One Dimensional Cellular Automata Musical Experiments with Max

Dave Burraston

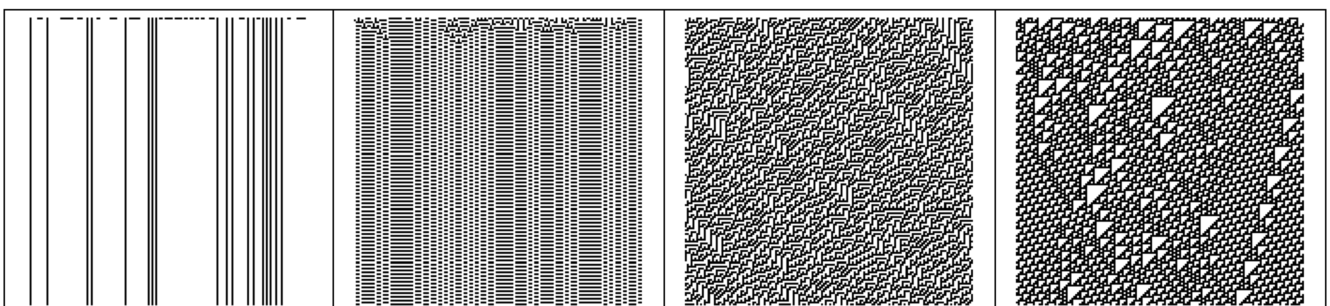
Creativity and Cognition Studios (CCS),
Faculty of Information Technology,
University of Technology Sydney (UTS),
Sydney,
Australia
dave@noyzelab.com

Abstract

This paper will give a significant shift of context and awareness for the artist or technologist interested in how an artist approaches creating Cellular Automata (CA) based generative music. One Dimensional CA offer the use of emergent computation and behaviours as compositional aids to the generative music process. CA have a diverse history beginning with parallel computation and now forms one of the cornerstones of the field of Artificial Life. Global dynamics and rule clustering have been used in CA research for over a decade and provide a different perspective on their behaviour. A methodological approach to generative music creation is described by reflective practice techniques, exposing the main aspects of the creative process. An interactive generative CA music system is presented which forms the basis for creating music compositions. Visualisations of CA evolutions and data are produced to illustrate behaviour that has been mapped to musical parameters. Evaluation of the results of this system will be judged against recognised criteria. The conclusion and future directions for this work are then discussed.

1 Introduction

Algorithmic and computational processes are an important tool for the technology based creative artist producing generative art systems (Dorin, 2001), (Candy & Edmonds, 2002), (Edmonds, 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 musicians for many years, assisting an emerging culture of artificial aesthetics and algorithmic generative electronic art.



Class 1 fixed behaviour	Class 2 cyclic behaviour	Class 3 chaotic behaviour	Class 4 complex behaviour

Figure 1: Examples of Wolfram's four classes of CA behaviour

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 behaviours and represent an important generative tool for the artist. Musical Instrument Digital Interface (MIDI) was the first manufacturer wide standard enabling the interconnection of electronic instruments to each other and to computers (Rumsey, 1994). A review of CA in the MIDI domain is presented in (Burraston, Edmonds, Livingstone & Miranda, 2004).

Complexity theory demonstrates that complex systems of simple units, such as the cells in a CA, produce a variety of behaviours. Complex systems such as CA produce global behaviour based on the interactions of these 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). 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. 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 have been classed by Stephen Wolfram with one of four behaviours (Wolfram, 1984) as shown, above (Figure 1) with cell space in the horizontal and time evolution running vertically downwards.

The global dynamics of CA (Wuensche & 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. Wuensche's Discrete Dynamics Lab (DDLab) software allows for the exploration of global dynamics (Wuensche, 2001). A brief overview of Wuensche and Lesser's work on global dynamics and rule clustering is presented in the next section. A detailed example of MIDI experiments based on global dynamics is presented in (Burraston & Edmonds, 2004), (Burraston 2005).

2 Global Dynamics and Rule Clustering

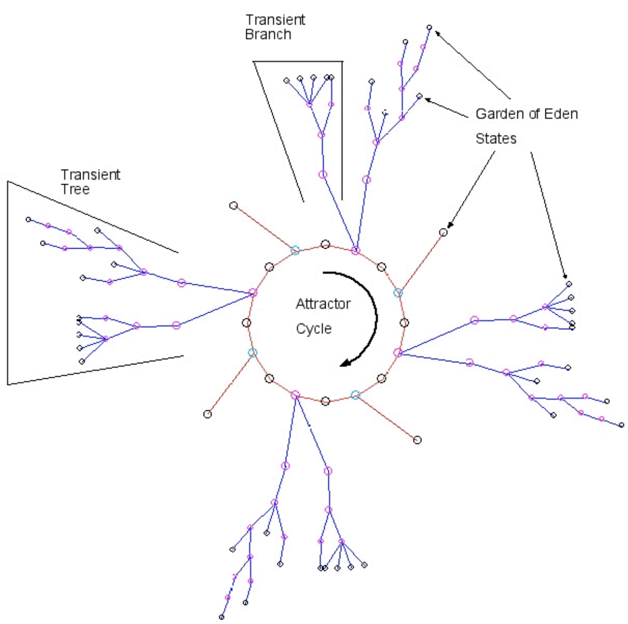


Figure 2: Basin of attraction

A CA state space consists of all possible global states. In a finite deterministic CA all state transitions must eventually repeat with period 1 or more. States are either part of an attractor cycle or lie on a transient leading to the attractor cycle. If a transient exists there will be states unreachable by any other states at the extremity. These extremities are called garden of Eden (**goE**) states. All transients leading to an attractor, and the attractor cycle, is termed the basin of attraction (**boa**) of that individual attractor. An example basin of attraction is shown, above (Figure 2). State space for a particular CA rule and size is populated by one or more basins of attraction, termed the basin of attraction field. The boa field may contain equivalent basins, where the states of other basins are rotationally symmetric.

DDLab constructs boa fields by the computation of pre-images of all states by a reverse algorithm, and can suppress equivalent basins during display. Measures can be taken of the number of basins in the field and garden of Eden states, the attractor periods, basin sizes, maximum transient and cycle period. Attractor basin topology reflects the dynamics of a CA rule and can be used as a method of identifying ordered, complex and chaotic behaviour. The goE density and in-degree frequency distribution of pre-images are two measures of CA dynamics (Wuensche, 1997). The in-degree is the number of pre-images of a node in the boa field and is plotted as a histogram. Ordered rules are classified by very high goE density and large numbers of high in-degrees. Complex rules will have medium values for both. Chaotic rules have a low goE density and large numbers of low in-degrees.

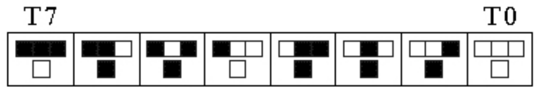


Figure 3: Transition rule table (rule 110)

In DDLab the number of state values is represented by **v** and the number of cells in the neighbourhood by **k**. The 1D v2k3 transition rule is specified by an 8 bit binary number between 0 and 255 and an example for rule 110 is shown, above (Figure 3). The 8 entries in a rule transition table are defined as **T7** to **T0** left to right. The 1D v2k3 rules can be grouped into 88 equivalence rule classes, by **negative**, **reflective** and composite **negative reflective** transforms, with a maximum of four rules per equivalence class (Walker & Aadryan, 1971). In a rule cluster three basic rule table transformations are defined as shown, below (Table 1). The negative transform is inversion followed by four pairs of entries having their entries swapped. For example in the first pair **T0** becomes **T7** and **T0** becomes **T7**. The remaining pairs indicated are swapped in the same manner. The reflection transform involves pair swapping {**T6** <-> **T3**} and {**T4** <-> **T1**}. A further method called the **complementary** transform inverts the rule table contents. The rule cluster axis is shown, below (Figure 4 left) and the rule cluster layout is shown, below (Figure 4 right).

Table 1: Rule Transformations

Negative	Rule table is inverted & T pairs swapped : {**T0** <-> **T7**}, {**T1** <-> **T6**}, {**T2** <-> **T5**}, {**T3** <-> **T4**}
Reflection	T Pairs swapped : {**T6** <-> **T3**} and {**T4** <-> **T1**}
Complementary	Rule table entries are inverted

The lowest rule number (**R**) identifies a cluster and is always positioned in the top left corner. The negative (**Rn**) and reflection (**Rr**) transforms are identified along with a composite transform, the negative reflection (**Rnr**). The complement transform (**Rc**) also has corresponding negative (**Rcn**), reflection (**Rcr**) and negative reflection (**Rcnr**) transforms.

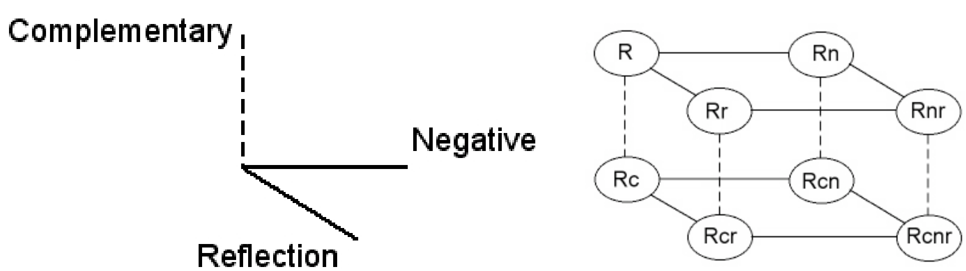


Figure 4: Rule cluster axis (left) and layout (right)

Each rule cluster has two boa fields, one for the top layer (**R**) and one for the complementary bottom layer (**Rc**). Rule clusters contain 2, 4 or 8 different rules depending on whether the transformations result in the same rule number. For both layers the other transformed rules have identical boa measures and the states in spacetime are simply related by being negative (**Rn**), mirror image (**Rr**) or both (**Rnr**), below (Figure 5). In some cases the clusters collapse further leaving no complementary rules, resulting in a single boa field. The symmetry categories are symmetrical, semi-asymmetrical and fully asymmetrical contained in a total of 48 rule clusters. Symmetric rules contain 2 or 4 rules per cluster, Semi-asymmetric rules have no collapsed clusters and always contain 8 rules. Fully asymmetric rules contain 4 or 8 rules per cluster. Examples of one of each possible cluster type for all categories is shown, below (Figure 6).

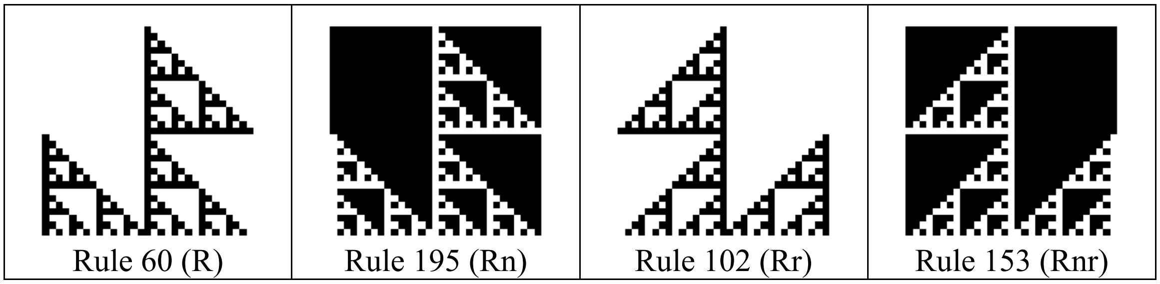


Figure 5: Rule 60 equivalent spacetime patterns

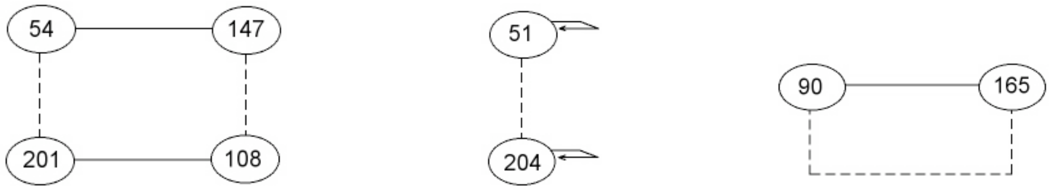


Figure 6a: Symmetric rule clusters, three types

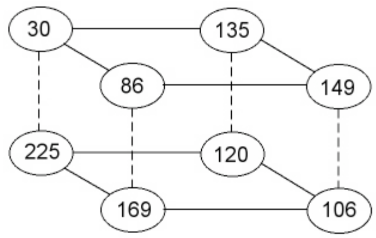


Figure 6b: Semi-asymmetric rule clusters all have this type

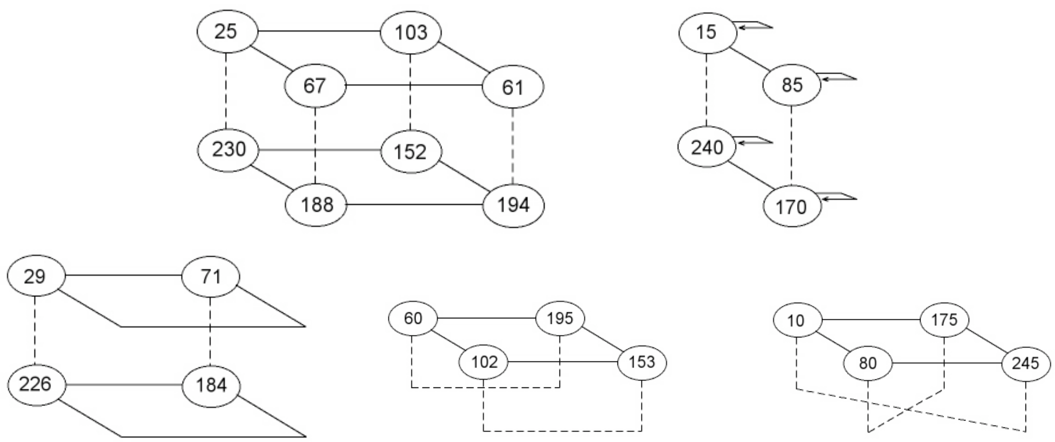


Figure 6c: Fully asymmetric rule clusters, five types

Rule cluster 15 is important because it exhibits behaviour that is directly comparable to that of analogue music sequencers. The cluster diagram for rule 15 is shown, above (Figure 6c). Rules 170 and 240 are known as the left and right shift rules, and form the complement part of the cluster. Any input pattern will be shifted left (rule 170) or right (rule 240) by one cell per CA step. Analogue sequencer modules commonly have 8, 12 or 16 steps and can be stepped sequentially one stage at a time from between 1 and the maximum number of steps available. This operation is from left to right and vice versa, analogous to rule 170 and 240 with a single active cell.

3 Reflective Practice and Assessment Criteria

Reflection-in-action is a method of researching within a practical context (Schon, 2003). The structure of reflection-in-action in practice based research, termed reflective practice, suggests move testing experiment as a useful methodology. In move testing an action is undertaken to produce intended change. Good results will affirm that move and bad results will negate it. Importantly unintended results that are good, should also affirm the move. Reflection on outcome and potential for future directions will be discussed.

Technical criteria for evaluating sound synthesis techniques were suggested by (Jaffe, 1995), however these are recognised as guides not rigid definitions. These criteria were suggested in the context of sound synthesis, however some of the criteria applies to generative music production with MIDI. The most appropriate criteria in this context will now be discussed. How efficient is the algorithm? Defined as an extremely important, yet context dependant criterion. Three efficiency categories suggested are memory, processing and control stream. How sparse is the control stream? This is related to efficiency and has particular impact on a real time system. How robust is the sounds identity? Asks if parametric adjustment alters the sound too extremely. What classes of sounds can be represented? Conversely this asks if a broad range of sound classes can be achieved by parametric adjustment.

These guideline criteria were revisited and extended in the context of physical modelling sound synthesis (Castagne & Cadoz, 2003). The context of physical modelling is relevant because it addresses the entire musical creation process and does not just represent a sound synthesis perspective. Ten criteria for evaluating physical modelling were defined in four areas : computer efficiency, phenomenology, usability of scheme and environment for using the scheme. The efficiency area is similar to Jaffe's proposed scheme. An extended phenomenological criteria is diversity of context and relates to Jaffe's robustness and sound class criteria. In usability particular emphasis is placed on modularity. Modular principles are a very important criterion in obtaining generality, power and simplicity. The environment for using the scheme has two criterion. The first studies if generation algorithms already exist and how effective they are. The second asks if there is a friendly musician oriented environment for using this scheme. Results of this work will be judged on these criteria.

4 Initial Experimental System

The move testing generative music experiment is defined as : **Action** : Construct visualisations and Max patch with a small, single rule CA. **Intended Change** : Simple, easy to use system with future scalability. The Max patch was implemented using Bill Vorn's 1D CA Max external. This is part of Vorn's LifeTools, available on the internet, and forms part of the IRCAM library of Max objects (Vorn, 1996) for Cycling 74's Max programming language ("Cycling 74", 2004). Rule 30 has been chosen simply because this is the default rule in Vorn's external. Once the Max patch is built as a standalone application this will be the only rule available. Changing the rule is possible when using the patch within Max. A CA size of 8 cells was chosen to directly map to a simple 8 voice system. The approach taken in building an experimental music system has some similarity to the University of York's Cellular Automata Workstation, for the Atari ST (Hunt, Kirk & Orton, 1991). The work described here represents a different approach due to the global dynamics perspective, and the system is viewed as a modular foundation for future work. A simple realtime system, named CA Simplistic Selector (CASS), was produced to map the CA cells to a set of adjustable MIDI events. This set of events will be termed a Cellular MIDI Event (CME) module. The system thus comprises of 8 CME modules, each module is connected to a unique corresponding cell. A block diagram and screenshot of the experimental system is shown, below (Figure 7).

Each CME is controllable in real time by the performer. The CME comprise of note, velocity, duration, MIDI channel and program controls. The system can thus be split up over MIDI channels as desired for multitimbral or one channel for single voice operation. A keyboard for each CME allows for note selection and playing within a seven octave range, a dialogue allows for note input for the full 0 – 127 range. MIDI velocity, channel and program can be input over their complete ranges. Duration is entered by a slider or dialogue from 10 to 10,000 milliseconds. The event onsets are controlled by the 1D CA, which may be seeded randomly, with a list or by a mouse. Individual CME's can be muted during performance. The performer has overall control of tempo and modifies the events to be triggered while the CA runs. This presents an interesting juxtaposition between performer and automated machine. A screen shot of the CASS system showing a close up of the CME module is shown, below (Figure 8 left). All of the CME settings can be saved and recalled for the whole system with the Max preset object.

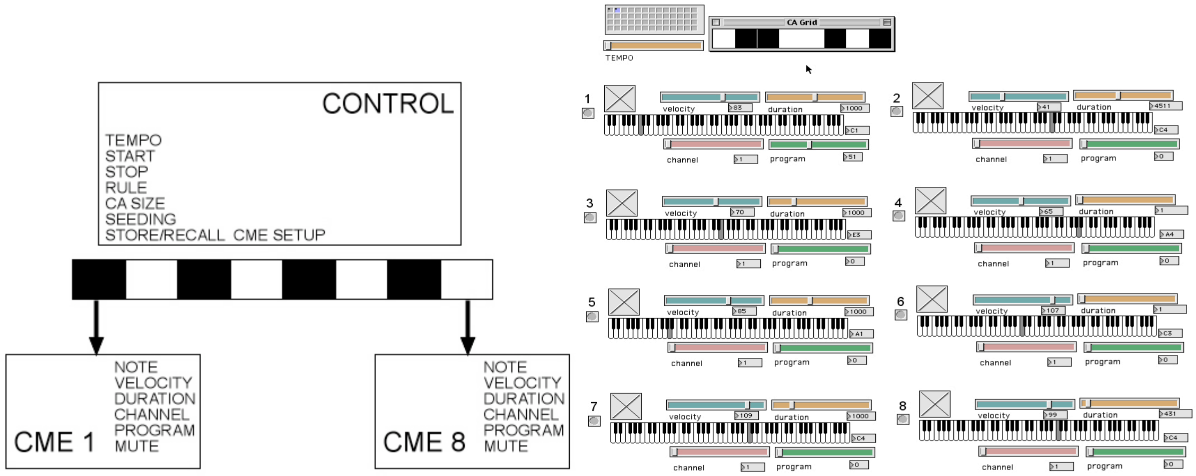


Figure 7: Block diagram of CASS experimental system (left) and screen shot (right)

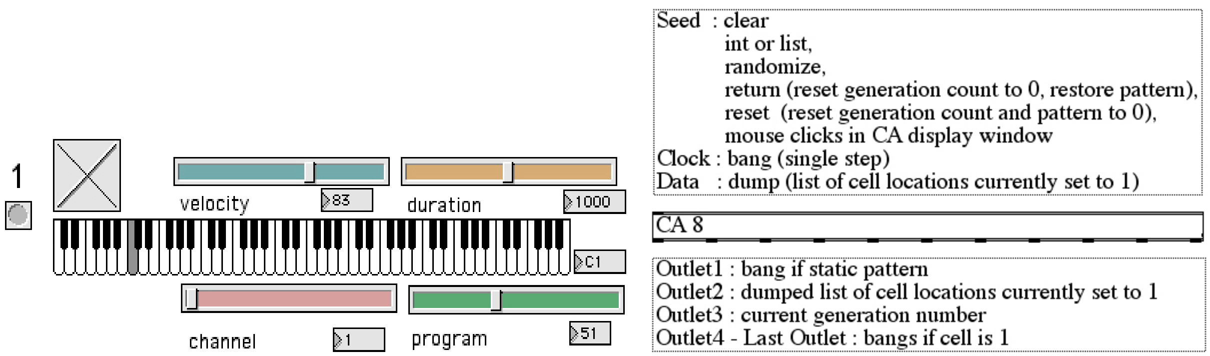


Figure 8: CME module (left) and I/O functionality of Vorn's CA Max external (right)

The I/O functionality of Vorn's CA Max external is shown, above (Figure 8 right). The inputs are categorised as seed, clock and data. A number of seeding options are available and the output can be taken in the form of bang messages from each cell or dumped as a list of active cells to outlet 2. Resetting of the CA by message is possible, either to an initial seed or to 0 for all cells. The CME modules are wired directly to the cell outputs. The CA external will send out a bang from outlet 1 if the output remains static. This allows for the inclusion of automatic reseeding mechanisms to be built within a patch. The current generation count is output as an integer from outlet 3.

Rule 30 is from the semi-asymmetric category and the cluster is shown, above (Figure 6b in Section 2). A boa field for the eight cell Rule 30 was constructed with DDLab and contains 256 states. The main boa is of period 40 and contains 224 states as shown, below (Figure 9 left). The longest transients in this boa contain 16 states and 2 states for the shortest. The remaining states are contained within four boa and are shown labelled in decimal, below (Figure 9 middle and right). These are a period 8 attractor, two single state period 1 attractors and a two state period 1 attractor. Spacetime plots of three attractor cycles from four example seed states are shown, below (Figure 10 left). Space is the horizontal axis and time progresses vertically downwards for 56 generations. Running the system for 56 generations allows for the maximum transient and cycle period to appear. From left to right these plots are; Period 1 with seed = 170, period 8 with goE seed = 3, longest transient to period 40 with goE seed = 157 and shortest transient to period 40 with goE seed = 110. The period 40 patterns show how the cycle will be entered at a different phase, and is also seen topologically on the boa diagrams. Generally speaking the period 8 and period 40 patterns appear to move right to left over time. The period 8 attractor looks reasonably ordered, whereas the period 40 visually has a slightly more random appearance in comparison A pre-image in-degree histogram is shown, below (Figure 10 right). The goE nodes with zero in-degree pre-images are in the left column, the remainder of the field has 1 or 2 in-degree pre-images. There is a low goE density, and high frequency of low in-degrees indicating by Wuensche's classification that this is a chaotic rule (Wuensche 1997).

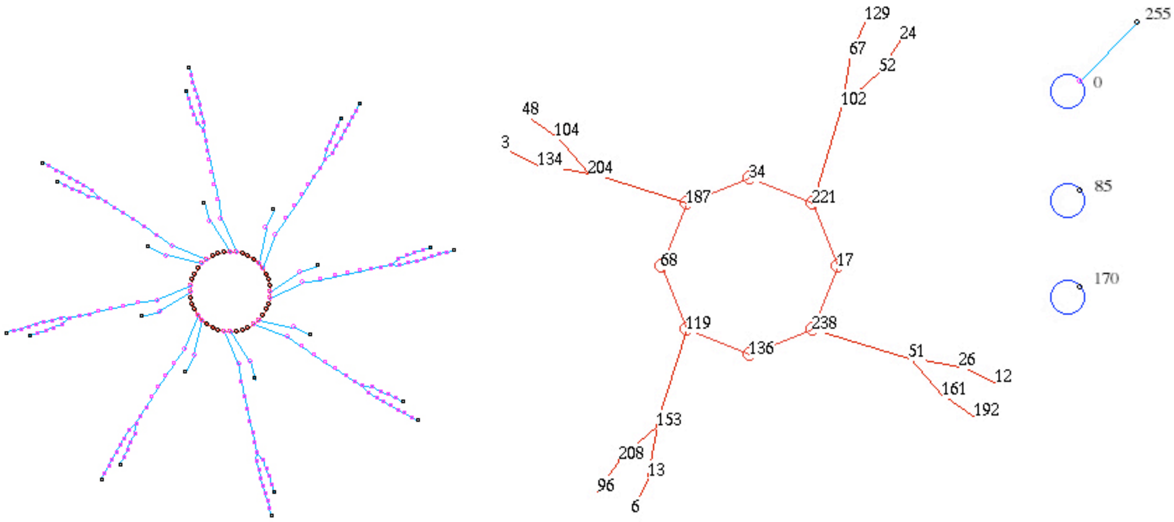


Figure 9: Eight cell rule 30 basin of attraction period 40 (left) and period 8 (middle) and 3 x period 1 (right)

Figure 10: Spacetime plots of 8 cell rule 30 with decimal seed states indicated (left) and in-degree histogram, indicating a chaotic dynamic (right).

5 Evaluation and Reflection : Towards a Modular System

Experimental results are now evaluated in terms of the four criteria areas; computer efficiency, phenomenology, usability of scheme and environment for using the scheme. Reflections will also be given on the system in use as part of the generative composition process and a glimpse of the current incarnation of the system is presented.

In terms of memory the 8 cell CA is very efficient. The CASS system is able to run and respond to user events in realtime over a broad tempo range. Vorn's CA external will not run in Max overdrive mode and this results in interruptions in MIDI output scheduling during some GUI events. This does not adversely affect studio use, but is sometimes noticeable in a performance context. Overall the control stream is light and messages are not sent when a cell is not active. This MIDI control stream is minimised by the simple and limited nature of each CME module. MIDI Channel and Program Change are only sent when changed by the user. MIDI Continuous Controllers such as Pitchbend, Mod Wheel and Pan have subsequently been informally tested without much additional efficiency overhead. It should be noted that when investigating the global dynamics with DDLab for larger numbers of cells there will be a practical limit (Wuensche, 2001).

Regarding phenomenological criteria, successful diversity of application context was achieved, as was robustness. The main quality of the system is a temporal mapping of the CA to specified sounds. The aesthetic results produced during the composition process up to date have been both rhythmic and textural. Four compositions emerged from rule clusters 15 and 30, exemplifying these approaches. The compositions from rule cluster 15 are titled "Right shift toggle" and "Right shift". Both of these are ambient compositions using a microtonal scaling of 96 tones per octave. "Right shift toggle" contains some light percussive elements and bell based textures. The CME parameters were selected at random and the only further interaction I allowed myself was changing the CA state by mouse clicks, muting/unmuting CME modules and slowly reducing the tempo at the end. "Right shift" contains synthetic strings with a sweeping envelope controlling a lowpass resonant filter. The CME parameters were chosen manually and also adjusted during the piece. The tempo of this piece was kept constant and interaction with CA state was permitted. The compositions relating to rule cluster 30 are titled "Ubendem, Wemendem" and "Cassorgize". "Ubendem, Wemendem" uses rhythmic elements in the form of percussive sounds triggered by selected CME modules. The majority of rhythm in this piece was created by selecting sounds while the system ran and storing them to build up a few basic patterns. "Cassorgize" is a slower tempo textural piece, using filtered organ sounds with a slow attack and long release. This approach involved selecting sounds first by assigning event parameters and then running the CASS system from a variety of different seed states.

Modular principles are a very important usability criterion in obtaining generality, power and simplicity. CASS has been successful by taking a basic modular approach. The modularity of this system could easily be extended by the addition of a simple switching matrix between the CA cells and the CME modules. This would allow for a single cell to trigger multiple CME's. The system should be able to scale easily both in CA size and the number of CME's, but this will have a bearing on efficiency criteria. Rule cluster 15 is particularly important as it has provided a direct link with modular analogue sequencers and functions as a prototype exemplar to the electronic musician.

Some I/O problems have been identified in the context of modular based generative music systems. The seed messages are passed as a list of active cell numbers and having an integer or binary seed is preferable. The cell output of the system would also benefit from being available in this format. The CA size is currently adjustable from 4 to 256 cells, which is perfectly adequate. However, changing the number of cells involves changing the argument of the external within the Max patch. An additional problem with changing CA size is that the rule is always reset to its default of 30. The rule can only be changed when the patch is being edited, using the Get Info command. The system would benefit by having these parameters available as inputs.

The overall environment for using the scheme is simple and easy to use. Generation algorithms exist in the context of both visualising CA dynamics, and within the Max programming environment. Both have been effective for generative music research and production with the CASS 8 cell experimental system. Global dynamics is easily studied on a variety of computer platforms with DDLab, which also has the added benefit of producing basin measures and diagrams in one software environment. Preparation of spacetime diagrams to confidence check Vorn's CA external was aided by the ability to label basin nodes with state values, both in decimal and binary. The Max programming environment allows for the investigation of experimental systems within a live and studio context. Vorn's CA is a Mac OS9 based external, currently untested and unlikely to work for OSX. This does not present a problem at present but future work will require an OS9 machine to be available or a new OSX based module. A 1D CA is available in Jitter, but this does not appear to implement periodic boundary conditions and an upgrade from Max is also required.

After considering these results and reflections a prototype 2 voice sequencer was constructed for a live performance. This contains multiple 1D CA triggering events and now also generating the MIDI data values. These data values have been expanded to include note, velocity, duration, pitchbend, modwheel, pan and tempo. The CA output can rescaled to each parameter by the performer and a module diagram is shown, below (Figure 11). The distinction between performer and CA generative aspects can be seen. The potential for scalability of the system with additional parameters is also identified. Performance control can be increasingly handed over to the CA. An important issue of efficiency now is that parameters may be calculated and not used if a Note On command is not produced. This is not a huge overhead in a simple system but would compound in a large system. The system has been used successfully for live performance and within the studio environment. A more detailed critique of version 2 is in preparation.

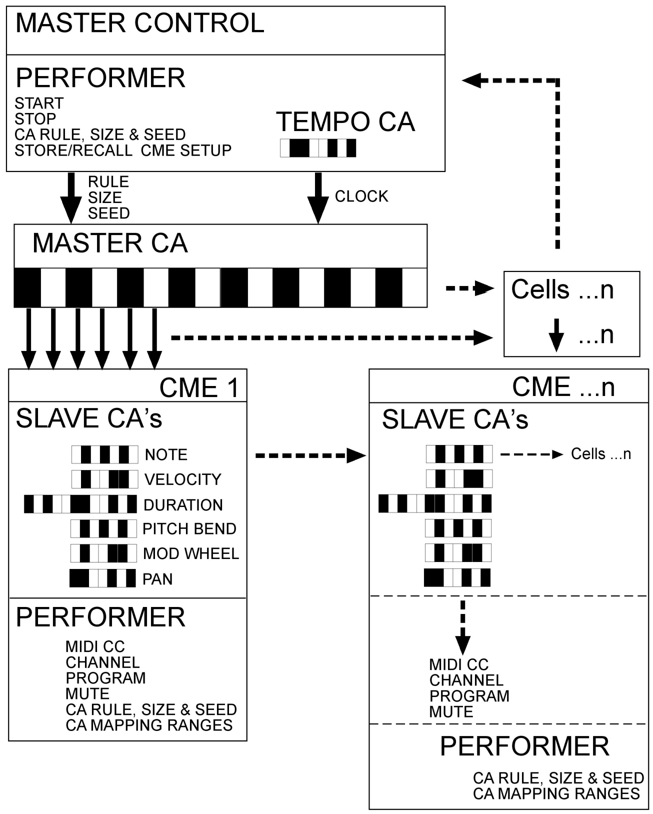


Figure 11: Modular diagram of CASS V2 and indication of scalability

6 Conclusions and Future Work

Art and science are different yet fundamentally connected disciplines. Here the science has inspired and assisted the artist to achieve specific results. It is hoped that the artwork produced will inspire the science community and generate better understanding in that community of the musical applications of CA. 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. The vast behaviour space of 1D CA will remain an untamed wilderness for music application unless these concepts are acknowledged, and actively researched in a musical context.

Vorn's CA external was created for his robotic art systems and not as a modular music component. The criticisms of its functionality are to be taken based on criteria for music composition, and not in terms of his individual application. The current work is made possible because of his placement of them within the public domain. The CASS system has been used in both performance, and studio production of generative compositions. It is a simple to use musical interface to CA dynamics, both within the Max environment and as a standalone application. Future work can now expand on this foundation and investigations are planned for a number of areas. These areas are currently identified as strategies for rule choices, number of cells, seeding mechanisms and musical mappings. The methodology presented in this paper will serve as the foundation for future music research in modular CA systems.

7 Acknowledgements

The author would like to thank Andrew Wuensche for allowing me to present basin of attraction data and images made at CCS with DDLab. Figures 2, 9 and 10 (right) were made with DDLab. Also thanks to Andrew Wuensche for allowing me to draw and present the rule clustering images. Figures 2, 4, 6, and rule 193 version of Figure 5 originally appear in (Wuensche & Lesser, 1992) and have been redrawn for this paper by Dave Burraston. Figures 1, 5 and 10 (left) were made at CCS with Mathematica. Figure 3 was made at CCS with NKS Explorer.

8 References

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

Burraston, D. Edmonds, E. Livingstone, D & Miranda, E. (2004). Cellular Automata in MIDI based Computer Music. *Proceedings of the 2004 International Computer Music Conference*.

Burraston, D. & 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. (2005). Composition at the Edge of Chaos. *Proceedings of the 2005 Australasian Computer Music Conference.*

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

Castagne, N. & C. Cadoz. (2003). "10 Criteria for Evaluating Physical Modelling Schemes for Music Creation." *Proc. of the 6th Int. Conference on Digital Audio Effects.*

Cycling 74. (2004). www.cycling74.com

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

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

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: International Computer Music Association. 165-168.

Jaffe, D. (1995). "Ten Criteria for Evaluating Synthesis Techniques." *Computer Music Journal* 19 (1), 76-87.

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

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

Rumsey, F. (1994) MIDI Systems and Control. 2nd Edition. Butterworth-Heinmann.

Schon, D. (2003). The Reflective Practitioner. Ashgate Publishing.

Vorn, B. (1996). *LifeTools.* www.billvorn.com or from the IRCAM ftp server.

Walker, C. C. & Aadryan, A. A. (1971). Amount of Computation Preceeding Externally Detectable Steady-State Behaviour in a Class of Complex Systems. *Bio-Medical Computing,* 2, 85-94.

Wolfram, S. (1984). Universality and Complexity in Cellular Automata. *Physica D*. 10D, 1-35.

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

Wuensche, A. (1997). Attractor Basins of Discrete Networks*.* Cognitive Science Research Paper 461, Univ. of Sussex, D.Phil thesis.

Wuensche, A. (2001). The DDLab Manual*.* Discrete Dynamics Inc. www.ddlab.com