Exploring the Self Through Algorithmic Composition

Roger Alsop

ABSTRACT

The author discusses his views on musical composition in the late twentieth century, focusing on the influence that communication and computer technology have had over his pursuit. He goes on to describe his use of computer-based algorithmic composition and how this particular approach enhances and refines his understanding of his own musical self-expression. He describes four computer algorithms, used in recent compositions and improvisations, that reflect his particular musical interests.

Examining reasons for choosing these particular processes, the author raises questions he believes are both personal and universal in the evolution of a composer in these times.

Developing a Process for Musical Composition

Musical composition is a path of continual self-exploration. While travelling this path, I continually make new discoveries about myself as an artist and my relationship to music. I find the challenges and discomforts encountered along this path must be overcome or reconciled in order to integrate myself with the music I create.

The first act of musical composition occurs when the composer acts musically in a way that draws a reaction within his or her self. This may be when a 4-year-old child realizes the joy he or she feels when banging on a piano keyboard. This young composer may then repeat that action, believing or hoping that a similar reaction may occur again. From here, an interest in creating music develops and the musical exploration begins.

As composers become more skillful they may begin to understand the emotional impact of their work. A more seasoned composer than our 4-year-old may feel his or her heartbeat slow down when writing a descending, slowing melody, or feel it speed up as he or she writes an ascending, quickening melody. Using these observations, composers can manipulate sound for their own ends.

Now the questions leap forward: Why did I compose this piece? Why did that work? Does this piece fit with my image of myself as a composer? Does this represent me in the universal context? Will it communicate effectively to my audience? These rather broad questions soon lead to more prosaic matters: Do I subscribe to the tension/release paradigm? Is Musical Instrument Digital Interface (MIDI) an appropriate system for me to use? How tightly should I adhere to a serial/tonal/aleatoric/modal/etc. system? Is this too difficult/easy/boring for a performer to play?

The pallet of processes, styles and sounds available to a composer is broader and more colorful than ever before. Global communication has made this so. Now all the world's cultures and their musical traditions are available to influence and educate a composer.

Conversely, today's recording and reproduction technologies reduce the control a composer has over his or her music, resulting in the almost total removal of a composition from the composer and firmly placing the interpretation of the musical expression on the listener. This transference of interpretation makes the uses of the composition more diverse. Now a published composition is available for exploitation by anyone with the appropriate technology.

The increasing availability, variety and quality of MIDI devices give the freedom to audition and mix sounds from everywhere. It is now possible to compose a piece with the sounds of gamelan, pipe organ, *oud* and *shakuhachi,* maintain the styles traditionally ascribed to these instruments, and fit the piece within a genre. This genre is usually prescribed by someone else in an attempt to describe the music and to place it in a context familiar to him- or herself.

Knowing who one's audience is can be problematic as well. A composer in Delhi who believes that a work fits minimalist traditions may find it is a dance floor favorite in New York. A disregarded Melbourne composer may be a hit in Amsterdam, for reasons the composer either does not understand or to which he or she does not even subscribe.

While it is possible to compose with an excessively broad range of instrumental sounds immediately available through MIDI, it is also possible to compose for entirely new, composer-created instruments requiring entirely new compositional processes. One can compose and audition a piece, print a score, or produce a CD in one's room, taking absolute and sole responsibility without the input—subversive or supporting—of others. It is also possible to compose via delegation of responsibility, either through other humans or ever-changing machines. Collation is a significant compositional process.

Within this vast and confounding milieu, a composer tries to find a voice, to develop a position and a process that expresses both individuality and common humanity.

What Is a Composition Algorithm?

Christopher Strachey defines an algorithm as "a procedure for performing a complicated operation by carrying out a precisely determined sequence of simpler ones . . . [algorithms] exclude all personal judgment" [1].

Roger Alsop (composer/sound designer), 14 Perry St., Fairfield, 3078 Victoria, Australia. E-mail: <rogeralsop@hotmail.com>.

This statement describes musical composition algorithms as I construct and use them. Musical composition is a complicated operation resulting from the simpler operation of setting the parameters of the algorithm. The exclusion of personal judgment gives me both the purity of my idea and novelty in the result. When building a composition algorithm I create a world within which to express myself.

ALGORITHMIC PROCESSES

Algorithmic processes are standard tools for a composer. Guido of Arezzo's method of deriving plainchant from text by ascribing pitches to vowel sounds [2] is an eleventh-century use of algorithmic processes.

The Western traditions of tonalism and serialism are based on algorithms that vary in their degrees of mechanization. The evolving mechanization of the twentieth century spawned the increasingly mechanistic approaches of serialism from the Romantic era, which had reached the extremes of tonal chromaticism, as can be seen in Wagner's *Tristan und Isolde* (1857–1859) [3]. This evolution to serialism can also be seen in the development from Schoenberg's Romantic/serial *Verklärte Nacht* (1899) [4] through Webern's Symphonie, Op. 21 [5] to Babbitt's use of serial techniques in all aspects of *Three Compositions for Piano* (1948) [6] and finally to Boulez's *Le Marteau sans mâitre* (1957), where that mechanization was relaxed in order to accommodate expression [7]. These composers used highly developed algorithms—tonalism and serialism—to create these works.

An early example of a computer-based algorithmic composition is Lejaren Hiller's *Algorithms 1,* (1968) for orchestral instruments and tape, which uses stochastic, serial, and then constraining algorithms for each of its three movements [8]. Some recent examples are listed in Robert Rowe's *Interactive Music Systems,* including Jean-Claude Risset's *Duet for One Pianist,* in which a computer algorithm supplies a second set of hands that respond to the performer's input [9].

WHY USE ALGORITHMIC PROCESSES?

An algorithmic system relieves the composer of many decisions. The tonal algorithm usually defines the start and end points of musical gestures and the variety of methods utilized in order to move between those points. In musical history, this variety increased over time until a far simpler, but more confining algorithm, serialism, was introduced. Within the confines of this simplicity, both immense freedom and diversity of result exist.

Freedom and diversity in expression is a requisite of artistic endeavor, but boundless freedom can be confounding. By creating or subjecting oneself to a set of rules, or a set of processes such as an algorithm, one hopes to confine one's actions to an area of truer self-expression. My approach is to build algorithms as agents that assist me in creating music. By using an algorithm in this way, the composer can make worlds of complex interrelations, generating cascading actions that trigger other foreseen and unforeseen reactions. It is possible, due to the increasing power, elegance and availability of computer hardware and software, to easily preview the results of these interactions. This ease allows me to examine my motives and actions and their results while working, without the preciousness of prolonged, painstaking and single-minded efforts required by more traditional compositional methods.

As each action of a composition algorithm has foreseen and unforeseen reactions, one tends to want to be accountable for these reactions. However, this sense of accountability is tempered as the process becomes more practiced and intuitive, much like the sense of accountability for each sound a novice violinist makes is reduced as the playing becomes more practiced and intuitive. At this point of expertise, the building of a computer algorithm becomes equivalent to making an instrument, learning how to play it and creating a composition all at the same time. Therefore the process and its result reflect the composer's relationship to all three of these activities.

Thus, the algorithm becomes a transparent, systematic and detached path of self-exploration. The above questions are continuous and are continually answered. The beauty of this approach is that the answers often come without the questions being articulated—or if they are articulated, they seem of little importance at the time of inspiration. One concentrates effort on building the world in which the music can exist and hopefully flourish. With the completion of a composition comes an understanding of the processes used.

This understanding comes about in two ways: firstly, through the resultant music, when the composer recognizes a piece as uniquely his or her own. On listening to the piece, one discovers things about it that are novel, challenging, familiar and comfortable. By asking what results fit these categories and why, the composer increases his or her understanding of their personal relationship to music.

Secondly, the algorithm that has generated the music is a map of the composer's processes in creating the world from which the piece has come. Within this map one can discover the processes and pathways one favors in creating and organizing sound.

MY ALGORITHMIC PROCESSES

Over the past few years my efforts have been in the electro-acoustic domain. A prime aim in my musical life is to draw connections between music and other art forms. The more physical and less interpretive these connections are, the better. The process of creating computer algorithms aids this aim through the mechanistic and transparent nature of the computer. I use the Opcode version of the Max programming environment for MIDI [10] and IRCAM's Signal Processing Workstation (ISPW) version [11] for audio processing, since I find that Max allows much creative freedom and is relatively easy to learn.

I use two approaches in creating music. In the first, the algorithm and the composition evolve together through continual audition and refinement. In the second, I improvise with an algorithm specifically designed to explore a particular aspect of improvisation. In the latter case, I build the algorithm before any music is created. Two examples of each approach, and the corresponding algorithms I use, are briefly described below [12].

EVOLUTIONARY APPROACHES TO BUILDING A COMPOSITION ALGORITHM

In my recent compositions, as opposed to my improvisations, I used poems as seeds from which to compose sets of pieces. I developed the algorithms in such a way as to create an intricate web of interactions affecting all aspects of the algorithm, resulting in self-referential compositions of complex internal integrity.

Two algorithms in which I used this evolutionary approach in their construction are *Someone,* using the ISPW version

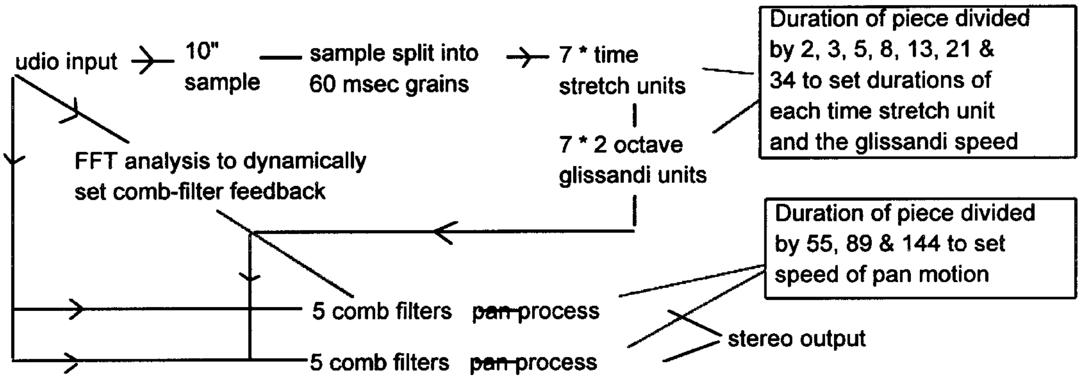

Fig. 1. ***Someone*** **algorithm, 1998. Signal and information paths of the algorithm. An unbroken line denotes the signal path, a broken line denotes the information path.**

of Max, and *Compose/Canon,* using Opcode's version of Max. Both algorithms were built with particular poems in mind but are effective with other sound sources.

Someone essentially follows two paths. One path takes a sound file, samples it and splits the sample into grains. This path allows for many transformations of the sound file, including time-stretching and contracting, glissandi and pitch shifting. The other path creates harmonies of the input using 10 comb filters. I use a Fast Fourier Transform (FFT) analysis of the sound file to dynamically vary the feedback of the comb filters in order to create the harmonies. The sound file that is being transformed dictates the effect of these transformations.

Figure 1 shows the process for the transformation of an audio input, in this case a poetry reading, into a musical composition.

Compose/Canon is made up of two subalgorithms. *Compose* takes a string of MIDI note-events, comprising a specific pitch, MIDI velocity, duration and inter-note offset; for example, pitch may be MIDI note 72, the MIDI velocity may be 63, the duration may be 64 time units and the inter-note offset may be 79 time units. *Compose* then applies simple arithmetical transformations to each of these attributes. The transformational process is: (1) multiply the input by an integer (×), (2) constrain the result within a modulus (%), (3) then transpose the result (+).

For example: if note 63 is multiplied by 7 (63 × 7 = 441), then constrained within a modulus of 24 (441 % 24 = 9) and then transposed by 56 (9 + 56 = 65), the resulting MIDI note number will be 65, which is MIDI note F3.

The resulting note-event string is fed through *Canon,* a harmonizing algorithm based on traditional approaches to harmonizing a melody.

In recent compositions, *Compose/Canon* uses the phoneme string of a poem that has been translated into a string of MIDI note-events. It is possible to use any other string of MIDI note-events as a trigger for *Compose/Canon.* Figure 2 shows the process of taking any note-event string, transforming it and providing it with an accompaniment.

Building an Algorithm for Improvisation

Two algorithms for improvisation that I have built within Opcode's version of Max are *Guitar 21* and *Selectnotes. Guitar 21* uses the first eight notes of each of the six strings of a MIDI guitar to trigger 48 different sub-algorithms that either affect different MIDI parameters or generate different notes. These sub-algorithms remain in continual motion, looping between points set by the musician striking any of those 48 notes.

The process works with the limitations of human computation, memory and physical control. Because there are so many continuously moving variables, it is difficult for the improviser to accurately predict how each sub-algorithm is affecting its assigned MIDI parameter. This means that the improviser must adjust his or her approach to the instrument being played with no preconceptions of the instrument's reaction to the subalgorithm.

Figure 3 gives an example of the composition ideas used in the algorithm. To fully describe all aspects of the algorithm here is unnecessary; suffice it to say that all musical aspects—pitch, amplitude, duration, inter-offset time, glissandi, timbre, articulation, etc.—are generated through similar processes.

Selectnotes assumes that improvisation between musicians is essentially reactive. It takes an input of MIDI notes, stores them, adjusts them with a number of possible operations and then selects certain sections of the input data to put out. These adjustments include:

- the transformations: retrograde, inversion, transposition
- selection of certain parts of the input for replay
- selection of similar, opposing or parallel reactions to the input (for example, if the input is rhythmically dense, then the output from *Selectnotes* may be rhythmically sparse)
- selection of part of the input to loop, making an ostinato
- displacement of rhythms and amplitudes against pitches to produce new melodies.

All these adjustments interact to form more complex and less predictable results, which can be sent to any number and variety of MIDI instruments, creating an ensemble of any size. Figure 4 shows the process of storing a performance, adjusting and interpreting it, and then playing a new version.

Algorithmic Processes and a Worldview

Every composer and every piece of music has a point of view that continually guides the process of composition, either consciously or unconsciously. This point of view is based on both the composer's musical or artistic beliefs and on his or her broader social or spiritual beliefs.

Algorithms Based on External Influences

I chose two poems to trigger *Someone* and *Compose/Canon.* "Saint Dymphna's Bells," the poem used with *Someone,* was written and read by Barry Dickins [13]. The poem deals with the execution of Ronald Ryan, the last person to be executed in my home state of Victoria, Australia. "Zooming In" [14], which I used in the algorithm *Compose/Canon* and which was written by Alex Skovron, describes a journey from the exterior to the interior of a human being.

While I was building the algorithms *Someone* and *Compose/Canon,* I was not consciously aware of any philosophy shaping my decisions. Later, it seemed to me that they embodied my view that personal, social and spiritual beliefs act with hegemonic control over a person's actions. These beliefs are often either unknown or unarticulated, and usually can be distilled to a single paradigm. Just as a person's spiritual beliefs in some way influence his or her actions, the poem influenced the operation of the algorithm.

Both poems involve the human condition, describing it from two opposing positions. "Saint Dymphna's Bells" is a passionate poem, passionately read. It discusses the role of the state in retribution and the resultant effects on society, and the role of the family in developing the character and subsequent actions of a person. Both of these agents have an irrevocable effect on the individual, the family and the state, leading in this case to the deaths of two men. Structurally, the poem meanders through a variety of times and positions, creating a web that describes the events leading up to the execution and immediately after it.

"Zooming In" describes a camera-like journey of surgical purpose and direction into the physical, mental and spiritual aspects of a person. The poem inserts itself through the exterior layers, revealing the physical, atomic interior, then exits in order to re-enter and expose the mental and spiritual core. The poem follows this process without distraction; it is dispassionate and detached, not commenting on what it describes.

These aspects of the poems are reflected in the algorithms. First the algorithm dissects the poem into its phonemes. As each phoneme of "Zooming In" is encountered, *Compose* triggers a specific note-event, creating a melody unique to the poem. This melody has a similar role to the poem's text; it is a surgical line that then becomes clothed in the accompanying harmony attached to

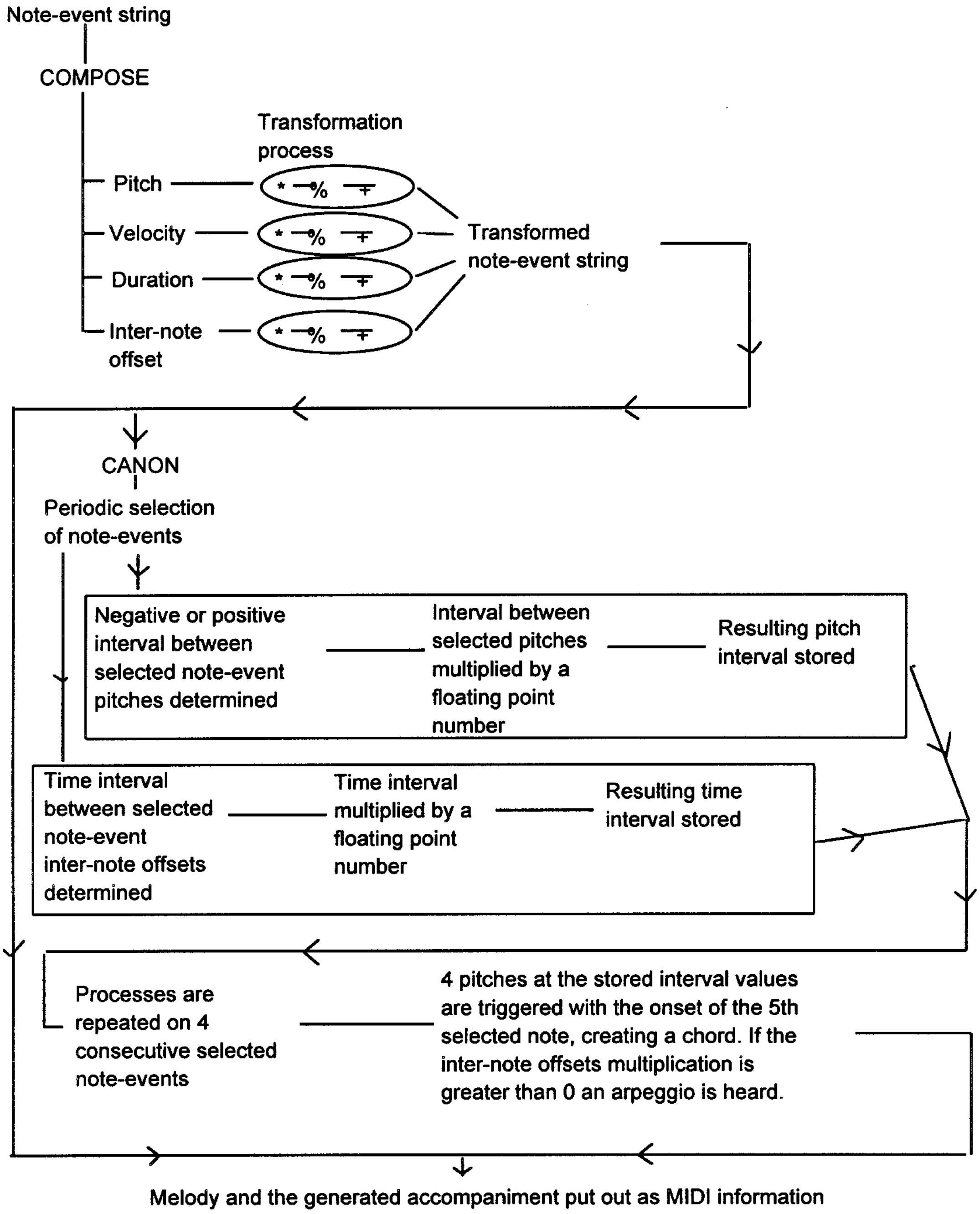
Note-event string
COMPOSE
Transformation process
Pitch
Velocity
Duration
Inter-note offset
* –% ∓
* –% ∓
* –% ∓
* –% ∓
Transformed note-event string
CANON
Periodic selection of note-events
Negative or positive interval between selected note-event pitches determined
Interval between selected pitches multiplied by a floating point number
Resulting pitch interval stored
Time interval between selected note-event inter-note offsets determined
Time interval multiplied by a floating point number
Resulting time interval stored
Processes are repeated on 4 consecutive selected note-events
4 pitches at the stored interval values are triggered with the onset of the 5th selected note, creating a chord. If the inter-note offsets multiplication is greater than 0 an arpeggio is heard.
Melody and the generated accompaniment put out as MIDI information

Fig. 2. ***Compose/ Canon*** **algorithm, 1998. Note-event path of the algorithm.**

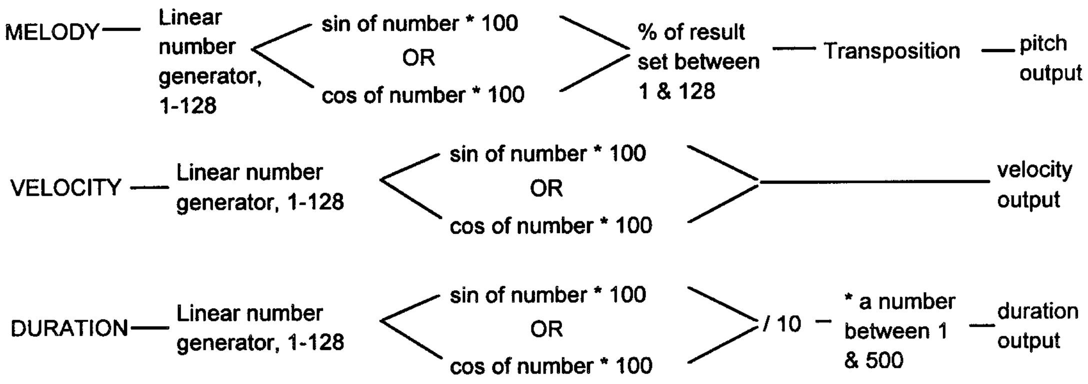

Fig. 3. ***Guitar 21*** **algorithm, 1998. Example of the information path.**

it by the *Canon* part of the algorithm. In this way, the resulting composition becomes the antithesis of the poem, clothing a quintessential core in a context.

Someone has a far more intricate set of interactions, reflecting the web-like interactions of the poem. Many aspects of the algorithm affect others in a looping fashion, much as the events of Ronald Ryan's life were linked together and amplified over time, eventually leading to his execution.

Algorithms Creating Improvisational Environments

The goal of *Guitar 21* and *Selectnotes* is to make the improviser approach his or her physical gestures in unfamiliar and challenging ways. Rather than starting from an exterior compositional influence, each of these algorithms create different contexts in which to improvise.

Guitar 21 removes the gesture-to-sound relationship of traditional instruments. Instead, the improviser creates a musical context and reacts to it with non-musical gestures. For example, holding the C# note on the B-string of a guitar for half a second may result in a pitch bend of a tritone being applied to all notes on a specified MIDI channel; holding the D# on the same string may change the time period over which the pitch bend occurs. In this way, the traditional gesture-to-sound relationship of the guitar is grossly distorted.

In this situation, the improviser cannot rely on any learned actions, as there is almost no relationship between the improviser's action and the musical reactions generated. Instead, each action results in many possible reactions, all of which have no expected relationship to the actions of the improviser. Here, two or three gestures may generate a whole composition, forcing the improviser to acknowledge the criticality of the minutae of each gesture. Subsequent gestures radically and irrevocably alter the path of the composition, creating new musical environments in which to react.

Most of the triggers in *Guitar 21* are time-based, set at increments of 250 milliseconds; this means that holding a note that effects a modulus for 1.5 seconds creates a very different musical reaction than one holding the same note for 1.25 seconds.

This algorithm exemplifies the view that each action, no matter how apparently insignificant, can have a fundamental impact on the doer by creating unforeseen but significant outcomes. It is not possible to have utter control over the results of each action, because it is not possible to fully control them or their interpretation. However, one aims for an improvisation that is a guided, if not absolutely meticulous, self-expression.

My purpose in designing *Selectnotes* was to provide an algorithm that would re-interpret the traditional musical gestures of the performer. Each note played is stored, to be adjusted into an almost infinite number of possible related note events. Here, the input from the performer is distorted and played back during performance. For example: an eight-note phrase may be played back at three-quarters of the tempo for the first three notes, then at twice the tempo for the next five notes, which have also been reversed. Many times, as I have listened to the playback of *Selectnotes* compositions, I heard novel

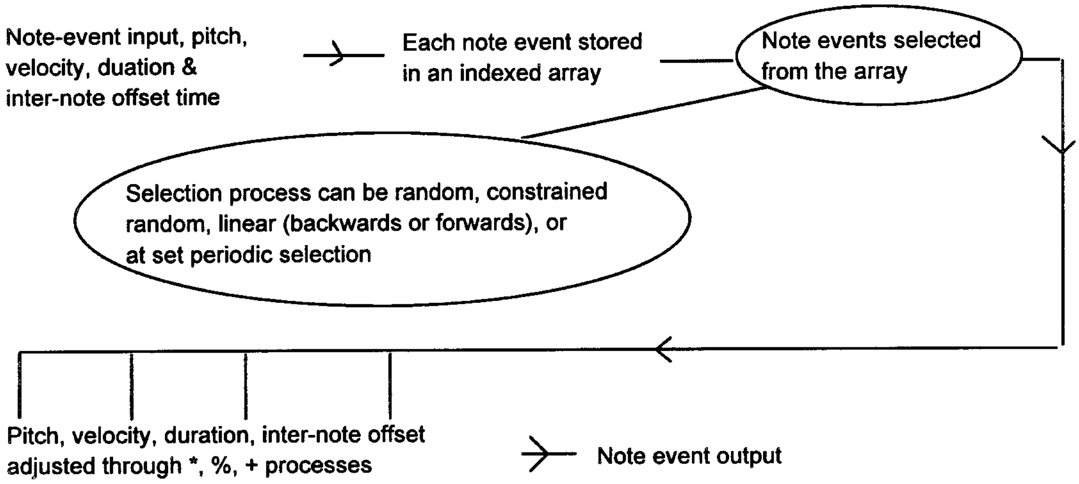
Note-event input, pitch, velocity, duation & inter-note offset time
Each note event stored in an indexed array
Note events selected from the array
Selection process can be random, constrained random, linear (backwards or forwards), or at set periodic selection
Pitch, velocity, duration, inter-note offset adjusted through *, %, + processes
Note event output

Fig. 4. ***Someone*** **algorithm, 1998. Information path of the algorithm.**

material that was based on the style of the original input.

This algorithm allows performers to listen to their own stylistic nuances as if filtered through another musician. These four algorithms are designed to force novel and challenging approaches to music-making: the first and second by using non-musical sources from which to generate music, the third and fourth by creating non-traditional environments in which to make music.

Relationships to the Results of the Algorithm

When I listen to the musical results of an algorithm, I find myself questioning how to view the different aspects of the piece, be they novel, challenging, familiar or comfortable. As I become more familiar with the piece and the purpose for listening to it changes, my views change. For example, something that succeeds as an ambient piece may fail in a concert situation.

Usually the composition fits my taste to some degree, as I have built the world in which it was made. If the musical surface does not fit my taste, but I believe that the generating idea is good, then I may discard or adjust the piece with very little extra effort.

I appreciate John Cage's notion that work should be "free of my own likes and dislikes, because I think music should be free of the feelings or ideas of the composer" [15]. I try to be very careful not to allow personal taste to overly dictate the workings of a piece or the selection of pieces for public display. While it is admirable to edit so that only the strongest pieces are aired, I feel it is important for a frank and less manufactured representation of the composer to also be available, for myself and for the wider audience.

The trap of over-protectiveness is especially dangerous when creating with algorithmic processes. There is an endless, almost overwhelming desire to adjust the algorithm in the hope that a better composition will result. This desire must be restrained in order for truer and more frank self-expression to take place.

Artistic self-representation is now so various and diverse, and the potential audience perhaps even more so, that anything goes; it just depends on choosing one's audience. As it is almost impossible to know the whereabouts, mood or desires of a potential audience, the most appropriate action is to disseminate as much music as widely as possible, without vetting too much according to mood or taste. To judge one's work too rigorously is not in keeping with the current age of mass and diverse communication, nor does it allow the full and proper self-disclosure or expression of the artist. Of course, this does not mean that an audience should be subjected to every result of every algorithm.

Conclusion

The act of musical composition is very different now than it was 15 years ago. The variety of styles that now exist, and the rapidity of musical changes, make complete originality an almost impossible goal. Similarities between pieces or derivations of other styles can be seen in almost all music without too much difficulty. This is due to the instantaneous nature of communication and, more directly, the variety of ways in which music is composed today.

The act of musical composition can be silent, putting notes on paper. It can be audible, through the MIDI environment, or it can be completely spontaneous, developed through improvisation. It is normal for a musician to be capable, if not adept, in all of these aspects of music-making.

It is also possible for a musician to compose highly individual and personal work, compose music for an advertisement, teach music and play in a wedding band all in one week or, like Alexander Borodin, have a completely non-musical day job [16]. All these influences can become tools for the composer to use. It is up to the composer to choose the tools that best suit his or her expressive needs and the needs of whatever he or she is creating. These tools may be hardware or knowledge, and are too diverse and fluid to fully comprehend. If one is to create music for this age, one must incorporate and selfishly use this diversity and fluidity in order to best represent oneself.

References and Notes

1. A. Bullock, O. Stallybrass and S. Trombley, *The Fontana Dictionary of Modern Thought* (Glasgow: Fontana Press, 1988) p. 24.

2. D. Harraân. *Word-Tone Relations in Musical Thought: From Antiquity to the Seventeenth Century* (Newhausen-Stuttgart: American Institute of Musicology, 1986) pp. 74–77.

3. D. Grout, *A History of Western Music* (London: J. M. Dent & Sons, 1973) pp. 612–616.

4. B. Lerner, liner notes for A. Schoenberg, "Verklärte Nacht Op. 4," *Verklärte Nacht, Op. 4 Suite, Op. 29,* Ensemble InterContemporain, P. Boulez, conductor. CBS Records Masterworks CD MK 39566, 1985, p. 3.

5. A. Webern, "Symphonie, Op. 21," *Complete Orchestral Works,* London Symphony Orchestra, P. Boulez, conductor. Sony Music CD SM3K45845, 1991.

6. Grout [3] p. 716.

7. Grout [3] p. 717.

8. L. Hiller, "Algorithms I, Versions I and IV," on *Avant garde: Evenings for New Music,* Deutsche Grammophon LP 2543 005 (no date of publication available).

9. R. Rowe, *Interactive Music Systems: Machine Listening and Composition* (Cambridge, MA: MIT Press, 1993) pp. 83–85.

10. Information regarding Max can be found at both the Opcode web site <http://www.opcode.com/products/max> and the web site for Max technical documentor Christopher Dobrian <http://www.arts.uci.edu/dobrian/max.htm>. For information regarding activities carried out using Max, the links on Cycling 74's web site <http://www.cycling74.com> provide a good range of what is being done and what is available through this platform.

11. The ISPW is based on the NeXT platform. Since the demise of NeXT, audio processing programs similar in style to the ISPW have been created. Cycling 74's version for the PowerMac, called MSP, works in tandem with Opcode's version. IRCAM's cross-platform version, called jMax, is only available on Silicon Graphics workstations and for PCs using the Linux operating system. <http://www.ircam.fr/jMax/>.

12. My compositions using these algorithms may be heard on *Frog Peak Collaborations Project,* Frog Peak Music CD FP 19 and at the MikroPolyphonie web site: <http://farben.latrobe.edu.au/mikropol/volume4/alsop-r/alsop.html>.

13. The sound file I used is an excerpt of a reading by Barry Dickins at the Hawthorn Town Hall in 1993.

14. A. Skovron, personal copy of poem supplied to the author, 1993.

15. R. Kostelantz, *Conversing with Cage* (New York: Limelight Press, 1987) p. 65.

16. D.M. Randel, *Harvard Concise Dictionary of Music* (Cambridge, MA: Belknap Press, Harvard Univ. Press, 1978) p. 59.

Glossary

comb filter—a simple feedback loop around a delay unit. In this case, the delay time is set to an audible frequency.

feedback—feeding the signal of a delay back into the delay unit; increasing the feedback increases the density of the delay output.

FFT—Fast Fourier Transform (FFT) analyzes a waveform for the purpose of reconstructing/synthesizing a similar waveform from a set of sinusoidal harmonics, each with a particular amplitude and phase.

grain—a very small audio sample; in this case each grain is 60 milliseconds long.

modulus—the remainder after an integer is divided by another integer. For example; if 23 is divided by 7 the remainder is 2 (3 ×7 = 21, 23 – 21 = 2).

sample—a digital recording of an audio signal that is usually stored in random access memory (RAM) and available for manipulation through computer technology.

sound file—a digital recording of an audio signal, usually stored on hard disk for playback through a computer.

Manuscript received 25 January 1999.