourhood of 3 or 5 cells. The number of generations is restricted from 1 to 100, mapped to a total sound duration of between 100 and 10,000 ms. Further sound synthesis control includes a variable smoothing function, affecting the attack and decay of the spectral components.

7 Reflections

Presented in Table 2 is a general comparison of the CA music systems reviewed, based on the main features. 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 last two columns indicate the particular domains of application, MIDI or audio.

It is immediately apparent that the MIDI domain has proved more popular with researchers. 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 Life has also influenced researchers. A wide range of choice in the number of cells used is apparent, with *Chaosynth* allowing up to 999x999 cells on a sufficiently powerful machine. It is surprising that many of the logic based systems do not have enough cells for complex behaviour to emerge. With the benefit of the background knowledge of CA introduced in Section 3 it was shown that a reasonable minimum of around 150 cells are required. CAM OSX will produce complex behaviour, but cell size is restricted to five fixed settings. Fractmus 2000 is variable from 128 to 512 cells, but only k2r1 rule 110 (and its 3 other equivalents) will exhibit complex behaviour. A larger number of cells will consume more computer resources and it should be noted that many of these systems were created in an age when computer power was significantly lower than today's high powered desktops.

York's *Cellular Automata Workstation* and Andrew Martin's Reaction-Diffusion work are prime examples of a unified approach at investigation of both audio and MIDI domains. *LASy* is particularly notable in that the number of states is equal to the 12 bit synthesized audio stream, resulting in a 4096 state CA. Overall 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, where random, single cell and user specified seeds are the norm for logic based systems. In 2D/3D Life many starting combinations have been documented and these are often referred to as Lifeforms. Andrew Martin's Reaction-Diffusion systems are usually started at an equilibrium value. *LASy* represents a special case and the starting conditions are a wavetable, which may be any combination of values within the table.

The diversity of CA, mappings and domains used in these applications, though interesting, makes direct comparison somewhat difficult. 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 off the shelf CA 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 of loudness/note velocity, as well as tempo and timing, has a marked lack of implementation. Sound synthesis rather than processing formed the majority of investigation in the audio domain. Granular synthesis has been the most popular method of mapping the CA output. The implementation of parametric control is evident in CAW, *Chaosynth* and the work of Andrew Martin. Although many systems have been described not all are publicly accessible. CAM and CAMUS 2/3D programs are the most easily obtainable in the MIDI domain. For sound synthesis *Chaosynth* is the only one still being developed, but *LASy* can still be used under Mac OS9 and is certainly worth investigation.

Based on the reviews presented an overview of the CA design process is visualised in Figure 13. This presents a perspective of the criteria and flow useful for generative music. The process may include a specification for desired behaviour, but this is not always apparent or necessary. The architectural and rule set variables once chosen comprise the definition of the required CA. This part of the process may be repeated to obtain more than one CA definition. Following this step the seeding, evolution and mapping strategies can then be formed.

Music based academic resources exist for deeper investigation of CA for algorithmic composition in various forms. The Stanford University Center for Computer Research in Music and Acoustics (CCRMA) course notes on algorithmic composition include an interesting section on implementing CA algorithms in Common Lisp Music.

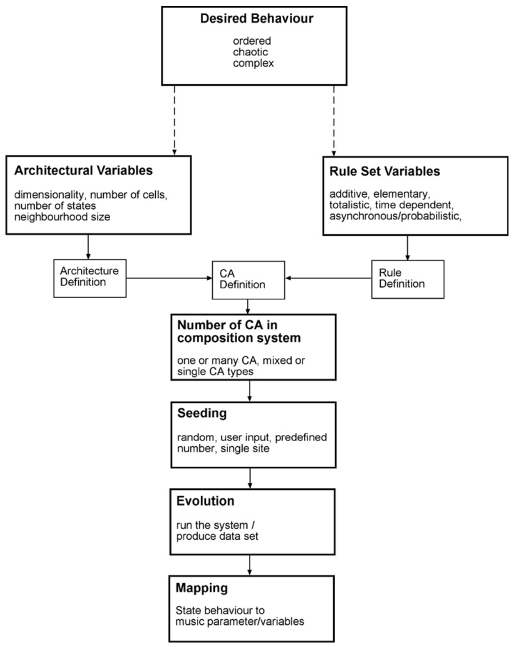

Figure 13. Process overview of generative music CA.

Binary CA of one and two dimensions are investigated and course notes are available (CCRMA 2005). Queensland University of Technology offer a 1D rhythmic CA, described in greater detail in (Brown 2005) and 2D Life as examples for the jMusic programming language. Under the leadership of Rudy Rucker the San Jose State University have created a 1D CA sound synthesis Java applet (Rucker 2005). The *Kyma Capybara* sound design environment from Symbolic Sound Corporation includes 1D CA objects as part of its standard library. Kyma offers these as harmonic and rhythmic interpretations of the CA evolution (Scaletti 1997). The popular and free *Super Collider* program also has a CA class library available at (SCOL 2005).

A number of open questions occur when reflecting on using CA for generative music. Is it currently possible for an engineer, artist or scientist to buy a CA chip or hardware module of any kind? Is there a market for such a device in any type of appliance, whether consumer, artistic or scientific? How much space would a device occupy in a studio? What are the differences between hardware and software implementation? How would the interface be presented? To date the only music industry interest in these systems was brief, during the early work of Peter Beyls. Music industry interest would surely require an evaluation of cost, technical requirements and development time.

8 Conclusions and future work

There is difficulty with comparatively assessing the diverse work published so far because it draws from a variety of different technologies and mapping strategies. The difference in creative media technology application from scientific computing application is not a distinct boundary and crossover does occur. There does not appear to have been much interdisciplinary working between the scientific world and the art world.

The sonic artist and musician must be prepared to investigate the theoretical background in order to successfully employ this vast behaviour space within their compositional strategy. In using CA for generative music successful and sensitive application requires understanding of the science by the artist. Global dynamics and rule clustering are important concepts for the generative artist intent on using CA, and more importantly not esoteric or difficult to understand. The creation of scalable CA architectures and systems targeted at music applications can only be achieved by deeper research into this area. CA behaviour space will remain an untamed wilderness for music application unless these concepts are acknowledged.

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

MIDI domain, particularly in note sequencing alone and further investigations are still imperative. The domain of MIDI loudness dynamics, and other temporal parameters, has received less attention to date. The astounding variety of behaviour capable by such a simple mechanism as CA provides 'pattern for free'. Their place in the field of generative music has a long and interesting history. Will the electronic musical instrument manufacturers ever show an interest?

Acknowledgements

The authors would like to thank Andrew Wuensche for his continued support in allowing us to use data and present images made with *DDLab* software. The entropy plots of Figures 5 and 6 are originally presented in Wuensche (1997, 1999) and have been computed and graphed for this paper by Dave Burraston using *DDLab*.

Note

1 The true spirit of the underground, Iain von Trapp, died tragically as this paper was being completed. A great friend and fellow musician, this paper is dedicated to your memory.

References

Adamatzky, A. (1994) *Identification of cellular automata.* Taylor and Francis, Oxford.

Adamatzky, A. (2001) *Computing in nonlinear media and automata collectives*. Institute of Physics Publishing.

Bentley, P. (ed.) (2001) *Creative evolutionary systems.* Morgan Kaufmann.

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

Beyls, P. (1989) The musical universe of cellular automata. In Wells, T. and Butler, D. (eds) *Proceedings of the 1989 International Computer Music Conference*. International Computer Music Association, pp. 34–41.

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. *Proceedings of the 1991 International Computer Music Conference.* Montreal, Canada: International Computer Music Association. pp.254–257.

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

Beyls, P. (1998) Interactive cellular automata. *Evolution 2.0* CD-ROM, 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.

Beyls, P. (2004) Cellular automata mapping procedures. P*roceedings of the 2004 International Computer Music Conference*.

Bowcott, P. (1989) Cellular automata as a means of high level control of granular synthesis. *Proceedings of the 1989 International Computer Music Conference*. San Francisco, pp.55–57.

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. (2000) In search for musical fitness on consonance. *Electronic Musicological Review, Special Issue* **5**(3).

Bilotta, E. and Pantano, P. (2001) Artificial life music tells of complexity. *Proc. of Artificial Life Models for Musical Applications* (ECAL 2001 Workshop).

Bilotta, E. and Pantano, P. (2002) Synthetic harmonies: recent results. *Leonardo* **35**(2) 35–42.

Brown, A. (2005) Exploring rhythmic automata. *Proceedings of the 3rd European Workshop on Evolutionary Music and Art.*

Burks, A. (ed.) (1970) *Essays on cellular automata*. Univ. of Illinois Press.

Burraston, D. and Edmonds, E. (2004). Global dynamics approach to generative music experiments with one dimensional cellular automata. *Proceedings of the 2004 Australasian*

Computer Music Conference.

Burraston, D., Edmonds, E., Livingstone, D. and Miranda, E. (2004) Cellular automata in MIDI based computer music. *Proceedings of the 2004 International Computer Music Conference*.

Burraston, D. (2005a) Composition at the edge of chaos. *Proceedings of the 2005 Australasian Computer Music Conference,* Brisbane, July 2005.

Burraston, D. (2005b) One dimensional cellular automata musical experiments with Max. *Proceedings of the11th International Conference on Human-Computer Interaction*. Las Vegas, July 2005.

Burraston, D. (2005c) www.noyzelab.com

Candy, L. and Edmonds, E. (2002) *Explorations in art and technology.* Springer.

CCRMA (2005) ccrma-www.stanford.edu/courses/220b/lectures/4/

Chadabe, J. (1997) *Electric sound: the past and promise of electronic music*. Prentice Hall.

Chareyron, J. (1988) Sound synthesis and processing by means of linear cellular automata. *Proceedings of the 1988 International Computer Music Conference*.

Chareyron, J. (1990) Digital synthesis of self-modifying waveforms by means of linear automata. *Computer Music Journal* **14**(4) 25–41.

Coveney, P. and Highfield, R. (1995) *Frontiers of complexity: the search for order in a chaotic world*. Faber and Faber.

Davies, H. (1986) *Musical notation – old and new. Eye music: the graphic art of new musical notation*. Arts Council of Great Britain.

Delicate Ear (2005) http://cycling74.com/share/jhno/des/software.html

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

Dorin, A. (2001) Generative processes and the electronic arts. *Organised Sound* **6**(1) 47–53.

Dorin, A. (2002) Liquiprism: generating polyrhythms with cellular automata. *Proceedings of the 8th International Conference on Auditory Display*. Advanced Telecommunications Research International (ATR), Kyoto, Japan, July 2002, pp.447–451.

Dorin, A. (2005) www.csse.monash.edu.au/~aland

Dunn, J. (2005) *SoftStep manual.* www.geneticmusic.com

Edgar, R. and Ryan, J. (1986) LINA. *Exhibition of the 1986 International Computer Music Conference*.

Edmonds, E. (2003) Logics for constructing generative art systems. *Digital Creativity* **14**(1) 23–28.

Flake, G. W. (1999) *The computational beauty of nature.* MIT Press.

Flanagan, J. L. and Golden, R. (1966) Phase vocoder. *Bell Systems Technical Journal* **45** 1493–1509.

FractMus (2005). www.geocities.com/SiliconValley/Haven/4386/

Fredkin, E. (1992) Finite nature. www.digitalphilosophy.org

Freff, C. (1989) Raymond Scott's Electronomium (1965). *Keyboard* **15**(2) 50–56.

Gabor, D. (1947) Acoustical quanta and the theory of hearing. *Nature* **4044** 591–594.

Gardner, M. 1970. The fantastic combinations of John Conway's new solitaire game of 'life'. *Scientific American* **223**(4) 120–123.

Griffeath, D. and Moore, C. (2003) *New directions in cellular automata*. Oxford University Press.

Hiller, L. and Isaacson, L. (1959) *Experimental music.* McGraw Hill, New York.

Hiller, L. (1970) Music composed with computers – a historical survey. In Lincoln, H. (ed.) *The computer and music.* Cornell University Press. Ithaca, pp.42–96.

Hunt, A., Kirk, R. and Orton, R. (1991) Musical applications of a cellular automata workstation. *Proceedings of the 1991 International Computer Music Conference*. Montreal, Canada, pp.165–168.

IRCAM (2005) ftp://www.forumnet.ircam.fr/pub/max

IsleEx (2005) http://jmge.net/camusic.htm

Karplus, K. and Strong, A. (1983) Digital synthesis of plucked-string and drum timbres. *Computer Music Journal* **7**(2) 43–55.

Katrami, A. I., Kirk, R. and Myatt, A. (1991) Manipulation of cellular automata and fractal landscape mappings. *Proceedings of the 1991 International Computer Music Conference*. Montreal, Canada, pp.106–109.

Kreger, T. (1997) Cellular automata in the spectral domain. *Proceedings of Interface 1997*. University of Auckland, Auckland.

Kreger, T. (1999) Real-time cellular automata filters implemented with Max MSP. *Proceedings of the Australasian Computer Music Conference 1999*. Victoria University of Wellington, N.Z.

Langton, C. G. (1990) Computation at the edge of chaos: phase transitions and emergent computation. *Physica D* **42** 12–37.

Langton, C. G. (1991) Life at the edge of chaos. In *Artificial Life II, Proceedings Vol. X. SFI Studies in the Sciences of Complexity*. Addison-Wesley.

Levy, S. (1992) *Artificial life: the quest for a new creation*. Jonathon Cape, London.

Li, W. (1989) Complex patterns generated by next nearest neigbors cellular automata. *Computers and Graphics* **13**(4) 531–537.

Li, W. Packard, N. H. and Langton, C. G. (1990) Transition phenomena in cellular automata rule space. *Physica D* **45** 77–94.

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. (1995) Sound synthesis for one dimensional reaction-diffusion systems. *Proceedings of the ACMA 1995 Conference.* Australasian Computer Music Association.

Martin, A. (1996a) 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.

Martin, A. (1996b) Reaction-diffusion system for algorithmic composition. *Organised Sound* **1**(3).

McAlpine, K., Miranda, E. R. and Hoggar, S. (1999) Making music with algorithms: a case-study system. *Computer Music Journal* **23**(2).

McCormack, J. (2003) Art and the mirror of nature. *Digital Creativity* **14**(1) 3–22.

McIntosh, H. V. (1990) What has and hasn't been done with cellular automata. http://delta.cs.cinvestav.mx/~mcintosh/newweb/marcowhat.html

Meinhardt, H. (2003) *The algorithmic beauty of sea shells*. Springer.

Millen, D. (1990) Cellular automata music. *Proceedings of the 1990 International Computer Music Conference*. ICMA, San Francisco, pp.314–316.

Millen, D. (1992) Generation of formal patterns for music composition by means of cellular automata. *Proceedings of the 1992 International Computer Music Conference*. San Francisco: ICMA, pp.398–399.

Millen, D. (2004) An interactive cellular automata music application in Cocoa. *Proceedings of the 2004 International Computer Music Conference*.

Millen, D. (2005) comp.uark.edu/~dmillen/cam.html

Miranda, E. R. (1993) Cellular automata music: an interdisciplinary project. *Interface* **22**(1) 3–21.

Miranda, E. R. (1995a) Granular synthesis of sounds by means of a cellular automaton. *Leonardo* **28**(4) 297–300.

Miranda, E. R. (1995b) Cellular automata synthesis of acoustic particles. *Banff ICMC 95 Proceedings*, pp.233–234.

Miranda, E. R. (2001) *Composing with music computers.* Focal Press.

Miranda, E. R. (2002) *Computer sound synthesis for the electronic musician*. 2nd Edition, Focal Press.

Miranda, E. R. (2003). On the evolution of music in a society of self-taught digital creatures. *Digital Creativity* **14**(1) 29–42.

Nisho, H. (1975) A classified bibliography on cellular automata theory – with a focus on recent Japanese references. *Proc. Int. Symp. On Uniform Structures, Automata and Logic.* Tokyo.

NYR (2005). www.nyrsound.com

Oliveira, G., Oliveira, P. and Omar, N. (2001) Definition and application of a five-parameter characterization of one-dimensional cellular automata rule space. *Artificial Life* **7** 277–301.

Olsen, H. F. and Belar, H. (1961) Aid to music composition employing a random probability system. *J. Acoust. Soc. Am*. **33** 1163–1170.

Orton, R., Hunt, A. and Kirk, R. (1991) Graphical control of granular synthesis using a cellular automata and the Freehand program. *ICMC 1991 Proceedings,* Montreal, pp.416–418.

Penfold, R. A. (1995) *Practical MIDI handbook*. 3rd Edition. PC Publishing.

Roads, C. (1978) Automated granular synthesis of sound. *Computer Music Journal* **2**(2) 61–62.

Roads, C. (1996) *The computer music tutorial*. MIT Press.

Robot Software. (2005) www.robotsoftware.co.uk

Rucker, R. (2005) sjsu.rudyrucker.com/~karl.schramm/applet/

Rumsey, F. (1994) *MIDI systems and control*. 2nd Edition. Butterworth-Heinmann.

Russ, M. (1997) *Sound synthesis and sampling*. Focal Press.

Scaletti, C. (1997) *The Kyma language for sound design V4.5.* Symbolic Sound Corporation.

SCOL (2005) swiki.hfbk-hamburg.de:8888/MusicTechnology/163

Sipper, M. (1997) *Evolution of parallel cellular machines: the cellular programming approach.* Springer Verlag.

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

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

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

Truax, B. (1988) Real-time granular synthesis with a digital signal processor. *Computer Music Journal* **12**(2) 14–26.

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* **25**(4) 289–298.

Vaidhyanathan, S., Minai, A. and Helmuth, M. (1999) ca : a system for granular processing of sound using cellular automata. *Proceedings of the 2nd COST G-6 Workshop on Digital Audio (DAFx99) 1999*. NTNU, Trondheim.

Varga, B. A. (1996) *Conversations with Iannis Xenakis*. Faber and Faber, London.

Vorn, B. (2005) www.billvorn.com

Wolf, S. and Thompson, A. (2002) The Sound Gallery – an interactive A-Life artwork. In Bentley, P. (ed.) *Creative evolutionary system*s. Morgan Kaufmann.

Wolfram, S. (1983). Statistical mechanics of cellular automata. *Reviews of Modern Physic*s **55**(3) 601–644.

Wolfram, S. (1984). Universality and complexity in cellular automata. *Physica D* **10D** 1–35.

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. (Available as PDF from www.ddlab.com)

Wuensche, A. (1997) *Attractor basins of discrete networks.* Cognitive Science Research Paper 461, Univ. of Sussex, D.Phil thesis.

Wuensche, A. (1999) Classifying cellular automata automatically: finding gliders, filtering, and relating space-time patterns, attractor basins, and the Z parameter. *Complexity* **4**(3) 47–66.

Wuensche, A. (2005) The DDLab manual. Discrete Dynamics Inc. (Available as PDF from www.ddlab.com)

Xenakis, I. (1992) *Formalised music*. Revised Edition. Pendragon Press.

Zuse, K. *Rechnender Raum* (Vieweg, Braunschweig, 1969); translated as *Calculating space*, AZT-70-164- GEMIT MIT Project MAC.