Michael Edwards. Algorithmic Composition: Computational Thinking in Music.

Citation for published version:
Edwards, M 2011, 'Michael Edwards. Algorithmic Composition: Computational Thinking in Music.', *Communications of the ACM*, vol. 54, no. 7, pp. 58–67. <http://cacm.acm.org/magazines/2011/7/109891-algorithmic-composition/fulltext>

Document Version:
Peer reviewed version

Published In:
Communications of the ACM

Algorithmic Composition: Computational Thinking in Music

Michael Edwards
Reader in Music Technology
School of Arts, Culture and Environment
University of Edinburgh
Edinburgh, UK
http://uofe.michael-edwards.org
michael.edwards@ed.ac.uk

ABSTRACT

Despite the still-prevalent but essentially nineteenth century perception of the Western creative artist, an algorithmic approach to music composition has been in evidence in Western classical music for at least one thousand years. The history of algorithmic composition—from both before and after the invention of the digital computer—will be presented along with specific techniques and musical examples from the distant and recent past.

Keywords

Algorithmic Composition, Computer-aided Composition, Automatic Composition, Computer Music, Stochastic Music, Xenakis, Ligeti, Lejaren Hiller.

1. INTRODUCTION

In the West, the layman's vision of the creative artist is largely bound in romantic notions of inspiration sacred or secular in origin. Images are plentiful; for example, a man standing tall on a cliff top, the wind blowing through his long hair (naturally), waiting for that particular iconoclastic idea to arrive through the ether.[1] Tales, some even true, of genii penning whole operas in a matter of days, further blur the reality of the usually slowly-wrought process of composition. Mozart, with his speed of writing, is a famous example who to some extent fits the cliché, though perhaps not quite as well as legend would have it.[2]

Non-specialists may be disappointed that composition includes seemingly arbitrary, uninspired formal methods and calculation.[3] What we shall see is that calculation has been part of the Western composition tradition for at least a thousand years. This paper will outline the history of algorithmic composition from the pre- and post-digital computer age, concentrating in particular, but not exclusively, on how it developed out of the avant-garde Western classical tradition in the second half of the twentieth century. This survey will be more illustrative than all-inclusive; it will present examples of particular techniques and some of the music that has been produced with them.

2. A BRIEF HISTORY OF ALGORITHMIC COMPOSITION

Models of musical process are arguably natural to human musical activity. Listening involves both the enjoyment of the sensual sonic experience and the setting up of expectations and possibilities of what is to come: "Retention in short-term memory permits the experience of coherent musical entities, comparison with other events in the musical flow, conscious or subconscious comparison with previous musical experience stored in long-term memory, and the continuous formation of expectations of coming musical events." [8, 42]

This second, active part of musical listening is what gives rise to the possibility, the development of musical form: "Because we spontaneously compare any new feature appearing in consciousness with the features already experienced, and from this comparison draw conclusions about coming features, we pass through the musical edifice as if its construction were present in its totality. The interaction of association, abstraction, memory and prediction is the prerequisite for the formation of the web of relations that renders the conception of musical form possible." [30]

For centuries, composers have taken advantage of this property of music cognition to formalise compositional structure. We cannot of course conflate formal planning with algorithmic techniques, but that the former should lead to the

[1] I'm thinking in particular of Caspar David Friedrich's painting *From the Summit*, in the Hamburg Kunsthalle.

[2] Mozart's compositional process is a complex and often misunderstood matter, complicated by myth—especially regarding his now refuted ability to compose everything in his head [14, 104]—and Mozart's own statements such as "I must finish now, because I've got to write at breakneck speed—everything's composed—but not written yet—" (letter to his father, 30th December 1780). Mozart apparently distinguished between composing (at the keyboard, in sketches) and writing (i.e. preparing the full and final score), hence the confusion about the length of time taken to write certain pieces of music.

[3] For example, in the realm of pitch: transposition, inversion, retrogradation, intervallic expansion or compression; with rhythm: augmentation, diminution, and addition.

latter was, as this paper shall argue, a historical inevitability.

Around 1026, Guido d'Arezzo (the inventor of staff notation) developed a formal technique to set a text to music. A pitch was assigned to each vowel so that the melody varied according to the vowels in the text [21]. The 14th and 15th centuries saw the development of the quasi-algorithmic isorhythmic technique, where rhythmic cycles (*talea*) are repeated, often with melodic cycles (*color*) of the same or differing lengths (potentially, though not generally in practice, leading to very long forms before the beginning of a rhythmic and melodic repeat coincide). Across ages and cultures, repetition, and therefore memory—of short motifs, longer themes, or whole sections—is central to the development of musical form. In the Western context this is seen in various guises: the Classical rondo (with section structures such as ABACA); the Baroque fugue; and the Classical sonata form, with its return not just of themes but tonality too.

Compositions based on number ratios are also found throughout musical history; for example, Dufay's (1400–74) isorhythmic motet *Nuper Rosarum Flores*, written for the consecration of Florence Cathedral on March 25th, 1436. The temporal structure of the motet is based on the ratios 6:4:2:3, these being the proportions of the nave, the crossing, the apse, and the height of the arch of the cathedral. A subject of much debate is how far the use of proportional systems was conscious on the part of various composers, especially with regards to Fibonacci numbers and the Golden Section.[4] Evidence of Fibonacci relationships have been found, for instance, in the music of Bach [32], Schubert [18], and Bartók [27], as well as in various other works of the 20th century [24].

Mozart is thought to have used algorithmic techniques explicitly at least once. His *Musikalisches Würfelspiel* ("Musical Dice")[5] uses musical fragments which are to be combined randomly, according to dice throws (see Figure 1). Such formalisation procedures have not been limited to religious or art music. The Quadrille Melodist, sold by Professor Clinton of the Royal Conservatory of Music, London, in 1865, was marketed as a set of cards which allowed a pianist to generate quadrille music (similar to a square dance). Apparently 428 million quadrilles could be made with the system [34, 823].

Right at the outset of the computer age, algorithmic composition moved straight into the popular, kit-builder's domain. The Geniac Electric Brain of 1956 allowed customers to build a computer with which they could generate automatic tunes (see Figure 2) [36]. Such systems find their modern counterpart in the automatic musical accompaniment software Band-in-a-Box.

2.1 The Avant Garde

After World War II, many Western classical music composers continued to develop the serial[6] technique invented

[4]Fibonacci was the Italian mathematician (c.1170–c.1250) after whom the famous number series is named. This is a simple progression where successive numbers are the sum of the previous two: (0), 1, 1, 2, 3, 5, 8, 13, 21.... As we ascend the sequence, the ratio of two adjacent numbers becomes closer to the so-called Golden Ratio (approx. 1:1.618).

[5]Attributed to Mozart though not officially authenticated despite being designated K. Anh. 294d in the Köchel Catalogue of his works.

[6]Serialism is an organisational system in which pitches (first

	A	B	C	D	E	F	G	H
2	96	22	141	41	105	122	11	30
3	32	6	128	63	146	46	134	81
4	69	95	158	13	153	55	110	24
5	40	17	113	85	161	2	159	100
6	148	74	163	45	80	97	36	107
7	104	157	27	167	154	68	118	91
8	152	60	171	53	99	133	21	127
9	119	84	114	50	140	86	169	94
2	98	142	42	156	75	129	62	123
11	3	87	165	61	135	47	147	33
12	54	130	10	103	28	37	106	5

Figure 1: First part of Mozart's Musikalisches Würfelspiel ("Musical Dice"). Letters over columns refer to eight parts of a waltz; numbers to the left of rows indicate possible values of two thrown dice; numbers in the matrix refer to bar numbers of four pages of musical fragments which are accordingly combined to create the algorithmic waltz.

Figure 2: Part of an advertisement from 1958 for The Geniac Brain, a DIY music computer kit.

by Arnold Schönberg (1874–1951) *et al.* Though generally seen as a radical break with tradition, in light of the earlier historical examples just presented, serialism's detailed organisation can be viewed as no more than a continuation of the tradition of formalising musical composition. Indeed, one of the new generation's criticisms of Schönberg was that he had only radicalised pitch structure, leaving other parameters, such as rhythm, dynamic, even form, in the nineteenth century [5]. They looked to the music of Schönberg's pupil Webern for inspiration in organising these other parameters according to serial principles. Hence the rise of the *total serialists*: Boulez, Stockhausen, Pousseur, Nono *et al* in Europe; Milton Babbitt and his students at Princeton.[7]

Several composers, notably Xenakis (1922–2001) and Ligeti (1923–2006), offered criticisms and alternatives to serialism but, significantly, their music was also often governed by complex, even algorithmic, procedures.[8] The complexity of new composition systems made their implementation in computer programmes ever more attractive. Furthermore, the development of software algorithms in other disciplines made cross-fertilization rife. Thus some techniques are inspired by systems outside the realm of music, e.g. Chaos Theory (Ligeti, *Désordre)*, Neural Networks (Gerhard E. Winkler, *Hybrid II "Networks"*) [39], and Brownian Motion (Xenakis, *Eonta*).

3. COMPUTER-BASED ALGORITHMIC COMPOSITION

Lejaren Hiller (1924–1994) is widely recognised as the first person to have applied computer programmes to algorithmic composition. The use of specially-designed, unique computer hardware was common at US universities in the mid-twentieth century. Hiller used the Illiac computer of the University of Illinois, Urbana-Champaign, to create experimental new music with algorithms. His collaboration with Leonard Isaacson resulted in 1956 in the first known computer-aided composition, *The Illiac Suite for String Quartet*, programmed in binary and using, amongst other techniques, Markov Chains[9] in 'random walk' pitch-generation algorithms [38, 2].

of all) are organised into so-called twelve-tone rows, where each pitch in a musical octave is present and, ideally, equally distributed throughout the piece. This was developed most famously by Arnold Schönberg in the early 1920s at least partially as a response to the difficulty of structuring atonal music i.e. music which has no tonal centre or key (e.g. C major).

[7] At this point we begin to distinguish between pieces which only organise pitch according to the series (dodecaphony) from those which extend organisation into music's other parameters (now strictly speaking serialism, otherwise known as integral or total serialism).

[8] For further discussion and a very approachable introduction to the musical thought of Ligeti and Xenakis, see chapter two of The Musical Timespace [8], in particular pages 36–39.

[9] Familiar no doubt to most readers and first presented in 1906, Markov chains are named after the Russian mathematician Andrey Markov (1856-1922) whose research into random processes led to his eponymous theory. They are amongst the most popular algorithmic composition tools. Being stochastic processes, where future states are dependent on current and perhaps past states, they are perfect for e.g. pitch selection.

Famous for his own random-process influenced compositions if not his work with computers, composer John Cage recognised the potential of Hiller's systems earlier than most. The two collaborated on HPSCHD, a piece for "7 harpsichords playing randomly-processed music by Mozart and other composers, 51 tapes of computer-generated sounds, approximately 5,000 slides of abstract designs and space exploration, and several films" [15]. This was premiered at the University of Illinois at Urbana-Champaign in 1969. Summarising perspicaciously an essential difference between traditional and computer-assisted composition, Cage said in an interview conducted during the composition of HPSCHD that "formerly, when one worked alone, at a given point a decision was made, and one went in one direction rather than another; whereas, in the case of working with another person and with computer facilities, the need to work as though decisions were scarce—as though you had to limit yourself to one idea—is no longer pressing. It's a change from the influences of scarcity or economy to the influences of abundance and—I'd be willing to say—waste." [26, 21].

3.1 Stochastic versus Deterministic procedures

A basic historical division in the world of algorithmic composition is between indeterminate and determinate models, i.e. those that use stochastic/random procedures (e.g. Markov chains) and those where results are fixed by the algorithms and remain unchanged no matter how often the algorithms are run. Examples of the latter are cellular automata (though these can be deterministic or stochastic [34, 860-865]); Lindenmayer Systems (see section 3.3 for more on the deterministic vs. stochastic debate in this context); Charles Ames' *constrained search* algorithms for selecting material properties against a series of constraints[1]; and the compositions of David Cope which use his *Experiments in Musical Intelligence* system [9]. The latter is based on the concept of recombinacy, where new music is created from already existing works; it thus allows the recreation of music in the style of various classical composers, to the shock and delight of many.

3.1.1 Xenakis

Known primarily for his instrumental compositions but also an engineer and architect, Iannis Xenakis was a pioneer of algorithmic composition and computer music. Using language typical for the sci-fi age he wrote: "With the aid of electronic computers, the composer becomes a sort of pilot: he presses buttons, introduces coordinates, and supervises the controls of a cosmic vessel sailing in the space of sound, across sonic constellations and galaxies that he could formerly glimpse only in a distant dream." [40, 144]

Xenakis's approach, which led to the Stochastic Music Programme (henceforth SMP) and radically new pieces such as *Pithoprakta* (1956), used formulae originally developed by scientists to explain the behaviour of gas particles (Maxwell and Boltzmann's Kinetic Theory of Gases) [31, 92]. He saw his stochastic compositions as clouds of sound, individual notes[10] being the analogue of gas particles. The choice and distribution of notes was decided by procedures that involved random choice, probability tables that weigh the occurrence of specific events against those of others. Xenakis created several works with SMP, often more than one

[10] Notes being the combination of pitch and duration, as opposed to simply pitch.

work with the output of a single computer batch process[11] (most probably because of limited access to the IBM 7090 he used for this work). *Eonta* (1963–4), for two trumpets, three tenor trombones, and piano, was composed with SMP. The programme was applied in particular to the creation of the massively complex opening piano solo.

Like another algorithmic composition and computer music pioneer, Gottfried Michael Koenig (1926–), Xenakis had no compunction in adapting the output of his algorithms as he saw fit. Regarding *Atrées* (1962), Matossian claims Xenakis used "75% computer material, composing the remainder himself." [31, 161]. At least in his *Projekt 1* (1964)[12] Koenig saw transcription (i.e. from computer output to musical score) as an important part of the process of algorithmic composition: "Neither the histograms nor the connection algorithm contains any hints about the envisaged, 'unfolded' score, which consists of instructions for dividing the labor of the production changes mode, that is, the division into performance parts. The histogram, unfolded to reveal the individual time and parameter values, has to be split up into voices" [23, 30].

Hiller, on the other hand, believed that if the output of the algorithm is deemed insufficient, then the programme should be modified and the output regenerated [34, 845]. Of course, several programmes which facilitate algorithmic composition include direct connection to their own or third-party computer sound generation.[13] This obviates the need for transcription and even hinders this arguably fruitful intervention. Furthermore, such systems allow the traditional or even conceptual score to become redundant. Thus algorithmic composition techniques allow a fluid and unified relationship between macrostructural musical form and microstructural sound synthesis/processing, as evidenced again by Xenakis in his Dynamic Stochastic Synthesis programme *Gendy3* (1992) [40, 289].

3.2 More current examples

Contemporary techniques tend to be hybrids of deterministic and stochastic approaches. Systems which use techniques from the area of Artificial Intelligence (AI) and/or Linguistics are the generative-grammar[14] based system Bol Processor [3], and expert systems such as Kemal Ebcioglu's CHORAL [10]. Other statistical approaches that use, for instance, Hidden Markov Models (e.g. [19]), tend to need a significant amount of data to train the system; they therefore rely on and generate pastiche copies of the music of a particular composer (which must be codified in machine-readable form) or historical style. Whilst naturally of great significance to researchers in the field of AI, Linguistics, Computer Science, etc., in the author's opinion such systems tend to be of limited use to composers who write music in a modern and personal style (which perhaps resists codification because of its notational and sonic complexity, and, more simply, its lack of sufficient and stylistically consistent data: the so-called *sparse data* problem). But this is also to some extent indicative of the general difficulty of modeling language and human cognition: the software codification of the workings of a spoken language that is understood by many and reasonably standardised is one thing; the codification of the quickly developing and widely divergent field of contemporary music is another matter altogether. Thus we can witness a division in the field between composers who are concerned with creating new music with personalised systems, and researchers interested in developing systems for Machine Learning, AI etc. The latter may quite understandably find it more useful to generate music in well-known styles not only because there is extant data but also because familiarity of material will simplify some aspects of the assessment of results. Naturally though, more collaboration between composers and researchers could lead to very fruitful and aesthetically progressive results.

3.2.1 Outside academia

The application of algorithmic composition techniques has not been restricted to academia or the classical avant garde. Pop/ambient musician Brian Eno (1948–) is known for his admiration and use of generative systems in pieces such as *Music for Airports* (1978). Eno was inspired by the American minimalists, in particular Steve Reich (1936–) and his tape piece *It's gonna rain* (1965). This is not computer music but it is process music, whereby a system is devised—usually repetitive in the case of the minimalists—and allowed to run, generating music in the form of notation or electronic sound. About his *Discreet Music* (1975), Eno said: "Since I have always preferred making plans to executing them, I have gravitated towards situations and systems that, once set into operation, could create music with little or no intervention on my part. That is to say, I tend towards the roles of planner and programmer, and then become an audience to the results" [17, 252].

3.2.2 Improvisation systems

Algorithmic composition techniques are, then, clearly not limited to music of a certain aesthetic or stylistic persuasion. Neither are they limited to a completely fixed view of composition where all the pitches, rhythms, etc., are set down in advance. George Lewis's *Voyager* is a work for human improvisors and "computer-driven, interactive 'virtual improvising orchestra'" [29, 33]. Its roots are, according to Lewis, in the African-American tradition of multidominance, described by him (and borrowing from Jeff Donaldson) as involving multiple simultaneous structural streams, these being in the case of *Voyager* at "both the logical structure of the software and its performance articulation" [29, 34]. Lewis programmed *Voyager* in the Forth language popular with computer musicians in the 1980s. Though in *Voyager* the computer is used to analyse and respond to the human improvisors' input, this is not essential for the programme to generate music (via MIDI[15]). As Lewis writes, "I conceive a performance of Voyager as multiple parallel streams of music generation, emanating from both the computers and the humans—a nonhierarchical, improvisational,

[11] "With a single 45-minute programme on the IBM 7090, he succeeded in producing not only eight compositions which stand up as integral works but also in leading the development of computer-aided composition" [31, 161].

[12] Written to test the rules of serial music but involving random decisions [22].

[13] Especially modern examples such as Common Music, Pure Data, and SuperCollider.

[14] Such systems are generally inspired by Chomsky's grammar models [7] and Lerdahl and Jackendorff's applications of such approaches to generative music theory [28].

[15] MIDI (Musical Instrument Digital Interface): the standard music industry protocol for interconnecting electronic instruments and related devices.

subject-subject model of discourse, rather than a stimulus/response setup" [29, 36]. A related improvisation system, OMAX, from IRCAM, is available within the now more widely used computer music systems Max/MSP and OpenMusic. OMAX uses Artificial Intelligence-based Machine Learning techniques to parse incoming musical data from human musicians, then the results of analysis to generate new material in an improvisatory context [2].

3.2.3 slippery chicken

In my own case, work on the specialised algorithmic composition programme *slippery chicken* [12] has been ongoing since 2000. Written in Common Lisp and its object-oriented extension CLOS, it is mainly deterministic but also has stochastic elements. It has been used to create musical structure for pieces since its inception and is now at the stage where it can generate, in one pass, complete musical scores for traditional instruments, or with the same data write sound files using samples[16] or MIDI file realisations of the instrumental score.[17] The project's main aim is to facilitate a melding of electronic and instrumental sound worlds, not just at the sonic but at the structural level. Hence certain processes common in one medium (for instance audio slicing and looping) are transferred to another (the slicing up of notated musical phrases and the instigation of sub-phrase loops, for example). Techniques for innovative combination of rhythmic and pitch data—in my opinion one of the most difficult aspects of making convincing musical algorithms—are also offered.

3.3 Lindenmayer Systems

Like writing a paper, composing music—perhaps especially with computer-based algorithms—is most often an iterative process. Material is first set down in raw form, only to be edited, developed, and reworked over several passes before the final refined form is achieved. Stochastic procedures, if they are not simply to be used to generate material that is to be reworked by hand or in some other fashion, presents therefore particular problems to the composer. If an alteration of the algorithm is deemed necessary, no matter how small, then re-running the procedure is essential. But this will generate a different set of randomly-controlled results, these perhaps now lacking some of the characteristics the composer deemed musically significant after the first pass.[18]

But deterministic procedures may be more apposite. For instance, Lindenmayer Systems[19] (henceforth L-Systems) whose simplicity, elegance, yet resulting self-similarity make them ideal for composition. Take a very simple example, where a set of rules is defined. These associate a key with a result of two further keys which then in turn form indices for an arbitrary number of iterations of key substitution (see Figure 3).

$$1 \rightarrow 2\ 3$$
$$2 \rightarrow 1\ 3$$
$$3 \rightarrow 2\ 1$$

Figure 3: Simple L-System rules.

Given a starting seed for the lookup and substitution procedure (or rewriting, as it is more generally known), an infinite number of results can be generated (see Figure 4).

Seed: 2
1 3
2 3 | 2 1
1 3 | 2 1 | 1 3 | 2 3
2 3 | 2 1 | 1 3 | 2 3 | 2 3 | 2 1 | 1 3 | 2 1

Figure 4: Step-by-step generation of results from simple L-System rules and a seed.

Self-similarity becomes clear when large result sets are produced (see Figure 5 and note the repetitions of sequences such as 2 1 1 3 or 2 3 2 3).

```
2 3 2 1 1 3 2 3 2 3 2 1 1 3 2 1 1 3 2 1 1 3 2 3 2 3 2
1 1 3 2 3 2 3 2 1 1 3 2 3 2 3 2 1 1 3 2 1 1 3 2 1 1 3
2 3 2 3 2 1 1 3 2 1 1 3 2 1 1 3 2 3 2 3 2 1 1 3 2 1 1
3 2 1 1 3 2 3 2 3 2 1 1 3 2 3 2 3 2 1 1 3 2 3 2 3 2 1
```

Figure 5: Larger result set from simple L-System rules.

These numbers can of course be applied to any musical parameter or material (pitch, rhythm, dynamic, phrase, harmony, etc.) Seen musically, the results of such simple L-Systems tend towards stasis in that only results that are part of the original rules are returned and all results are present throughout the returned sequence. The result is, though, dependent on the rules defined: subtle manipulations of more complex/numerous rules can result in musically interesting developments. Composers have, for instance, used more finessed L-Systems—where the result of a particular rule may be dependent on a sub-rule perhaps—leading to more organic, developing forms. Hanspeter Kyburz's *Cells* for saxophone and ensemble is one such example. Martin Supper describes Kyburz's use of L-Systems in [38, 52]: Results of thirteen generations of L-System rewrites are used to select pre-composed musical motifs. Like Hiller before him, Kyburz uses algorithmic composition techniques to generate and select musical material for the preparation of instrumental scores. The listener, however, will most probably be unaware of the application of software in the composition of such music.

[16] Samples are usually short digital sound files of individual or an arbitrary number of notes/sonic events.

[17] To accomplish this, the software interfaces with parts of the open-source software systems Common Music, Common Lisp Music, and Common Music Notation (all freely available from http://ccrma.stanford.edu/software).

[18] This is, though, a simplistic description of the matter. Most stochastic procedures involve the encapsulation of various tendencies over large data sets, the random details of which are insignificant when compared with the structure of the whole. Still, some details may take on more musical importance than was intended, and to lose these may detrimentally affect the composition. Of course, the composer could avoid such problems by using a random number generator with a fixed and stored seed, guaranteeing that the pseudo-random numbers are generated in the same order each time the process is restarted. Better still would be to modify the algorithm to take these salient though originally unforeseen features into account.

[19] Named after biologist Aristid Lindenmayer (1925–1989) who developed this system (or formal language, based on grammars by Noam Chomsky [33, 3]) which is able to model various natural growth processes, e.g. those of plants.

3.3.1 Transitioning L-Systems: Tramontana

As I tend to write music that is concerned with development and transition, my use of L-Systems is somewhat more convoluted. *Tramontana*, for viola and computer [13] uses L-Systems in the last section. Unlike normal L-Systems however, I employ Transitioning L-Systems, an invention of my own whereby the numbers returned by the L-System are used as lookup indices into a table whose result depends on transitions between related but developing material types. The transitions themselves use Fibonacci-based 'folding-in' structures where the new material is interspersed gradually until it becomes dominant. For example, a transition from material 0 to material 1 may look like Figure 6.

0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0
0 1 0 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 1 1 0 1 1 0 1 1 1 1 0 1 1
1 1 1 1 1 1

Figure 6: Fibonacci-based transition from material 0 to material 1. Note that the first appearance of 1 is at position thirteen, the next being eight positions after this, the next again five positions later, etc., all these numbers being so-called Fibonacci numbers.

In the case of the last section of *Tramontana*, there is a slow development from fast, repeated chords towards more and more flageolets[20] on the C and G strings. Normal pitches and half flageolets[21] then begin to dominate, with a tendency towards more and more of the former. At this point, flageolets on the D string are also introduced. All these developments are created with transitioning L-Systems. The score, a short extract of which is presented in Figure 7, was generated with Bill Schottstaedt's *Common Music Notation* software, taking advantage of its ability to include algorithmically-placed non-standard note heads and other musical signs. It is perhaps worth noting that even before I began work with computers, I was already composing in such a manner. Now, with *slippery chicken* algorithms, it is possible to programme these structures, generate the music, test, re-work, and re-generate, etc., etc. A particular advantage of working with the computer here is that it is a simple matter to extend or shorten sections, something that would be so time-consuming as to become prohibitive with pencil and paper.

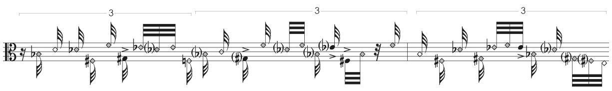

Figure 7: Extract beginning bar 293 of the author's *Tramontana* for viola and computer.

4. MUSICAL EXAMPLE: LIGETI'S DÉSORDRE

György Ligeti (1923-2006) is known to the general public mainly through the use of his music in several Stanley Kubrick films: *2001: A Space Odyssey* uses *Lux Aeterna* and *Requiem* (without Ligeti's permission: this was subjected to a protracted but failed lawsuit); *The Shining* uses *Lontano*; and *Eyes Wide Shut* uses *Musica Ricercata*.

In the late 1950s, after leaving his native Hungary, Ligeti worked in the same studios as Cologne electronic music pioneers Karlheinz Stockhausen and Gottfried Michael Koenig. Nevertheless, he produced very little electronic music of his own. His interest in science and mathematics, however, led to several instrumental pieces influenced by, for example, fractal geometry or chaos theory. But these influences did not lead to a computer-based algorithmic approach:[22] "Somewhere underneath, very deeply, there's a common place in our spirit where the beauty of mathematics and the beauty of music meet. But they don't meet on the level of algorithms or making music by calculation. It's much lower, much deeper—or much higher, you could say." (Ligeti, quoted in [37, 14]).

Nevertheless, as a further example, allow a presentation of the structure of György Ligeti's *Désordre* from his first book of *Piano Études*. This is a particularly fine example for several reasons:

1. The structures of *Désordre* are deceptively simple in concept yet beautifully elegant in effect. The clearly deterministic algorithmic thinking lends itself quite naturally to a software implementation.

2. Ligeti is a major composer, admired by experts and non-experts alike. He is generally not associated with algorithmic composition however. Indeed, *Désordre* was almost certainly composed "algorithmically" by hand, with a pencil and paper, as opposed to at a computer. As such, *Désordre* illustrates the clear link in the history of composition to algorithmic/computational thinking, bringing algorithmic composition back into mainstream musical focus.

3. I have implemented algorithmic models of the first part of *Désordre* in the open-source software system *Pure Data* (PD). This software, and the discussion presented below, is based on analyses by Tobias Kunze [25] (used here with permission) and Hartmut Kinzler [20]. It is freely downloadable [11]; tinkering with the initial data states is instructive and fun.

4.1 Désordre's algorithms

The main argument of *Désordre* consists of foreground and background textures:

- Foreground (accented, loud): two simultaneous instances of the same basic process (melodic/rhythmic: see below for details), one in each hand, both doubled at the octave, and using white note (right hand) and black note[23] (pentatonic, left hand) modes.

- Background (quiet): continuous, generally rising quaver (eighth note) pulse notes, centred between the foreground octaves, one in each hand, in the same mode as the foreground hand.

[20] Familiar to guitarists, flageolets, or harmonics, are special pitches achieved by touching the string lightly with a left-hand finger at a nodal point in order to bring out higher frequencies which are related to the fundamental of the open string by integer multiples.

[21] Half flageolets are achieved by pressing the string as with a full flageolet, but not at a nodal point; the result is a darker, dead-sounding pitch.

[22] Ligeti's son, Lukas, has confirmed to the author that his father was interested conceptually in computers, read a lot about them over the years, but never worked with them in practice.

[23] White and black here refer to the colour of the keys on the modern piano.

In the first part of the piece the basic foreground process consists of a melodic pattern cycle consisting of the scale-step shape given in Figure 8. This is stated on successively higher (right hand, 14 times, 1 diatonic step transposition) and lower (left hand, 11 times, 2 diatonic steps transposition) degrees. Thus a global, long-term movement is created from the middle of the piano outwards, to the high and low extremes.

```
Right hand (white notes), 26 notes, 14 bars
Phrase a:   0  0  1  0  2  1 -1
Phrase a': -1 -1  2  1  3  2 -2
Phrase b:   2  2  4  3  5  4 -1  0  3  2  6  5

Left hand (black notes), 33 notes, 18 bars
Phrase a:   0  0  1  0  2  2  0
Phrase a':  1  1  2  1 -2 -2 -1
Phrase b:   1  1  2  2  0 -1 -4 -3  0 -1  3  2  1 -1  0 -3 -2 -3 -5
```

Figure 8: Foreground melodic pattern (scale steps) of *Désordre* [25].

The foreground rhythmic process consists of slower-moving, irregular combinations of quaver-multiples that tend to reduce in duration over the melodic cycle repeats to create an acceleration towards continuous quaver pulses (see Figure 9).

```
right hand:                                left hand:
cycle 1: a:  3 5 3 5 5 3 7                 3 5 3 5 5 3 8
         a': 3 5 3 5 5 3 7                 3 5 3 5 5 3 8
         b:  3 5 3 5 5 3 3 4 5 3 3 5       3 5 3 5 5 3 3 5 5 3 3 5 3 5 3 5 5 3 8
cycle 2:     3 5 3 4 5 3 8                 3 5 3 5 5 3 8
             3 5 3 4 5 3 8                 3 5 3 5 5 3 8
             3 5 3 4 5 3 3 5 5 3 3 4       3 5 3 5 5 3 3 5 5 3 3 5 3 5 3 5 5 3 8
cycle 3:     3 5 3 5 5 3 7                 3 5 3 5 5 3 8
             3 5 3 5 5 3 7                 3 5 3 5 5 2 7
             3 5 3 5 5 3 3 4 5 3 3 5       3 4 3 4 4 2 2 4 4 2 2 3 2 3 1 3 3 1 4
cycle 4:     3 5 3 4 5 2 7                 1 3 1 2 2 1 3
             2 4 2 4 4 2 5                 1 2 1 2 2 1 3
             2 3 2 3 3 1 1 3 3 1 1 3       1 2 1 2 2 1 1 2 2 1 1 2 1 2 1 2 2 1 3
cycle 5:     1 2 1 2 2 1 3                 1 3 1 2 2 1 3
             1 2 1 2 2 1 3                 1 2 1 2 2 1 3
             1 2 1 2 2 1 1 2 2 1 1 2       1 2 1 2 2 1 1 2 2 1 1 2 1 2 1 2 2 1 2
             ...                           ...
```

Figure 9: Foreground rhythmic pattern (quaver durations) of *Désordre* [25].

The similarity between the two hands' foreground rhythmic structure is obvious, but the duration of seven quavers in the right hand at the end of cycle 1a, as opposed to eight in the left, makes for the clearly audible decoupling of the two parts. This is the beginning of the process of 'disorder', or chaos, and is reflected in the unsynchronised bar lines of the score starting at this point (see Figure 10).

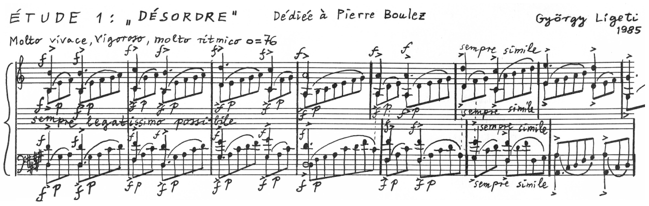

Figure 10: *Désordre*: first system of score (©1986 Schott Music GmbH & Co. KG, Mainz - Germany. Reproduced by permission. All rights reserved.)

To summarise then, in *Désordre* we have a clear, compelling, yet not entirely predictable musical development of rhythmic acceleration coupled with a movement from the middle piano register to the extremes of high and low, all expressed through two related and repeating melodic cycles whose slightly differing lengths result in a combination that dislocates and leads to metrical disorder. I invite the reader to investigate this in more detail by downloading my software implementation available at [11].

5. CONCLUSION: RESISTANCE TO ALGORITHMIC COMPOSITION

There has been considerable resistance to algorithmic composition from all sides, from musicians to the general public. This resistance bears comparison to the reception of the supposedly overly-mathematical serial approach established by the composers of the Second Viennese School. Alongside the techniques of other music composed from the beginning of the twentieth century onwards, the serial principle itself is frequently considered to be the reason why the music—so-called modern music, but now actually close to a hundred years old—may not appeal. I propose that a more enlightened approach to the arts in general, especially those that present a challenge, would be a more inward-looking examination of the individual response, a deferral of judgment and acknowledgment that, first and foremost, a lack of familiarity with the style and content may lead to a neutral or negative response. Only after further investigation and familiarisation can deficiencies in the work be considered.[24]

Algorithmic composition is often viewed as a sideline in contemporary musical activity, as opposed to a logical application and incorporation of compositional technique into the digital domain. Without wishing to imply that instrumental composition is in a general state of stagnation, if the computer is *the* universal tool—there is surely no doubt—then not to apply it to composition would be, if not exactly an example of Ludditism, then at least to risk missing important aesthetic developments that only the computer can stimulate and facilitate, and which other artistic fields are already taking advantage of. That algorithmic thinking has been present in Western composition for at least a thousand years has been established. That such thinking should lend itself to formalisation in software algorithms was inevitable.

But Hiller's work and his 1959 article for the *Scientific American* [16] led to much controversy and press attention. Hostility to his achievements[25] was such that the Grove Dictionary of Music and Musicians[26] did not include an article on it until shortly before his death. This hostility arose no doubt more from a misperception of compositional practice than from anything intrinsic to Hiller's work.

Much of the resistance to algorithmic composition that persists to this day stems from a basic misunderstanding

[24]To paraphrase Ludger Brümmer, from information theory we know that new information is perceived as chaotic or interesting but not expressive. New information needs to be structured before it can be understood, and in the case of aesthetic experience, this structuring process involves comparison to an ideal, i.e. an established notion of beauty [6, 36].

[25]Speaking of the reaction to *The Illiac Suite*, Hiller said "There was a great [deal] of hostility, certainly in the musical world... I was immediately pigeonholed as an ex-chemist who had bungled into writing music and probably wouldn't know how to resolve a dominant seventh chord." (Interview with Vincent Plush, 1983, from [4, 12].)

[26]The Grove is the English-speaking world's most widely-used and arguably authoritative musicological resource.

that the computers compose the music, not the composer. This is, in the vast majority of cases where the composer is also the programmer, simply not true. As Curtis Roads points out, it takes a good composer to design algorithms that will result in music that captures the imagination [34, 852].

Furthermore, using algorithmic composition techniques does not by necessity imply less composition work or a shortcut to musical results; rather, it is a change of focus from note-to-note composition to a top-down formalisation of compositional process. Composition is in fact often slowed down by the requirement to express musical ideas and encapsulate their characteristics in a highly structured and non-musical general programming language. Learning the discipline of programming itself is an altogether time-consuming and, for some composers, insurmountable problem.

Perhaps counter-intuitively though, such a formalisation of personal composition technique allows the composer to proceed from concrete musical or abstract formal ideas into realms hitherto unimagined—some, I would argue, impossible to achieve via any other means than with computer software. And as composer Helmut Lachenmann wrote, “a composer who knows exactly what he wants, wants only what he knows—and that is one way or another too little” [35, 24]. The computer can help composers overcome recreating what they and we already know by aiding more thorough investigations of material: once procedures are programmed, modifications and manipulations are simpler than with traditional pen and paper. By “pressing buttons, introducing coordinates, and supervising the controls,” to quote Xenakis again [40, 144], the composer is able to stand back and develop compositional material *en masse*, applying procedures and assessing, rejecting, accepting, or further processing results of an often surprising nature. Algorithmic composition techniques clearly further individual musical and compositional development through computer-programming enabled voyages of musical discovery.

6. REFERENCES

[1] Charles Ames. Stylistic Automata in “Gradient”. *Computer Music Journal*, 7(4):45–56, 1983.

[2] Gérard Assayag, Georges Bloch, Marc Chemillier, Arshia Cont, and Shlomo Dubnov. OMax brothers: a Dynamic Topology of Agents for Improvization Learning. In *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pages 125–132, Santa Barbara, California, USA, 2006. ACM.

[3] Bernard Bel. Migrating Musical Concepts: An Overview of the Bol Processor. *Computer Music Journal*, 22(2):56–64, 1998.

[4] John Bewley. Lejaren A. Hiller: Computer Music Pioneer. Music Library Exhibit, University of Buffalo, 2004. PDF available at http://library.buffalo.edu/libraries/units/music/exhibits/hillerexhibitsummary.pdf (accessed August 12th 2009).

[5] Pierre Boulez. Schoenberg est mort. *Score*, (6):18–22, February 1952.

[6] Ludger Brümmer. Using a Digital Synthesis Language in Composition. *Computer Music Journal*, 18(4):35–46, 1994.

[7] Noam Chomsky. *Syntactic Structures*. Mouton, The Hague, 1957.

[8] Erik Christensen. *The Musical Timespace, a Theory of Music Listening*. Aalborg University Press, Aalborg, 1996.

[9] David Cope. *Experiments in Musical Intelligence*. A-R Editions, Madison, WI, 1996.

[10] Kemal Ebcioglu. An Expert System for Harmonizing Four-Part chorales. *Computer Music Journal*, 12(3):43–51, 1988.

[11] Michael Edwards. A Pure Data implementation of Ligeti’s Désordre. Open-source music software. http://www.michael-edwards.org/software/desordre.zip.

[12] Michael Edwards. slippery chicken: a specialised algorithmic composition program. Unpublished object-oriented Common Lisp software. See http://www.michael-edwards.org/slippery-chicken.

[13] Michael Edwards. Tramontana. Sheet music: sumtone, 2004. http://www.sumtone.com/work.php?workid=101.

[14] Cliff Eisen and Simon P. Keefe, editors. *The Cambridge Mozart Encyclopedia*. Cambridge University Press, Cambridge, 2006.

[15] The Electronic Music Foundation. HPSCHD. http://emfinstitute.emf.org/exhibits/hpschd.html (accessed 17th August 2009).

[16] Lejaren Hiller. Computer Music. *Scientific American*, 201(6):109–120, December 1959.

[17] Thomas Holmes. *Electronic and Experimental Music*. Taylor & Francis Ltd, London, 2003.

[18] Roy Howat. Architecture as drama in late Schubert. In Brian Newbould, editor, *Schubert Studies*, pages 168–192. Ashgate Press, London, 1998.

[19] Anna Jordanous and Alan Smail. Artificially Intelligent Accompaniment using Hidden Markov Models to Model Musical Structure. In C. Tsougras and R. Parncutt, editors, *Proceedings of the fourth Conference on Interdisciplinary Musicology (CIM08)*, 2008.

[20] Hartmut Kinzler. György Ligeti: decision and automatism in “Désordre”, 1er Étude, Premier Livre. *Interface, Journal of New Music Research*, 20(2):89–124, 1991.

[21] H. Kirchmeyer. On the historical construction of rationalistic music. *Die Reihe*, 8:11–29, 1962.

[22] Gottfried Michael Koenig. Project 1. http://home.planet.nl/ gkoenig/indexe.htm (accessed 17th August 2009).

[23] Gottfried Michael Koenig. Aesthetic Integration of Computer-Composer Scores. *Computer Music Journal*, 7(4):27–32, 1983.

[24] J Kramer. The Fibonacci Series in Twentieth Century Music. *Journal of Music Theory*, (17):111–148, 1973.

[25] Tobias Kunze. Désordre. Unpublished article available at http://www.fictive.com/t/pbl/1999_desordre/ligeti.html (accessed 5th August 2009).

[26] Larry Austin, John Cage, and Lejaren Hiller. An Interview with John Cage and Lejaren Hiller. *Computer Music Journal*, 16(4):15–29, 1992.

[27] Erno Lendvai. *Bela Bartók: an analysis of his music*. Kahn & Averill, London, 1971.

[28] Lerdahl and Jackendorff. *A Generative Theory of Tonal Music*. MIT Press, Cambridge, Mass., 1983.

[29] George Lewis. Too Many Notes: Computers, Complexity and Culture in Voyager. *Leonardo Music Journal*, 10:33–39, 2000.

[30] György Ligeti. Über Form in der neuen Musik. *Darmstädter Beiträge zur neuen Musik*, 10:23–35, 1966.

[31] Nouritza Matossian. *Xenakis*. Kahn & Averill, London, 1986.

[32] Hugo Norden. Proportions in Music. *Fibonacci Quarterly*, 2:219–222, 1964.

[33] Prusinkiewicz and Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag, New York, 1990.

[34] Curtis Roads. *The Computer Music Tutorial*. MIT Press, Cambridge, Massachusetts, 1996.

[35] David Ryan and Helmut Lachenmann. Composer in Interview: Helmut Lachenmann. *Tempo*, (210):20–24, 1999.

[36] J Sowa. *A machine to compose music*. Oliver Garfield Co., New Haven, 1956. Instruction manual for GENIAC.

[37] Richard Steinitz. Music, Maths & Chaos. *Musical Times*, 137(1837):14–20, March 1996.

[38] Martin Supper. A Few Remarks on Algorithmic Composition. *Computer Music Journal*, 25(1):48–53, 2001.

[39] Gerhard E Winkler. Hybrid II: "Networks". CD recording, 2003. sumtone cd1: *stryngebite*. See http://www.sumtone.com/recording.php?id=17.

[40] Iannis Xenakis. *Formalized Music*. Pendragon, Hillsdale NY, 1992.