Stochastic processes in the musification of cellular automata: A case study of the Livecell project

Kingsley Ash
Echochroma New Music Research Group,
Leeds Metropolitan University, UK
K.Ash [at] leedsmet.ac.uk

Nikos Stavropoulos
Echochroma New Music Research Group,
Leeds Metropolitan University, UK
N.Stavropoulos [at] leedsmet.ac.uk

This paper discusses Livecell, a system for interactive generative composition through real-time score generation as well as the triggering and treatment of sound files. The paper outlines the authors' rationale, discusses the use of cellular automata in this context and provides an insight into the application's structural design. The authors summarise their approach to CA musification, the representation of data relations in a musical score, and interface design as well as score generation and display. The paper concludes with observations on the musical output and discussion of future developments.

Adaptation Livecell combines elements of generative composition using cellular automata (CA) and stochastic processes with real-time score generation in addition to the triggering and treatment of sound files, all controlled by real –time user input. The CA graphical user interface enables interactive control of the system, whilst the real-time score generation allows for the immediate realisation of the notated output by a string quartet concurrent with the audio output. The tripartite structure of the system comprises the CA graphical user interface, the data musification engine and the score generator [Figure 1]. Musification is defined as the representation of data relations in a musical score. This is distinct from sonification, in which data is represented directly in acoustic signals.

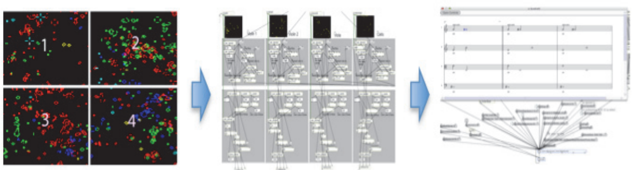

Figure 1. System structure

The user is able to interact with the system by drawing and erasing cells into the CA grid via the graphical user interface. Different regions of the interface correspond to the different instruments in the string quartet. Cells are able to grow and move between these regions allowing the composition and the instrumentation to evolve under the direction of the user and through the natural evolution of the cellular automata. Furthermore, the interface allows access to a collection of cellular automata rule sets, which correspond to distinct macroscopic behaviours, as well as a series of pitch class sets and network and audio controls.

Rationale

The initial objective behind this project was to develop a system for composition and performance that would allow a single user to generate and control a score, alongside digital audio, in real-time. CA have been employed as a component in the system to enable greater musical complexity than would be achievable by a single user directly controlling all available parameters. The use of string instruments stems from their capacity to produce a wide range of timbral variations using established techniques thus allowing the development of a system which produces sonically diverse output without resorting to extended instrumental techniques.

The system generates a musical discourse that displays complexity and temporal development analogous to the graphical realisation of the CA. These analogies manifest themselves either in a descriptive manner, where direct correspondence between the state of the CA on the screen and the musical events in the score / performance is explicit, e.g. increased cell activity results in faster rhythms, or via association, where behaviours and/or structures that appear in the graphical representation are associated with distinct musical schemata, e.g. a particular set of CA rules produces ostinati with subtle rhythmic and melodic variations. The rationale behind this approach is "to guide selection and manipulation of variables or attention to specific outcomes, for users who have no model or prior knowledge" (Brophy / Schwartz 1998) of the system and the musification algorithm.

Billota and Pantano, in their work on musical semiosis by means of cellular automata, emphasise that "the translation of a sequence of sounds through physical parameters does not amount to generating music [sic]" acknowledging that the syntax of the musical work, unlike that of CA, is culturally developed and everchanging (Billota / Pantano 2002). Evidently there are numerous examples where the invariant rules of natural or mathematical systems have been used to generate the totality of the musical work, including several musifications of CA. By accepting some validity in the aforementioned thesis and allowing user intervention to the CA process via graphical input and access to the parameters of the musification algorithm, the authors' intention was not to develop a generative system that is designed and then merely set in motion, but to use the generative process as a device for low level control. Higher order control resides with the user and functions as a device to generate local structures to be exploited / developed by user input. This approach borrows from the notion of supervisory control defined by Sheridan as the situation where "one or more human operators are intermittently programming and continually receiving information from a computer that itself closes an autonomous control loop through artificial effectors to the controlled process or task environment" (Sheridan 1992). Sheridan has also identified a significant relationship between supervisory control and music making in terms of performance, composition or both (Sheridan 2004).

In the mapping of CA data to music parameters, discussed in detail below, the authors purposefully avoid fixed translations of cell position to interval distance (Ariza 2007) in an effort to circumvent obviously formulaic musical results that typify the lattice structure of the two-dimensional CA. Livecell uses probabilistic mapping processes similar to those discussed in Brown's work on "Appearances" to generate note durations from CA, however it appears that Brown abandons this in the final version of the work "as the patterns of pitch and rhythm that result from simple mappings of CA were found to be too uneven for performers to comfortably sight-read." (Brown 2005).

Data musification

Cellular automata

Cellular automata have been widely employed as a tool for sound synthesis, composition and performance since the 1980s, as documented by Burraston and others (Burraston / Edmonds 2005). Approaches to employing CA in composition (rather than synthesis) tend to focus on mapping the output values from the CA to pitch and duration (Burraston et al. 2004) with fixed translations of cell position to interval distance (Ariza 2007) and/or note duration. In these systems there is often limited scope for manipulating rules and cells in real-time. In contrast, this application makes use of a combination of statistical and probabilistic mapping techniques and real-time user input.

Cellular Automaton	Rule	Behaviour
Game of Life	B3/S23	Complex evolving patterns, oscillators, spaceships
Seeds	B2/S	Generally rapid and chaotic growth, but some small spaceships and stable patterns
Move	B368/S245	Contracts to low density stability
Coral	B3/S45678	Slowly growing solid patterns
Stains	B3678/S23567	Solid stable patterns

Table 1. CA Rule Sets.

Livecell employs a two dimensional CA system operating in a finite but unbounded space in which cells are wrapped around the vertical and horizontal edges. In common with other life-like CA, each cell has two states (on or off) with a neighbourhood consisting of the eight adjacent cells to the one under consideration. The system operates using one of five possible rule sets selected to produce a range of emergent behaviours that tend towards stability, expansion or contraction [Table 1]. The active rule set can be changed by the user at any point during performance, which combined with the ability to add and remove cells via the user interface enables automata bending to take place (Ariza 2007). The system is iterated at 100ms intervals, when the cell values are updated and the graphical representation redrawn.

Score mapping

The CA grid is notionally divided into four quadrants with activity in the top left quadrant mapped to the first violin, the top right to the second violin, the bottom left to the viola and the bottom right to the cello. The state of the CA grid in each of these quadrants is sampled independently at an interval relative to the density of the system in that quadrant at the previous sampling point.

For a given quadrant at each sampling point the average density of live cells is calculated. The resulting density value is categorised as being low, medium or high, and the result used to determine the next sampling interval for that quadrant. High density corresponds to a two second sampling interval, medium density to a three second interval and low density to a four second interval.

Cell Density Range (%)	No. Events	Available Note/Rest Durations		
0 - 10	1	𝅝		
10 - 19	2	𝅝	𝅗𝅥	𝄼
20 - 29	3	𝅝	𝅗𝅥.	𝅘𝅥
30 - 39	4	𝅗𝅥	𝅘𝅥	𝄽
40 - 49	5	𝅗𝅥	𝅘𝅥.	𝅘𝅥𝅮
50 - 59	6	𝅘𝅥	𝅘𝅥𝅮	𝄾
60 - 69	7	𝅘𝅥	𝅘𝅥𝅮.	𝅘𝅥𝅯
70 - 79	8	𝅘𝅥𝅮	𝅘𝅥𝅯	𝄿
80 - 89	9	𝅘𝅥𝅮	𝅘𝅥𝅯	
90 - 100	10	𝅘𝅥𝅯		

Table 2. Score mapping

The average density of live cells is also used to determine the number and duration of musical events (notes or rests) output by the system at each sampling point. The number of musical events is calculated by dividing the percentage of active cells in the quadrant by ten and rounding up to the nearest integer. Durations are selected at random from a pool of values specified for each density range with higher densities corresponding to shorter values and vice versa [Table 2].

To determine the note pitches, each quadrant is divided into a number of vertical slices corresponding to the number of musical events to be generated. Each vertical slice is then analysed to calculate the average vertical position of cell activity within it. This value is then used as an index to a lookup table containing the pitch set class selected in the graphical user interface. Higher vertical activity in a slice corresponds to a higher pitch value in the output and vice versa.

The average vertical position of cell activity (AVPA) is determined according to the following formula, where n is the vertical height of the slice in pixels, m is the total number of active cells in the slice and the cell state is the number of active cells on row i:

$$(1)\quad AVPA = \left(\sum_{i=1}^{n} i \times cell\ state\right) / m$$

The output of this calculation is scaled and used to retrieve the corresponding pitch from the lookup table. The resulting pitches are then assigned to the previously determined note durations to produce the complete musical phrase.

The result of this process is that a quadrant with low activity (<10% of cells active) will produce a single note, the pitch of which will depend on the average vertical position of cell activity in the whole quadrant. A quadrant with a high number of active cells (>90%) will result in a phrase containing ten notes, the pitches of which are determined by the average vertical position of cell activity in each of ten equally spaced vertical slices of the quadrant.

The dynamic markings for each instrument are determined by calculating the amount of change in cell activity between the current and the previous sampling points. Little or no change in the number of live cells in a quadrant results in a quieter dynamic marking for that instrument and vice versa.

Expression markings for each instrument are randomly generated from one of three pools, corresponding to the result of the cell density calculation (low, medium or high). Available expressions include legato, pizzicato, staccato, tremolo and marcato. Expression markings that produce shorter envelopes appear with greater frequency at high cell density and therefore shorter note values.

This mapping process results in a complete bar for each instrument, which appears in the score for immediate realisation by the performers.

Audio mapping

The system is also using the cellular automata data to trigger and treat a set of predetermined sound files which are presented alongside the string quartet. These sound files are comprised of two distinct groups according to their spectromorphological characteristics; group A - granular / glitch, group B - resonant / textures. The cellular automata data is used to determine the sound type to be used, the number of simultaneous sounds, as well the playback speed (pitch) and its variation over time (varispeed). The start and end points within each sound file are randomly selected

In contrast to the quadripartite structure of the score mapping, the audio mapping process uses data from the entirety of the CA window. The state of the CA grid is sampled at an interval relative to the density of the system at the previous sampling point. At CA density between 1% and 10% there is an equal probability (25%) of the following four outcomes:

- No sound file playback
- Single sound file from group A at stable playback speed (fixed pitch)
- Single sound file from group B at decreasing playback speed (Descending glissando)
- Both of the above simultaneously

Similar probabilistic behaviours are also used to determine the triggering and treatment of sound files for the remaining density ranges as per the table below.

CA Density %	No SF	Group A (SF 1-4)	Group B (SF 5-10)	A+B
1-10	25%	25% 1 x (Fixed Pitch)	25% 1x (DG)	25%
10-20	25%	25% 1 x (RPT)	25% 1x (DG)	25%
20-30	25%	25% 1-4 x (DG)	25% 1-2x (DG)	25%
30-40	25%	25% 1-4 x (AG)	25% 1-2x (DG)	25%
40-50	33%	0%	33% 1-3 x (RPT) 33% 1-3 x (RPT)	0%
50-60	33%	0%	33% 1-3 x (DG) 33% 1-3 x (DG)	0%
60-70	25%	0%	25% 1-3 x SF8 (RPT) 25% 1-3 x SF9 (RPT) 25% both	0%
70-100	25%	0%	25% 1-3 x SF8 (DG) 25% 1-3 x SF9 (DG) 25% both	0%

Table 3. Audio mapping (SF= Sound file, RPT= Random Pitch Trajectory, DG= Descending Glissando, AG=Ascending Glissando)

The aim of the audio mapping strategy is to generate another layer of musical discourse that displays complexity and temporal development analogous to the graphical realisation of the CA and the score. Increased cell activity corresponds to increased sound file activity and specific rhythmic activity in the score favours specific sound files.

The score

Livecell uses the MaxScore object for Max/MSP to produce standard western music notation in real-time for each instrument of the string quartet (Didkovsky / Hajdu 2008), including clef, stave, notes, dynamics and expression markings. The phrase produced at each step of the mapping process is placed within a single bar on the left hand side of the stave. As new phrases appear, preceding ones shift to the right allowing performers to complete the passage if they have not yet done so. Conventional time signatures and tempo markings are replaced by a time value in seconds, which indicates the time available for the performers to play through the notes in the bar. Note values are therefore relative only to others within each bar as there is no universal meter or tempo. A progression bar above each stave moves through the bar in one second increments to assist performance. Performers are not required to be exact in their timing or indeed likely to be playing the same tempi at the same time.

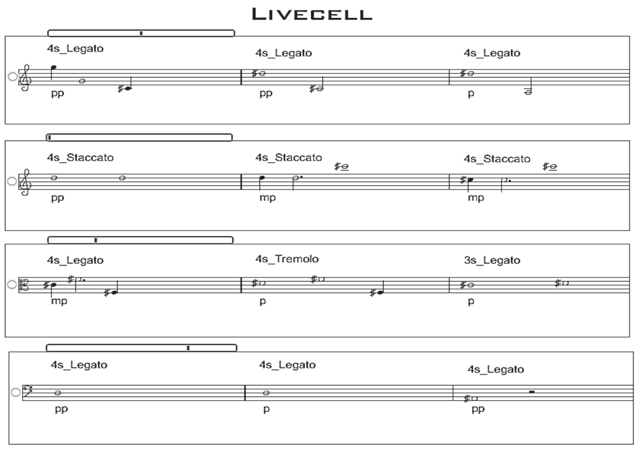

Figure 2. Score window

The graphical user interface

The graphical user interface of the application [Figure 3] consists of a two dimensional CA grid and four sets of control functions in the right and left hand side columns.

The grid displaying the state of the CA allows users to create or destroy cells using a mouse pointer or other controllers, which enables supervision over the shape and direction of the CA evolution. Users can determine the function of the pointer (create or destroy) and clear the grid by selecting from the first set of control functions on the right hand side.

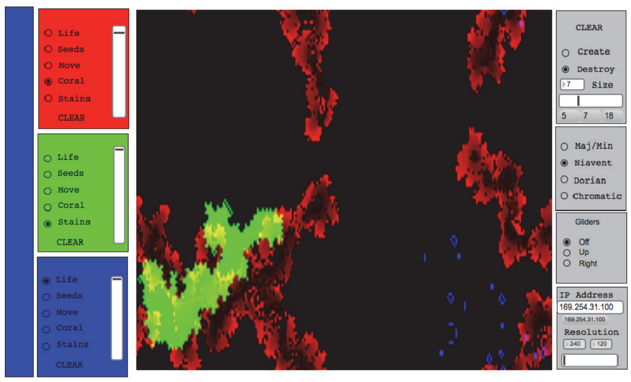

Figure 3. The graphical user interface window

The next set of controls on that column allows users to determine the pitch set class to be used. The four pitch set classes available in this implementation of the system are derivative of scales and modes and named after those. The choice of scales and mode names does not imply the use of tonal or modal functions (harmonic or melodic), as the system has not been designed to gravitate towards tonal centres.

The set of controls on the left hand side allows users to determine the CA rule set, allowing up to three different rule sets to be run simultaneously. Cells in each active rule set are visualized using three different colours as can be seen in figure 3. The colour bar on the left signifies the input active rule set at any point. Furthermore, the life span of active cells can be determined through the use of the slider adjacent to each rule set selector.

The last set of controls allow users to set the resolution of the CA grid, configure the MIDI output of the system, enter an IP address for use of the system over a network and control the level of the sound file output. Although Livecell has been developed for live performance, MIDI realisation is accommodated for aural monitoring and presentation in different contexts. The IP address configures the system for performance over split locations, with the possibility of user input taking place in a geographical location remote from the performance space.

Discussion of results

Users have responded positively to the system, regardless of their musical background and training, reporting a meaningful and engaging interactive experience. This is apparently the result of the identification of patterns in the connections between their input and the system's output. O'Sullivan outlines the importance of this in physical computing projects in the arts, making specific reference to issues of laptop music performance (O'Sullivan / Igoe 2004). Following on from this, it is apparent that the CA rule sets in combination with the musification algorithm produce a range of musical schemata and transitions between these that can be replicated on demand. Once those patterns have been identified, the user can employ them to develop more complex, longer musical structures.

It has been observed that users with a greater understanding of musical structure can sustain engagement with the system for longer periods of time and are more likely to generate a meaningful musical discourse – at least on the poietic level (Nattiez 1991). Musicians working with the system reported a more rewarding experience when the user was a composer, exemplifying a supervisory control system rather than a purely generative one. It has become evident that Livecell could work both in more formal conditions that are associated with the composition and performance of a complete work and as a music installation where users are encouraged to interact with the system in a more informal way. Evidently the latter would call for a MIDI realisation of the score or musicians equipped with Jovian patience.

Future work

While Livecell has been presented in various events and has been awarded several honours, it remains a system in development. Future work will focus on three areas:

1. Development of the graphical user interface to institute further refined user control over the CA and musification processes with view to expand the capacity for more diverse musical results. Enhancements will include independent control of rule sets and pitch class sets for each instrument as well as additional pitch class sets and the capacity for these to be defined by the user.
2. Development of the audio mapping processes to institute more refined control over the sound file playback in an effort to rationalize further its behaviour relative to the instrumental performance.
3. Development of an educational system for teaching creative music practice in primary education. The architecture of Livecell would allow students from diverse cultural backgrounds, and therefore with a diverse range of approaches to music making, to access composition tasks without prior knowledge of music notation. Input via a graphical interface and the use of a self-organising process would enable students to construct complex musical structures using visual analogies. Furthermore, the system's ability to translate students' interaction with the graphical interface to a score would allow them to engage with western notation and draw parallels between the latter and their own practice.

References

Ariza, C. (2007). Automata Bending: Applications of Dynamic Mutation and Dynamic Rules in Modular One-Dimensional Cellular Automata. *Computer Music Journal* 31/1: 29-49.

Bilotta, E. / Pantano, P. (2002). Synthetic Harmonies: An Approach to Musical Semiosis by Means of Cellular Automata. *Leonardo Journal* 35/2: 153-159.

Brophy, S. P. / Schwartz, D. L. (1998). Interactive Analogies. *Proceedings of the International Conference of the Learning Sciences*. 56-62, AACE.

Brown, A. R. (2005). Generative Music in Live Performance. *Proceedings of the Australasian Computer Music Conference, Brisbane*.

Burraston, D. / Edmonds, E. (2005). Cellular Automata in Generative Electronic Music and sonic Art: Historical and Technical Review. *Digital Creativity* 16/3: 165-185.

Burraston, D. et al. (2004). Cellular Automata in MIDI-Based Computer Music. *Proceedings of the International Computer Music Conference, San Francisco, California*.

Didkovsky, N. / Hajdu, G. (2008). MaxScore, standard music notation in Max/MSP, *Proceedings of the International Computer Music Conference*, Belfast.

Miranda, E. / Biles, J. [eds.] (2007). *Evolutionary Computer Music*. London: Springer.

Nattiez, J. (1991). *Music and Discourse: Towards a Semiology of Music*. Princeton: Princeton University Press.

O'Sullivan, D. / Igoe, T. (2004). *Physical Computing: Sensing and Controlling the Physical World with Computers*. Boston: Thomson Course Technology PTR.

Sheridan, T. B. (1992). *Telerobotics, Automation, and Human Supervisory Control*. Cambridge, MA: MIT Press.

____________. (2004). Musings on Music Making and Listening: Supervisory Control and Virtual Reality. *Proceedings of the IEEE* 92/4: 601-605.

[Abstract in Korean | 국문 요약]

셀룰러 오토마타cellular automata의 음악화에 필요한 확률론적 절차: 라이브셀Livecell 프로젝트의 사례 연구

킹슬리 애시 / 니코스 스타브로폴로스

이 글은 실시간으로 악보를 만들며 소리 파일들을 구동하고 처리함으로써 '상호 작용을 통해 음을 생산하는 작곡interactive generative composition'을 위한 시스템인 '라이브셀Livecell'에 대하여 논하고 있다. 이를 위해 저자의 이론적 근거를 제시하고, 맥락에 따른 셀룰러 오토마타cellular automata의 적용을 예시하며, 응용 프로그램의 구조 설계를 밝힌다. 또한 셀룰러 오토마타의 음악화에 대한 접근 방법과 악보에서 데이터 간의 관계를 제시하며, 악보의 제작과 그 디스플레이를 비롯한 인터페이스 디자인에 대해 기술한 후, 음악적 최종 생산물에 대한 관찰과 향후 발전에 대한 논의로 끝을 맺는다.