A Computer Music Retrospective

ERIC LYON

Sonic Arts Research Centre, Queen's University Belfast, Belfast, BT7 1NN, Northern Ireland
E-mail: e.lyon@qub.ac.uk

A composer who has worked in the field of computer music for the last 25 years reflects on how technological developments during that time have affected his work in computer music, instrumental music and hybrid combinations of the two. The compositional trajectories traced here often run parallel to opportunities afforded by the evolution of computer technology suitable for the generation of digital audio.

A NOTE ABOUT TERMINOLOGY

I use the term 'computer music' to denote any music for which computation is essential to its production. I use the term 'electroacoustic music' to describe music in which electricity is required for the synthesis or processing of the sound materials. Given the steep rise of the use of computers in music production over the time period described here, computer music and electroacoustic music now significantly overlap. However, the terms are not synonymous. An algorithmically computer-generated musical score is computer music but not electroacoustic music. A recording produced using exclusively analogue synthesisers is electroacoustic music but not computer music. My definition of electroacoustic music is similar to the definition given on the EARS website (Landy 2002).

1. INTRODUCTION

This is a composer's report, based on my efforts to coax music from computers during the last 25 years. I will discuss how my musical practice has evolved in response to dramatic developments of the means for creating music on computers, with a particular focus on the interaction between hardware, software and musical aesthetics. I will take an autobiographical approach, and, within the necessary constraints of my individual musical perspective, I hope to provide a useful historical perspective to other creative artists developing sound and music on computers, especially those who have more recently entered the field.

2. MUSIC ON A MAINFRAME

I composed my first computer music on an IBM 3081 mainframe at Princeton University in 1983. Princeton faculty members Paul Lansky, J. K. Randall and Milton Babbitt had all done pioneering work in computer-generated music, with either direct digital synthesis or, in Babbitt's case, with the computer-controlled Mark II RCA synthesiser. The music of these three composers dealt with compositional issues that were central for me: the precise realisation of complex pitch and rhythmic structures, new forms of musical expression and testing the boundaries of musical perception. A group of students at Princeton were pursuing computer music in a range of aesthetic modes from serialism to free-form timbral exploration, and they regularly presented this work at concerts on campus. Stimulated by this local computer-music culture and tempted by the musical possibilities afforded by total control over frequency, timing and dynamics, I enrolled in an upper division computer music class with Paul Lansky and got down to work.

3. COMPUTER MUSIC EXERCISE

I composed one piece using Lansky's computer music program Mix, which was under development in the early 1980s, and which later evolved into Cmix. My sound palette was simple and synthetic: a collection of fixed-spectrum waveforms specified by Fourier summation. The music was note-based and was sketched on manuscript paper before being transcribed to Mix. The specification process was similar to how one might today write a Csound orchestra and score. However, the conversion process involved a bit more than simply running the program on a modern laptop computer and immediately hearing the result. At Princeton, I typed my music into a terminal in a room filled with students from such disciplines as computer science, economics and philosophy. The computation of audio was sent as a batch job that was queued for service by the computer. Once the audio data had been generated to hard disk, the computer operator would be e-mailed a request to copy the data to magnetic tape. In the basement of the computer center, I would sign out and take possession of the magnetic tape. Finally, tape in hand, I would take a scenic 15-minute walk to the engineering quadrangle, which contained a lab equipped with DACs and

Organised Sound **13**(3): 209–216 © 2008 Cambridge University Press. Printed in the United Kingdom. doi: 10.1017/S1355771808000307

https://doi.org/10.1017/S1355771808000307 Published online by Cambridge University Press

speakers. The magnetic tape would be mounted to a drive, and the data played through the DACs, followed by the joy of hearing whatever my program had managed to produce. I eventually completed the piece, and was pleased by the novel experience of hearing my pitches and rhythms performed precisely as I wrote them. However, I immediately returned to composing instrumental music, and did not compose any more computer music at Princeton. Although I did not realise it at the time, my first computer piece was essentially an imitation of instrumental music, and it was simpler to write such music for instruments than for computer. Computer music was now on my radar, but I had not yet caught the bug.

4. LOCKED IN A ROOM WITH A COMPUTER

I continued my composition studies at the Eastman School of Music in 1984. The Eastman computer music studio consisted of an acoustically treated room containing a CRT terminal, speakers and a microphone. A PDP 11/34 minicomputer, disk drive and tape drive were isolated in a separate room, keeping the studio workspace nice and quiet. The computer file system contained a library of instrumental sounds recorded in-house that helped composers to get started in working with sound. I immediately enrolled in a computer music course with Allan Schindler. While reading the manual to Music 11, the precursor to Csound that was the main program used in Schindler's class, it struck me that sounds could be segregated into structurally useful groups by processing them with chains of DSP modules, essentially the idea behind modern plugins. This insight redirected my focus from pitch/rhythmic structures to timbre as a compositional determinant. As a result, I finally began composing in a manner idiomatic to the computer, working with sound as an object, rather than with abstractions of sound. I now came under the influence of the music and writings of Karlheinz Stockhausen, but surprisingly, not Pierre Schaeffer. As a musical formalist, I might have preferred the school of *Elektronische Musik* to that of *musique concrète*, but I encountered neither school in my computer music studies, and discovered Stockhausen's theories only by way of his music. Pierre Schaeffer's ideas and music were not considered seminal in the American educational institutions that I attended during the 1980s, and I did not learn about his work until much later. From Stockhausen's notion of formal schemes, however, I took the idea that micro and macro forms could be vastly disparate (unlike Schenker's self-similar tonal forms), and that an inventive formal scheme could elicit surprising behaviours from familiar musical materials. These concepts fed directly into my first serious computer music composition.

5. BABY COMPUTER MUSIC

I spent the summer of 1985 working on a computer piece called *Hot Shot*. As in my Princeton study, most of *Hot Shot* was composed on paper before being realised on computer. This was my last computer piece from this compositional period to maintain a strong connection to an instrumental musical conception. *Hot Shot* has two superimposed formal structures: (i) a multi-layered polyrhythmic texture in 12-tone equal temperament, played with an FM mallet instrument designed by Alan Schindler and modulated by very slow, large-scale amplitude curves; and (ii) a block structure articulated by processed recordings of a baby (my two-month-old sister) and chords in the five-tone equal-tempered scale played on a synthetic instrument of my own design. The baby sounds were processed with LPC, pitch shifting and hand editing, and scaled with rapid panning and amplitude curves. All panning, amplitude and pitch data was composed in advance of assembly on computer. There is a striking resemblance between the use of baby sounds in *Hot Shot*, and Stockhausen's use of the singing boy in *Gesang der Jünglinge*. This association was not intentional, and I only became aware of it much later. However, given the extent to which Stockhausen's formal concepts influenced *Hot Shot*, it is fitting that there was a subconscious sonic reference to Stockhausen's electroacoustic music as well. Sound example 1 is an excerpt from *Hot Shot* in which all constituent elements described above can be heard.

6. INFLUENCES ON INSTRUMENTAL COMPOSITION

My instrumental music soon became influenced by my work in computer music. The idea of separating form from content led me away from the compositional practice of developing core materials into a coherent web of musical connections. Instead, large-scale formal structures were designed, specifying types of material and sectional durations. The materials were shoehorned into structurally determined blocks that were sometimes either considerably longer or shorter than the materials might suggest, resulting in striking continuities, quite different to those that occurred when I had designed my formal structures more intuitively. The deliberate crafting of tension between form and content, in what might be called formal dissonance, has remained a central musical interest of mine to the present. It is my aesthetic preference to hear musical materials stressed in some way, and formal dissonance is an effective way to achieve this effect.

7. MOVING TO CALIFORNIA

In 1985 I enrolled at the University of California at San Diego, attracted by its formidable strength in computer

https://doi.org/10.1017/S1355771808000307 Published online by Cambridge University Press

music, both in personnel and infrastructure. Among the computer music composers and researchers active at UCSD in 1985 were Mark Dolson, Gareth Loy, F. Richard Moore, Roger Reynolds, and visiting composer Gordon Mumma. The UCSD music department was dedicated to all aspects of experimental music, and computer music was well represented at the Computer Audio Research Lab. I began formal training in the mathematics of digital signal processing and foundational techniques of computer music. I quickly learned C programming, used initially for algorithmic generation of note lists, and shortly thereafter for generating formal schemes and DSP sequences. I also learned cmusic, the acoustic compiler of choice at UCSD in the mid-1980s.

8. SYNTHESIS AND PROCESSING STUDIES

My first two UCSD computer music pieces focused respectively on synthesis and processing. The synthetic piece, *Claw* (1987), was built in three superimposed structural layers: (i) a rapid series of high-tessitura FM sounds; (ii) intermittent blocks of feedback filters driven by noise into self-oscillation, then convolved with more noise; and (iii) a drum pattern in which each synthetic percussion sound follows a different tempo curve, gradually pulling the metric grid apart. I designed the structure so that each of the three layers existed in its own distinct sonic space, and never had any causal influence on events in the other layers. Sound example 2 is an excerpt from *Claw*, commencing with the convolved feedback, followed by the initial entry of the drum pattern. The high-tessitura FM layer is heard throughout the example. The processing piece, *Polka Fantasy* (1988), was inspired by the live electronic improvisations in which I was becoming increasingly involved. My goal for the work was to simulate a complex live performance that was also non-causal and thus impossible to perform in real-time. Some sounds would be heard backwards and time stretched, before the original source had been played. The generative source sound from which everything else in the piece was derived, a two-second sample of mariachi music from a Tijuana radio broadcast, is only heard unprocessed at the end. The durational structure was manually composed, specifying constituent blocks to be synthesised by algorithmically designed processing chains. The computation of *Polka Fantasy* took almost four weeks, given the large amount of processing to be realised on a multi-user Vax 11/780 computer with a clock speed of approximately 1 MHz. The cmusic instruments were algorithmically generated in C, in an early example of what I now call 'algorithmic sound design'. The assembly of *Polka Fantasy* was fully automated by shell scripts, with no compositional intervention at any point in the process. Paradoxically, the work sounds quite improvisatory despite its rigid temporal structure. Sound example 3 is an excerpt from *Polka Fantasy*, continuing to the end of the piece where the source sample is finally heard unprocessed.

9. INSTRUMENTAL MUSIC WITH RECORDINGS

I soon began to compose pieces for acoustic instruments and computer-generated music. Despite the excitement that onstage performance brought to tape playback, I had some dissatisfactions with the medium. Computer-generated sound and acoustic instrumental sound often does not blend well. A common solution is to amplify the acoustic instrument, rendering its sound electronic, but I have always preferred the sound of direct acoustic radiation from the instrument. Synchronisation between the performer and the recording often requires reliance on a click-track, an annoyance that, in the worst case, can leak into the sound environment, ruining the performance. It did not bother me that the fixed nature of the recording locks the performer into a rigid temporal structure. However, I was concerned that many subtleties of temporal evolution in electroacoustic music do not have a good notational representation, and thus could not be shared between performer and recording. One solution I favored was to compose computer parts with a clear metric basis, thus eliminating the need for a click-track and enabling satisfying interplay between the performer and recording. Sound example 4 is an excerpt from *String Quartet no. 2*. The pulse-based style obviates the need for a click-track. Despite its advantages, this solution imposes significant stylistic constraints, excluding extreme rubato and music without a clear pulse. Faced with these limitations in the hybrid instrumental/computer medium, I composed the majority of my computer music in the 1990s for fixed medium alone.

10. IMPACT ON INSTRUMENTAL COMPOSITION

Throughout this entire period, I continued writing instrumental music as well. I used the computer to program large-scale background and middle-ground structures that guided this music. However, I did not algorithmically generate instrumental music at the surface level. I derived too much enjoyment from composing the surfaces of my music and saw no reason to turn that responsibility over to the computer. I did compose a few studies in which I generated musical surfaces algorithmically, but the results sounded too generic to me. I felt that humans could do a better job of composing than could a pseudo-random number generator filtered through a set of rules, no matter how sophisticated they might be. My views on algorithmic music generation would evolve, but at the time, viewed from a compositional perspective, the

https://doi.org/10.1017/S1355771808000307 Published online by Cambridge University Press

approach seemed an inefficient way to get at the music I wanted to hear. Of course I continued to employ algorithmic methods for generating larger-scale forms.

Most of my instrumental music since the mid-1980s was highly formalistic in its organisation. I designed all the types of material intended for a given piece before I wrote a single note, just as a chef would assemble all the required ingredients before beginning to cook the meal. The degree of pre-compositional specification varied, but it was always present in some form. Once I began composing computer music, I quickly realised that its sonic vocabulary was effectively unlimited. Unlike the physical limitations inherent in an instrumental ensemble, in a computer-generated piece, any sound could occur at any time. I began to re-evaluate my notions of musical coherence. I have always enjoyed striking moments in classical music that seem to stretch beyond their stylistic boundaries, such as the gloriously rug-pulling C♯ in bar 17 of the final movement of Beethoven's 8th Symphony. Such events would invariably be justified in their local context by the larger musical structure, as is indeed the case for that Beethoven C♯. But I became interested in events that might **not** have a justification in the logic of musical coherence, but would literally come out of nowhere and take the listener on a roller-coaster ride to unanticipated destinations. Sound example 5 is an excerpt from *Bonko Bonzo* (1992) for two pianos, with stylistic juxtapositions inspired by my computer music practice.

11. AUDIO SOFTWARE

By the early 1990s, writing software had become integral to my compositional process, blurring the distinction between composing and programming. In addition to using Csound, cmusic and Cmix, I developed a large battery of FFT-based processors in C. Successful processors were added to a growing library that I would iteratively and randomly apply to sounds through Perl scripting. The automated application of these processors produced an audio data explosion of rich, variegated sonic materials. The broad scope of these materials required a different compositional approach from the strict formalisms that I had so far employed. I increasingly adopted a bottom-up approach, working from the sound to the form, which could no longer be determined in advance of the composition. I combined sounds in mosaic patterning, often using algorithmic mixing strategies.

12. SAMPLING

As my algorithmic sound design processes required a large variety of sounds for input, I developed a significant reliance on sampling in my computer music throughout the 1990s. In addition to sounds that I had recorded, I drew on sounds from such sources as tapes, records, CDs and, increasingly, the World Wide Web. I was acquainted with sampling aesthetics through the work of John Oswald (1988). In the 1990s intellectual property issues were receiving increasing attention, especially as a result of the U2/Negativland legal battle, which was sensationalised in a Mondo 2000 interview of The Edge (Serious 1992). However, IP enforcement did not seem especially risky for the kind of sampling I was practising. At the time, experimental computer music was well below the radar of the pop music industry, and the sample sources I used were so obscure that identification of source recordings seemed extremely unlikely. With the occasional exception of spoken word samples, the samples that went into my pieces became processed beyond recognition. Since the processes were automated, often even I had no idea of the origin of a particular sound that I was working with. The samples were not intended to call attention to themselves as samples; rather, they were employed to create the rich and unpredictable textural environment that was an aesthetic goal for much of my fixed media computer music. To a classically trained composer, sampling seemed a natural compositional tool, given the many historical precedents from Bach to Berio of classical composers repurposing each other's work. Sound example 6 is an excerpt from *Tapeworm* (1994), in which a continuous sample-based texture is algorithmically interspersed with a vocal sample that is gradually extended to reveal its semantic content. As the vocal sample becomes more decipherable, the surrounding music breaks down into increasingly unrelated samples. Following this is a short choral section featuring heavily filtered samples tuned by algorithmically generated harmonic sequences.

13. LIVE COMPUTER MUSIC

I began performing live computer music in the early 1990s. MIDI-based live computer music had been on my radar since the mid-1980s, but it was a direction I did not pursue, for several reasons. Performance limitations inherent in the bandwidth of MIDI did not inspire confidence (Moore 1988). I found the note-centric MIDI protocol of little use, given the timbral focus of my computer music. And I was not enamoured of the sounds emerging from the synthesisers of the mid-1980s, compared to what was possible from direct synthesis on computer. But some interesting alternatives were beginning to emerge. At the 1987 ICMC I saw an early demo of Kyma, a programmable hardware/software system for audio DSP, which looked quite promising. When I arrived at Keio University in 1993, I ordered the system. I incorporated some of the sounds I generated on Kyma to my fixed media compositions. My Keio colleagues Toru Iwatake and Tamami Tono both found striking ways to combine Kyma performance with traditional Japanese musical instruments and practice.

https://doi.org/10.1017/S1355771808000307 Published online by Cambridge University Press

When I moved to IAMAS in 1996, I began performing Kyma in ensemble concerts with my colleague Masahiro Miwa and freelance musicians Yoshihiro Kawasaki, Morley Robertson and Satoro Wono. As my DSP configurations became increasingly complicated it became useful to control Kyma with Max. Sound example 7 is an excerpt from the overture to *Retirement Fund no. 2: The Audit* (1996), a computer music opera with a libretto by Erik Belgum. The drum machine is written in Kyma. The playback increments are controlled in Max. Although I liked the dataflow layout in Max, the process of building Max patches seemed like a more cumbersome version of the scripting that I had been doing for many years with Perl and, before that, C. However, coupled with Kyma, I found Max to be quite useful. And with the introduction of DSP in 1998, Max/MSP became a promising self-contained computer music environment, able to ride the increasing CPU power curve of personal computers.

14. MAX/MSP

In 1999 I wrote my first Max/MSP piece for computer and Piano, *Psychic Driving*, which I performed several times with pianist Daniel Koppelman. Much of the piece alternates between sections for either solo piano or solo computer, with just a few sections where computer and piano play simultaneously. Sound example 8 is an excerpt from *Psychic Driving* in which a solo piano section transitions to a computer processed version of the same passage captured in performance, and gradually mixed with pre-composed computer-generated materials. Toward the end of the piece, the piano cadences on an added 6th chord that must be immediately sampled into a spectral buffer, and then transformed into a sustained electronic resonance. Shortly before composing *Psychic Driving*, I had begun developing FFTease with Christopher Penrose, and this passage was the first to make use of an FFTease external (residency~) in public performance. This passage can be heard in Sound example 9.

The period from 1999 to 2005 saw a gradual transition in my work from fixed media to performed computer music. My fixed media computer music increasingly incorporated elements performed live in Max/MSP. I gave many solo improvisational laptop performances, and arrived at a performance strategy somewhere between composition and improvisation, 'comprovisation', where the overall structure of the piece and DSP strategies are worked out in advance, but the structure allows for considerable variation and exploration during performance. This practice bears some similarities to baroque keyboard performances in which the performer demonstrates both compositional prowess and a flair for extemporisation. Sound example 10 is an excerpt from *Stem Cells* (2004) in which transposition sequences interact with filter patterns, and algorithmic scrambling of the harmonic partials of four sawtooth waves, all manipulated in real-time.

15. COMPUTER CHAMBER MUSIC

When I moved to the University of Manchester in 2005, I encountered local instrumental performance opportunities of unprecedented strength. I commenced work with Kairos, an enthusiastic, professional-quality student ensemble eager to play hybrid computer/instrumental music. The Kairos members would meet to try out work-in-progress on a weekly basis, providing a crucial laboratory for musical experiments in computer/instrumental interaction. Much of my music up to this point had been instrumental, computer-generated, or an uneasy coexistence of the two in my instrument and tape pieces. Kairos provided an opportunity to develop a computer music performance practice that that would tightly mesh with instrumental performance.

16. PERFORMING COMPUTER MUSIC

Since the late 1990s most of the computer music performances I had heard contained a predominant improvisatory element. This was especially true of laptop performances. It is not surprising that improvisation would become predominant in the laptop performance scene. There is no notational practice for live electroacoustic music that approaches the specificity of Western common practice music notation, thus when the fixed nature of recording is placed aside in favour of live performance, an improvisatory approach becomes especially attractive. The best improvisatory computer music performances I have heard, by such artists as Farmer's Manual and George Lewis, have been spectacular. But as a classically trained composer I wanted to investigate compositional approaches to computer performance. My goal was to compose a computer part that could be performed by any musician sufficiently skilled on the instrument, just as one might compose an orchestral part for oboe. This goal has proved elusive. My computer parts are inherently difficult to play, like most of my instrumental parts, and computer performance lacks the historical performance practice on which instrumental performers can draw when learning a new piece. Notating computer parts is difficult, and I still have not found a fully satisfactory solution. My current approach is to write in the score, in text, what I want the computer part to accomplish sonically. I then build a Max/MSP patch that implements the required processing. Finally, I write in the score the actions that need to be taken with the Max/MSP patch in performance. In order to play my computer parts, at a minimum, a musician must know Max/MSP and read common practice notation fluently. I hope that we will see more musicians with both the necessary skills and the inclination to perform computer

https://doi.org/10.1017/S1355771808000307 Published online by Cambridge University Press

chamber music. But this will only happen if chamber music with computer parts provides compelling musical experiences that are not available from either purely instrumental or purely electronic music.

16.1. Early computer chamber music

Prior to my own work in computer chamber music, other classically trained musicians, notably Cort Lippe and Zack Settle, had created instrumental/computer works that are both interactive and in which the parts are largely compositionally determined. Since the early 1990s, Cort Lippe (1996) has shown great ingenuity in composing music that conjoins instrumental performance with the computer. In Lippe's music, as in my own computer chamber music, instrumental performers focus on performing their instrument, without manipulating additional hardware or software interfaces. It is the computer performer's responsibility to manipulate the digital processing environment. In Lippe's music, the instrumentalist is front and centre in the classical manner, and the computer operator is the 'wizard behind the curtain'. I developed a different approach. Instead of being hidden away behind the mixing console, I wanted the computer performer to be up onstage with the rest of the musicians, taking musical risks alongside them. In order to preserve the intimacy of the chamber music setting, I ask the instrumentalists to perform without amplification, and I mix the computer-generated sound to a level that matches the acoustic sound of the instruments in the space. That way the acoustic instruments are heard with all the richness inherent in their acoustic sound radiation patterns. In order to bring the acoustic and computer parts closer, all of the sound from the computer part is based on manipulations of instrumental sounds captured live during performance. A unique aspect of works based on live sampling is their highly contingent nature. Since textures later in the piece will be built upon specific materials sampled earlier on, missing any sampling cues early on can have disastrous results later in the work. And likewise, if a performer is out of tune when playing a passage to be sampled, the passage will be just as out of tune when it recurs, which might happen repeatedly! On the positive side, a correctly played passage will always sound slightly different, so that even a determinate DSP routine applied to sounds captured in performance will produce subtly different results in each performance. This sonic variability makes repeat performances especially rewarding for both performers and listeners.

16.2. Recent computer chamber works

In 2005–06 I composed *Introduction and Allegro* in close collaboration with Kairos. The work was premiered in 2006 by NeXT Ens, the Cincinnati-based group that commissioned the work. The work is a mini-concerto for laptop, with elaborate and virtuosic manipulations of ensemble sound samples, captured during performance. I was present at the premiere, but Margaret Schedel performed the laptop part, to test the theory that someone other than the composer could perform the part. Despite the difficulties posed by the laptop part, Margaret gave a terrific performance. I have since performed the work twice with Kairos. Sound example 11 is an excerpt from *Introduction and Allegro* in which a sequence of instrumental chords are sampled, rapidly segregated, and then performed in an improvised laptop solo. My next computer chamber music work was a string quartet with computer. Whereas *Introduction and Allegro* places the computer in a starring role, in my *String Quartet no. 3 with Computer Enhancements* (2007) the computer performer subtly enhances the string quartet parts, while surreptitiously sampling large blocks of quartet materials that are later reassembled out of sequence into a quodlibet of four recorded string quartets plus the live quartet, all synchronised based on shared harmonies. This quodlibet can be heard in Sound example 12. My most recent work of computer chamber music is a *Trio* for flute, clarinet and computer (2008). The *Trio* is my most complicated computer part to date with approximately 80 different computer configurations that need to be activated and performed at rhythmically precise moments in the piece. Each member of the trio functions at different points in the work as soloist, accompanist or equal contrapuntal partner. In this sense, the *Trio* is my most balanced computer chamber music composition to date.

16.3. Internalising the laptop

During the composition of *String Quartet no. 3* and *Trio* I developed the ability to hear the computer part internally, as I could already do with instrumental parts. During the composition of *Introduction and Allegro* I would bring in snippets of music and Max/MSP patches to try out in rehearsal, and invariably make large changes, revisions and deletions. In *String Quartet no. 3* and beyond, I have been able to sketch out the desired processing as text and code it up later in Max/MSP, and it all generally works as expected when taken to rehearsal. Having developed this capability, I tend to incorporate DSP much more prodigiously now that I merely have to jot down a few notes about the DSP sequence during the composition phase. Another advantage of this intermediary approach is that my Max/MSP patch is just one possible interpretation of the processing scheme written into the score, which could be recreated in a different audio programming language without much difficulty.

17. SPATIAL ORCHESTRATION

While at Dartmouth I composed my first octophonic work, *Trajectories* (2004) in which spatial positioning is

https://doi.org/10.1017/S1355771808000307 Published online by Cambridge University Press

the predominant structural element. Only simple, fixed-spectrum sounds are employed so that they might be heard primarily as carriers of spatial information. The highly complex octophonic trajectories were impractical to realise with standard mixing software, so I composed and assembled the piece using Csound and Perl scripts. *Trajectories* was premiered at the 2004 Harvest Moon Festival in Montreal. In rehearsal I discovered that the channels of *Trajectories* had been assigned randomly rather than to the locations specified in my technical notes. Without the correct spatial assignments, the piece literally collapsed like a jostled soufflé. Fortunately we were able to rapidly re-assign the channels to their correct locations and the piece almost magically reconstituted its structure.

17.1. Multi-channel performance spaces

Following the successful use of elementary synthetic sonic materials in *Trajectories* I finally abandoned my long-standing practice of incorporating samples from external sources into my computer music. Although the use of such samples does reflect an important aspect of today's Web-mediated world, I increasingly felt that referential aspects of some of the samples were becoming a distraction and inhibiting other musical ideas that I wished to explore. *Trajectories* also confirmed the need to press forward with an investigation of spatial attributes of computer music, but with many more channels than the eight that have become standard at most electroacoustic music festivals. With considerably more than eight speakers, any hope of standardisation disappears. And that may be a good thing. We value the unique acoustic properties of our concert halls, musicians adjust their performances to each space accordingly, and we celebrate truly outstanding acoustic spaces such as the Concertgebouw. It should be the same with multi-channel spaces. In adjusting a performance of a given electroacoustic composition from one space to the next, my working premise is that the piece may need to be significantly altered for each performance, either to take advantage of special properties of a particular space, or to redesign existing aspects of the piece which cannot be accommodated in that space, for example three-dimensional panning above and below the audience. I call this approach 'spatial orchestration'. I currently use primarily point source solutions since trajectories remain an important component to my spatial music. But spatial orchestration does not require or exclude any particular technique, and I am experimenting with a hybrid approach combining fixed point source with a limited number of channels that are available for live diffusion. Spatial orchestration is in the earliest stage of any of my computer music projects, but since moving to the Sonic Arts Research Centre at Queen's University Belfast, I have been able to test spatial orchestration techniques in the Sonic Lab, which is an ideal space for this work.

18. ARTICULATED NOISE

Computer chamber music has brought together my two great compositional enthusiasms: computer music and instrumental music. The path to computer chamber music emerged over several years. In late 2007 I discovered another path bringing these two pursuits together. The idea arose from a project related to noise. (For the purpose of this discussion I will refer to computer-generated pseudo-random noise as simply 'noise'.) I had been asked to contribute a composition to the CD Clairaudience: New Music from Electronic Voice Phenomena. Electronic Voice Phenomena or EVP are voices detected in otherwise blank media that has been recorded upon in an empty space said to contain spiritual presences. Inspired by the idea of inferring patterns from noise, an example of a musical 'false positive', I composed *Rose of the World* based on an EVP voice that appears to say, 'I live in spirit'. In *Rose of the World* I submerged the voice sample deeply into noise, reversing the EVP process that had discovered the voice in tape hiss in the first place. I began listening to, and thinking about noise, and found it a very attractive sound. Noise, played at a comfortable listening level, especially with spatial de-correlation, lends itself to all kinds of false pattern recognition: a statistical 'type I error' kind of listening. It's quite easy to superimpose patterns on noise: simply repeating a short block of noise will result in an audible loop. If the loop is short enough, the repetition will result in the sensation of tone. This is not surprising, since looped noise is simply the Karplus-Strong (1983) string simulation algorithm without the low pass filter and feedback element. I found that it was possible to produce and hear more complicated repeating noise patterns. I had already done this in *Trajectories*, albeit with short enveloped noise bursts that made it extremely easy to hear the rhythmic patterns. But it was clear that structured noise could be the basis of subtler sonic counterpoint, as experiments in the Sonic Lab confirmed. During this process, it occurred to me that the phenomenon of hearing patterns in noise could be reproduced in the context of traditionally notated instrumental music. By applying interpretive templates to noise, I could generate specifications for musical surfaces; not quite music, and that is the crucial point: the results of noise processes still need a final layer of compositional detailing to become a musical score. I use noise to make decisions about tempo, template selection, form, duration, pitch structure, pitch selection, and other elements. At the other end of this process, I compose the surface of the piece within very tight, noise-generated constraints. My articulated noise strategy is the opposite of much algorithmic music, where formal

https://doi.org/10.1017/S1355771808000307 Published online by Cambridge University Press

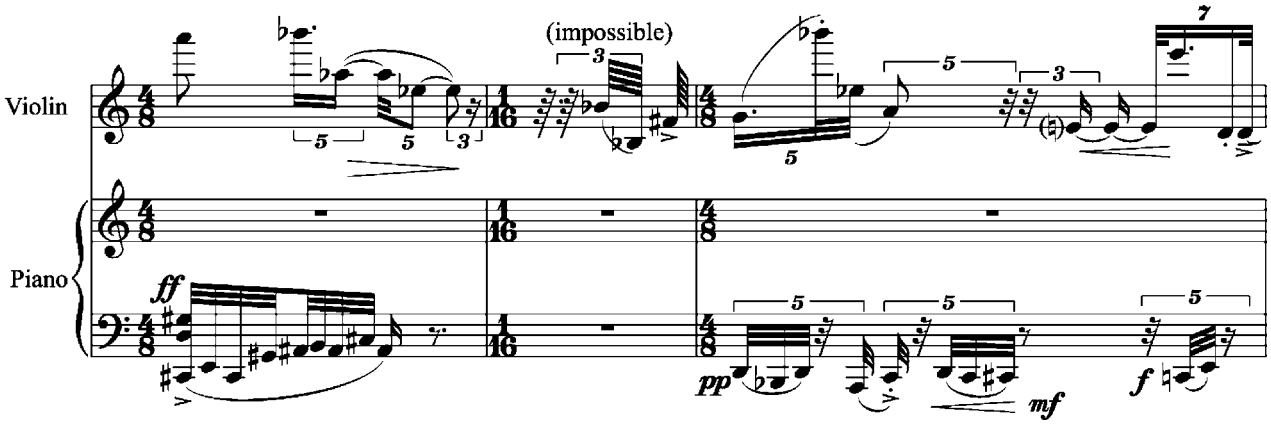


Figure 1. Excerpt from *Three Noises for Violin and Piano*.

decisions are made according to composition prerogatives, and then the surface is randomly generated according to rule-based constraints. By contrast, in articulated noise compositions, most or all significant structural decisions are made at random. But the surface itself is carefully composed. As a result the music seems quite traditional at first, but there is an increasing sense of instability as the piece continually wanders in unexpected directions. An example from a recent articulated noise composition is shown in figure 1.

19. HOW THE WORLD HAS CHANGED

What strikes me most in surveying this time period is the vast disparity in evolutionary speed between the tools of computer music and the tools of instrumental music. While acoustic classical music instruments have changed hardly at all, computer technology has made extraordinary advances. The one advance to instrumental music practice that **has** made a significant impact on my compositions is computer-realised notation. While I dislike the impersonal look of computer-generated scores, this deficiency is more than compensated by the great speed with which digital scores can be produced, modified, and electronically transmitted. I used to spend long hours drafting carefully ruled scores in ink. After a score was completed and a fair copy drafted, making significant structural alterations was nearly unthinkable. But today, making alterations is so easy that I constantly change my scores in preparation for new performances. Notation programs appear to have eliminated the possibility of an urtext.

20. THE FUTURE

Looking forward, I have great optimism for the field of computer music and, by extension, electroacoustic music. Whereas classical musicians and critics have in recent years wondered about the very survival of their field (Horowitz 2005), the ideas, technology and practices of computer-based music are so widespread across genres both inside and far beyond the boundaries of academia that it's hard to imagine a more robust milieu for an artistic practice to project itself into the future. The main problem that confronts a composer entering the field today is choosing from the plethora of promising computer-based musical possibilities. One can only imagine what another 25 years of technological innovation harnessed to musical invention will bring.

REFERENCES

Horowitz, J. 2005. *Classical Music in America: Its Rise and Fall*. New York: W.W. Norton & Co.

Karplus, K., and Strong, A. 1983. Digital Synthesis of Plucked String and Drum Timbres. *Computer Music Journal* **7**(2): 43–55.

Landy, L. 2002. The ElectroAcoustic Resource Site (EARS). http://www.mti.dmu.ac.uk/ears.

Lippe, Cort. 1996. Real-Time Interactive Digital Signal Processing: A View of Computer Music. *Computer Music Journal* **20**(4): 21–4.

Moore, F. R. 1988. The Dysfunctions of MIDI. *Computer Music Journal* **12**(1): 19–28.

Oswald, J. 1988. Plunderphonics, or Audio Piracy as a Compositional Prerogative. http://www.plunderphonics.com/xhtml/xplunder.html.

Serious, R. U. 1992. Negativland Interviews the Edge. http://www.negativland.com/edge.html.

DISCOGRAPHY

Belgum, E., and Lyon, E. 1996. *Retirement Fund no. 2: The Audit*. Voys Records.

Lyon, E. *Bonko Bonzo*. 1995. On *Thump*. New Hampshire: Frog Peak Music, FP13.

Lyon, E. 1996. *Red Velvet*. New Jersey: Smart Noise Records, SNR001.

Lyon, E. 2005. *Ex Cathedra*. Baton Rouge: Centaur Records, CRC2711.

Lyon, E. 2006. *Rose of the World. On Clairaudience: New Music from Electronic Voice Phenomena*. Boston: Bohn Media, BMC001.

https://doi.org/10.1017/S1355771808000307 Published online by Cambridge University Press