DOI:10.1145/1965724.1965742

The composer still composes but also gets to take a programming-enabled journey of musical discovery.

BY MICHAEL EDWARDS

Algorithmic Composition: Computational Thinking in Music

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, waiting for that particular iconoclastic idea to arrive through the ether.[a] 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 celebrated 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.[b]

Non-specialists may be disappointed that composition includes seemingly arbitrary, uninspired formal methods and calculation.[c] What we shall see here is that calculation has been part of the Western composition tradition for at least 1,000 years, This article outlines the history of algorithmic composition from the pre- and post-digital computer age, concentrating, but not exclusively, on how it developed out of the avant-garde Western classical tradition in the second half of the 20th century. This survey is more illustrative than all-inclusive, presenting examples of particular techniques and some of the music that has been produced with them.

A Brief History

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: musicologist Erik Christensen described it as follows: "Retention in short-term

» key insights

- **Music composition has always been guided by the composer's own computational thinking, sometimes even more than by traditional understanding of inspiration.**
- **Formalization of compositional technique in software can free the mind from musical and cultural clichés and lead to startlingly original results.**
- **Algorithmic composition systems cover all aesthetics and styles, with some open-ended variants offering an alternative to the fixed, never-changing compositions that for most of us define the musical limits.**

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

[b] Mozart's compositional process is complex and often misunderstood, complicated by myth, especially regarding his now refuted ability to compose everything in his head[15] and his own statements (such as "I must finish now, because I've got to write at breakneck speed—everything's composed—but not written yet" in a letter to his father, Dec. 30, 1780). Mozart apparently distinguished between composing (at the keyboard, in sketches) and writing (preparing a full and final score), hence the confusion about the length of time taken to write certain pieces of music.

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

ILLUSTRATION BY STUDIO TONNE

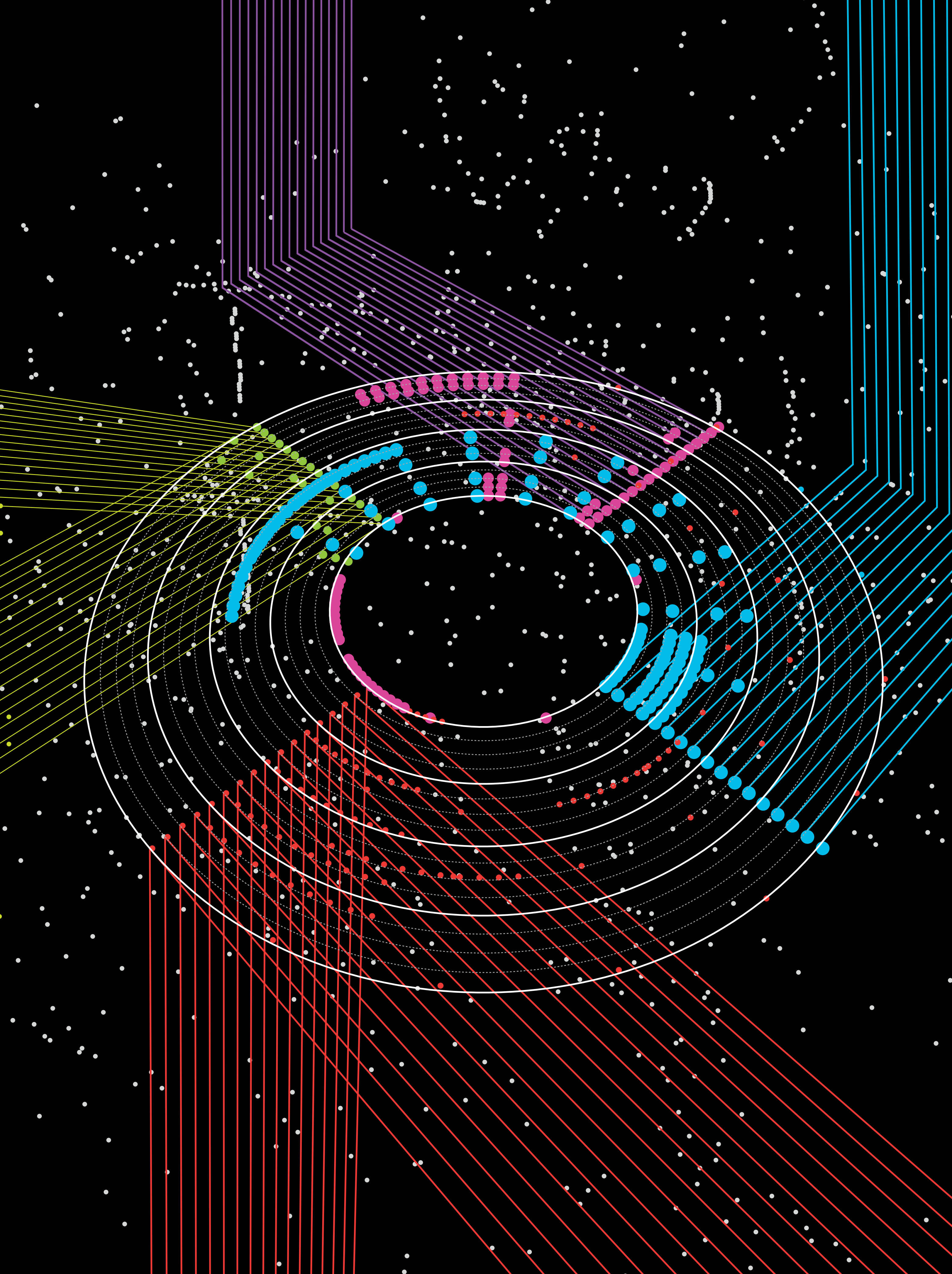

Figure 1. First part of Mozart's *Musikalisches Würfelspiel* ("Musical Dice"): Letters over columns refer to eight parts of a waltz; numbers to the left of rows indicate possible values of two thrown dice; and numbers in the matrix refer to bar numbers of four pages of musical fragments combined to create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

	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

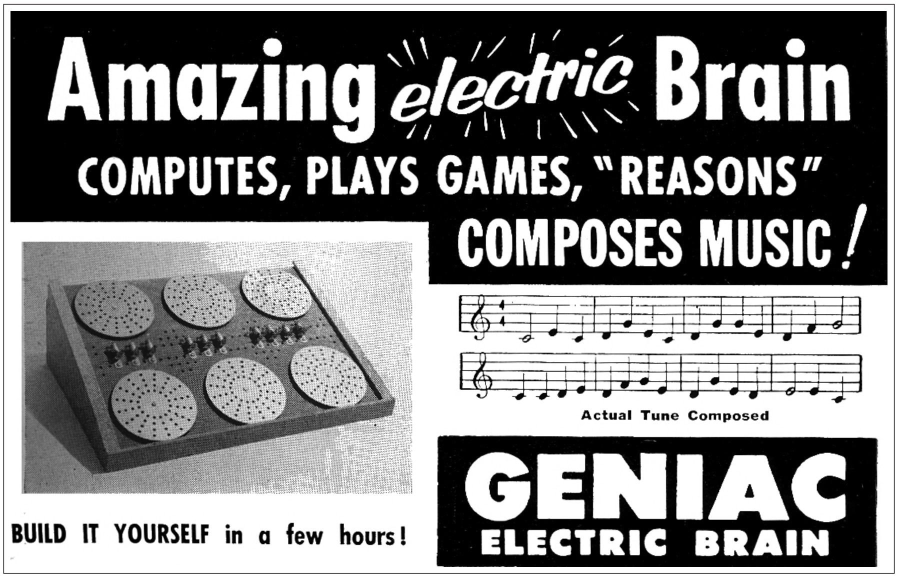

Figure 2. Part of an advertisement for *The Geniac Electric Brain*, a DIY music-computer kit.

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."[9]

This second active part of musical listening is what gives rise to the possibility and development of musical form; composer György Ligeti wrote, "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 formalize compositional structure. We cannot, of course, conflate formal planning with algorithmic techniques, but that the former should lead to the latter was, as I argue here, an 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 the melody varied according to the vowels in the text.[22] The 14th and 15th centuries saw 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, and whole sections) is central to the development of musical form. In the Western context, this repetition is seen in various guises, including 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 to tonality, too.

Compositions based on number ratios are also found throughout Western musical history; for example, Guillaume Dufay's (1400–1474) isorhythmic motet *Nuper Rosarum Flores*, written for the consecration of Florence Cathedral, March 25, 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.[d] Evidence of Fibonacci relationships haas been found in, for instance, the music of Bach,[32] Schubert,[19] and Bartók,[27] as well as in various other works of the 20th century.[25]

Mozart is thought to have used algorithmic techniques explicitly at least once. His *Musikalisches Würfelspiel* ("Musical Dice")[e] uses musical fragments that are to be combined randomly according to dice throws (see Figure 1). Such formalization procedures are

[d] Fibonacci was an Italian mathematician (c.1170–c.1250) for 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... Ascending the sequence, the ratio of two adjacent numbers gets closer to the so-called Golden Ratio (approximately 1:1.618).

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

not limited to religious or art music. The *Quadrille Melodist*, sold by Professor J. Clinton of the Royal Conservatory of Music, London (1865) was marketed as a set of cards that allowed a pianist to generate quadrille music (similar to a square dance). The system could apparently make 428 million quadrilles.[34]

Right at the outset of the computer age, algorithmic composition moved straight into the popular, kit-builder's domain. The Geniac Electric Brain 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 (http://band-in-a-box.com/).

The avant-garde. After World War II, many Western classical music composers continued to develop the serial[f] technique invented 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 organization can be viewed as no more than a continuation of the tradition of formalizing musical composition. Indeed, one of the new generation's criticisms of Schönberg was that he radicalized only pitch structure, leaving other parameters (such as rhythm, dynamic, even form) in the 19th century.[6] They looked to the music of Schönberg's pupil Anton von Webern for inspiration in organizing these other parameters according to serial principles. Hence the rise of the total serialists: Boulez, Stockhausen, Pousseur, Nono, and others in Europe, and Milton Babbitt and his students at Princeton.[g]

Much of the resistance to algorithmic composition that persists to this day stems from the misguided bias that the computer, not the composer, composes the music.

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

Computer-Based Algorithmic Composition

Lejaren Hiller (1924–1994) is widely recognized as the first composer to have applied computer programs to algorithmic composition. The use of specially designed, unique computer hardware was common at U.S. universities in the mid-20th century. Hiller used the Illiac computer at 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, among other techniques, Markov Chains[i] in "random walk" pitch-generation algorithms.[38]

Famous for his own random-process-influenced compositions, if not his work with computers, composer John Cage recognized 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

[f] Serialism is an organizational system in which pitches (first of all) are organized into so-called 12-tone rows, where each pitch in a musical octave is present and, ideally, equally distributed throughout the piece. This technique was developed most famously by Schönberg in the early 1920s at least in part as a response to the difficulty of structuring atonal music, music with no tonal center or key (such as C major).

[g] Here, we begin to distinguish between pieces that organize pitch only according to the series (dodecaphony) from those extending organization into music's other parameters—strictly speaking serialism, also known as integral or total serialism.

[h] For a very approachable introduction to the musical thought of Ligeti and Xenakis, see *The Musical Timespace*, chapter 2,[9] particularly pages 36–39.

[i] First presented in 1906, Markov chains are named for the Russian mathematician Andrey Markov (1856–1922), whose research into random processes led to his eponymous theory, and today are among the most popular algorithmic composition tools. Being stochastic processes, where future states are dependent on current and perhaps past states, they are applicable to, say, pitch selection.

designs and space exploration, and several films."[16] It premiered at the University of Illinois, Urbana-Champaign, in 1969. Summarizing perspicaciously an essential difference between traditional and computer-assisted composition, Cage said in an interview during the composition of *HPSCHD*, "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."[3]

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.

Stochastic versus deterministic procedures. A basic historical division in the world of algorithmic composition is between indeterminate and determinate models, or those that use stochastic/random procedures (such as 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 they can be deterministic or stochastic[34]); Lindenmayer Systems (see the section on the deterministic versus stochastic debate in this context); Charles Ames's constrained search algorithms for selecting material properties against a series of constraints[1]; and the compositions of David Cope that use his *Experiments in Musical Intelligence* system.[10] The latter is based on the concept of "recombinacy," where new music is created from existing works, thus allowing the recreation of music in the style of various classical composers, to the shock and delight of many.

Figure 3. Simple L-System rules.

1 → 2 3
2 → 1 3
3 → 2 1

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

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

Xenakis. Known primarily for his instrumental compositions but also as an engineer and architect, Iannis Xenakis was a pioneer of algorithmic composition and computer music. Using language typical of 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]

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 behavior of gas particles (Maxwell's and Boltzmann's Kinetic Theory of Gases).[31] He saw his stochastic compositions as clouds of sound, with individual notes[j] as the analogue of gas particles. The choice and distribution of notes was determined by procedures involving random choice, probability tables weighing the occurrence of specific events against those of others. Xenakis created several works with SMP, often more than one with the output of a single computer batch process,[k] probably due to limited access to the IBM 7090 he used. His *Eonta* (1963–1964) for two trumpets, three tenor trombones, and piano was composed with SMP. The program 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 adapting the output of his algorithms as he saw fit. Regarding *Atrées* (1962), Xenakis's biographer Nouritza Matossian claims Xenakis used "75% computer material,

[j] Notes are a combination of pitch and duration, rather than just pitch.

[k] Matossian wrote, "With a single 45-minute program on the IBM 7090, he [Xenakis] succeeded in producing not only eight compositions that stand up as integral works but also in leading the development of computer-aided composition."[31]

composing the remainder himself."[31] At least in Koenig's *Projekt 1* (1964)[l] Koenig saw transcription (from computer output to musical score) as an important part of the process of algorithmic composition, writing, "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."[24]

Hiller, on the other hand, believed that if the output of the algorithm is deemed insufficient, then the program should be modified and the output regenerated.[34] Several programs that facilitate algorithmic composition include direct connection to their own or to third-party computer sound generation.[m] This connection obviates the need for transcription and even hinders this arguably fruitful intervention. Furthermore, such systems allow the traditional or even conceptual score to be 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* program *Gendy3* (1992).[40]

More current examples. Contemporary (late 20th century) techniques tend to be hybrids of deterministic and stochastic approaches. Systems using techniques from artificial intelligence (AI) and/or linguistics are the generative-grammar[n]-based system Bol Processor software[4] and expert systems (such as Kemal Ebcioglu's CHORAL[11]). Other statistical approaches that use, say, Hidden Markov Models (as in Jordanous and Smaill[20]), 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 (that must be codified in machine-readable form) or historical style. While naturally significant to AI research, linguistics, and computer science, such systems tend to be of limited use to composers writing music in a modern and personal style that 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 understood by many and reasonably standardized is one thing; the codification of the quickly developing and widely divergent field of contemporary music is another thing altogether. Thus we can witness a division between composers concerned with creating new music with personalized systems and researchers interested in developing systems for machine learning and AI. 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 simplifies some aspects of the assessment of results. Naturally though, more collaboration between composers and researchers could lead to fruitful, aesthetically progressive results.

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

```
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
```

Outside academia. Application of algorithmic-composition techniques is not restricted to academia or to the classical avant garde. Pop/ambient musician Brian Eno (1948–) is known for his admiration and use of generative systems in *Music for Airports* (1978) and other pieces. 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 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. Eno said about his *Discreet Music* (1975), "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."[18]

Improvisation systems. Algorithmic composition techniques are, then, clearly not limited to music of a certain aesthetic or stylistic persuasion. Nor are they limited to a completely fixed view of composition, where all the pitches and rhythms are set down in advance. George Lewis's *Voyager* is a work for human improvisors and "computer-driven, interactive 'virtual improvising orchestra.'"[29] Its roots are, according to Lewis, in the African-American tradition of multi-dominance, described by him (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] Lewis programmed *Voyager* in the Forth language popular with computer musicians in the 1980s. Though in *Voyager* the computer is used to analyze and respond to a human improviser, such input is not essential for the program to generate music (via MIDI[o]). Lewis wrote, "I conceive a performance of *Voyager* as multiple parallel streams of music generation, emanating from both the computers and the humans—a nonhierarchical, improvisational, subject-subject model of discourse, rather than a stimulus/response setup."[29] A related improvisation system, OMAX, from the Institut de Recherche et Coordina-

[l] Written to test the rules of serial music but involving random decisions.[23]

[m] Especially modern examples (such as Common Music, Pure Data, and SuperCollider).

[n] Such systems are generally inspired by Chomsky's grammar models[8] and Lerdahl's and Jackendorff's applications of such approaches to generative music theory.[28]

[o] Musical Instrument Digital Interface, or MIDI, the standard music-industry protocol for interconnecting electronic instruments and related devices.

tion Acoustique/Musique in Paris, is available within the now more widely used computer-music systems Max/MSP and Open-Music. OMAX uses AI-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]

slippery chicken. In my own case, work on the specialized algorithmic composition program *slippery chicken*[13] is ongoing since 2000. Written in Common Lisp and its object-oriented extension, the Common Lisp Object System, 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 a single pass, complete musical scores for traditional instruments or with the same data write sound files using samples[p] or MIDI file realizations of the instrumental score.[q] 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 (such as audio slicing and looping) are transferred to another (such as the slicing up of notated musical phrases and instigation of sub-phrase loops). Also offered are techniques for innovative combination of rhythmic and pitch data, which is, in my opinion, one of the most difficult aspects of making convincing musical algorithms.

Figure 6. Fibonacci-based transition from material 0 to material 1. Note the first appearance of 1 is at position 13, with the next eight positions after that, the next again five positions after that, and so on; all these numbers are so-called Fibonacci numbers.

```
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
```

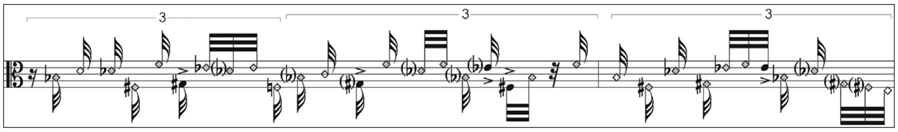

Figure 7. Extract beginning bar 293 of the author's *Tramontana* for viola and computer.

Figure 8. Foreground melodic pattern (scale steps) of *Désordre*.[26]

```
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
```

Lindenmayer systems. Like writing a paper, composing music, 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. For the composer, stochastic procedures, if not simply to be used to generate material to be reworked by hand or in some other fashion, represent particular problems. If an alteration of the algorithm is deemed necessary, no matter how small, then rerunning the procedure is essential. But rerunning will generate a different set of randomly controlled results, perhaps now lacking some characteristics the composer deemed musically significant after the first pass.[r]

Deterministic procedures can be more apposite. For instance, Lindenmayer Systems[s] (henceforth L-Systems) whose simplicity and elegance yet resulting self-similarity make them ideal for composition. Take a simple example, where a set of rules is defined and associates a key with a result of two further keys that in turn form indices for an arbitrary number of iterations of key substitution (see Figure 3).

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).

Self-similarity is clear when larger result sets are produced; see Figure 5, noting the repetitions of sequences (such as 2 1 1 3 and 2 3 2 3). These numbers can be applied to any musical parameter or material, including pitch, rhythm, dynamic, phrase, and harmony. Seen musically, the results of such simple L-Systems tend toward stasis in that only results that are part of the original rules are returned, and all results are present throughout the returned sequence. However, the result is dependent on the rules defined: subtle manipulations of more complex/numerous rules can result in musically interesting developments. For instance, composers have used more finessed L-Systems—where the result

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

[q] 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.

[r] This is a simplistic description. Most stochastic procedures involve encapsulation of various tendencies over arbitrarily large data sets, the random details of which are insignificant compared to the structure of the whole. Still, some details may take on more musical importance than intended, and losing them may detrimentally affect the composition. The composer could avoid such problems by using a random number generator with fixed and stored seed, guaranteeing 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.

[s] Named for biologist Aristid Lindenmayer (1925–1989) who developed this system (or formal language, based on grammars by Noam Chomsky[33]) that can model various natural-growth processes (such as those of plants).

of a particular rule may be dependent on a sub-rule—leading to more organic, developing forms. Hanspeter Kyburz's (1960–) *Cells* for saxophone and ensemble is an example. Martin Supper[38] described Kyburz's use of L-Systems, using results from 13 generations of L-System rewrites to select precomposed musical motifs. Like Hiller before him, Kyburz uses algorithmic composition techniques to generate and select musical material for the preparation of instrumental scores. However, the listener is probably unaware of the application of software in the composition of such music.

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. My own *Tramontana* (2004) for viola and computer[14] uses L-Systems in its concluding section. Unlike normal L-Systems, however, I employ Transitioning L-Systems, my own invention, 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. 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 might look like Figure 6.

In the case of the concluding section of *Tramontana*, there is slow development from fast, repeated chords toward more and more flageolets[t] on the C and G strings. Normal pitches and half flageolets[u] begin to dominate, with a tendency toward 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 (see Figure 7 for a short extract) was generated with Bill Schottstaedt's *Common Music Notation* software, taking advantage of its ability to include algorithmically placed nonstandard note heads and other musical signs. Perhaps worth noting is that even before I began work with computers, I was already composing in such a manner. Now, with *slippery chicken* algorithms, these structures can be programmed to generate the music, test, re-work, and re-generate. A particular advantage of working with the computer here is that it is a simple matter to extend or shorten sections, something that would, with pencil and paper, be so time-consuming as to be prohibitive.

CURTIS ROADS, 1996

"It takes a good composer to design algorithms that result in music that captures the imagination."

Musical Example: Ligeti's *Désordre*

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

After leaving his native Hungary in the late 1950s, Ligeti worked in the same studios as Cologne electronic music pioneers Karlheinz Stockhausen and Gottfried Michael Koenig though produced little electronic music of his own. However, his interest in science and mathematics led to several instrumental pieces influenced by, for example, fractal geometry and chaos theory. But these influences did not lead to a computer-based algorithmic approach.[v] He was quoted in Steinitz[37] saying, "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."

Nevertheless, as a further example, we shall consider the structure of György Ligeti's *Désordre* from his first book of *Piano Etudes* for several reasons:

Structures. The structures of *Désordre* are deceptively simple in concept

[t] Familiar to guitarists, flageolets, and 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 related to the fundamental of the open string by integer multiples.

[u] 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.

[v] Ligeti's son, Lukas, confirmed to me that his father was interested conceptually in computers, reading about them over the years, but never worked with them in practice.

yet beautifully elegant in effect, where the clearly deterministic algorithmic thinking lends itself quite naturally to software implementation;

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

Algorithmic models. I have implemented algorithmic models of the first part of *Désordre* in the open-source software system *Pure Data*, which, along with the following discussion, is based on analyses by Tobias Kunze,[26] used here with permission, and Hartmut Kinzler.[21] It is freely downloadable from my Web site http://www.michael-edwards.org/software/desordre.zip[12]; tinkering with the initial data states is instructive and fun.

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, one in each hand, both doubled at the octave, and white note (righthand) and black-note[w] (pentatonic, lefthand) modes; and

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

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

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 toward continuous quaver pulses (see Figure 9).

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 unsynchronized bar lines of the score starting at this point (see Figure 10).

In *Désordre* we experience 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 with slightly differing lengths resulting 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.[12]

Figure 9. Foreground rhythmic pattern (quaver/eighth-note durations) of *Désordre*.[26]

```
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
...                             ...
```

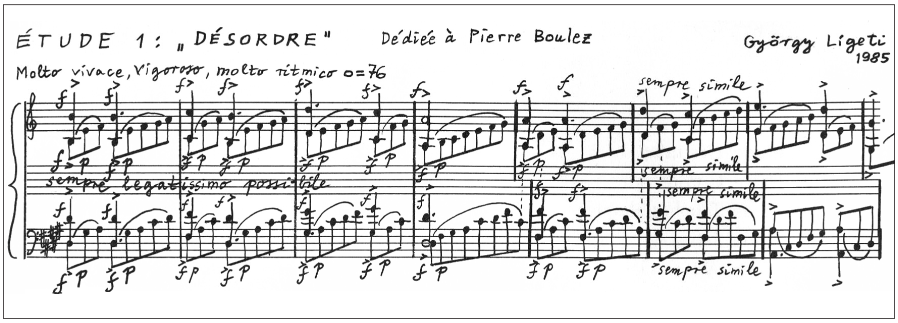

Figure 10. *Désordre*. First system of score © 1986 Schott Music GmbH & Co. KG, Mainz, Germany. Reproduced by permission. All rights reserved.

Conclusion

There has been (and still is) 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 introduced by the composers of the Second Viennese School of the 1920s and 1930s. Alongside the techniques of other music composed from the beginning of the 20th century onward, the serial principle itself is frequently considered to be the reason the music—so-called modern music, though now close to 100 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 audience

[w] White and black here refer to the color of the keys on the modern piano.

response. Only after further investigation and familiarization can deficiencies in the work be considered.[x]

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 that not applying it to composition would be, if not exactly an example of Luddism, then at least to risk missing important aesthetic developments that only the computer can facilitate, and that other artistic fields already take advantage of. That algorithmic thinking is present in Western composition for at least 1,000 years has been established. That such thinking should lend itself to formalization in software algorithms was inevitable.

However, Hiller's work and 1959 *Scientific American* article[17] led to much controversy and press attention. Hostility to his achievements[y] was such that the *Grove Dictionary of Music and Musicians*[z] did not include an article on it until shortly before his death in 1994. 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 the misguided bias that the computer, not the composer, composes the music. In the vast majority of cases where the composer is also the programmer, this is simply not true. As composer and computer musician Curtis Roads pointed out more than 15 years ago, it takes a good composer to design algorithms that result in music that captures the imagination.[34]

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 formalization of compositional process. Composition is, in fact, often slowed by the requirement that musical ideas be expressed and their characteristics encapsulated in a highly structured and non-musical general programming language. Learning the discipline of programming is itself a time-consuming and, for some composers, an insurmountable problem.

Perhaps counterintuitively, such formalization of personal composition technique allows the composer to proceed from concrete musical or abstract formal ideas into realms hitherto unimagined, sometimes impossible to achieve through any other means than computer software. 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] The computer can help composers overcome recreating what they already know by aiding more thorough investigations of the material, once procedures are programmed, modifications and manipulations are simpler than with pencil and paper. By "pressing buttons, introducing coordinates, and supervising the controls," to quote Xenakis again,[40] 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.

[x] To paraphrase Ludger Brümmer, from information theory we know that new information is perceived as chaotic or interesting but not expressive. New information must be structured before it can be understood, and, in the case of aesthetic experience, this structuring involves comparison to an ideal, or an established notion of beauty.[7]

[y] Concerning 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.[5]

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

References

1. Ames, C. Stylistic automata in Gradient. *Computer Music Journal 7*, 4 (1983), 45–56.
2. Assayag, G., Bloch, G., Chemillier, M., Cont, A., and Dubnov, S. OMax brothers: A dynamic topology of agents for improvization learning. In *Proceedings of the First ACM Workshop on Audio and Music Computing Multimedia* (Santa Barbara, CA). ACM Press, New York, 2006, 125–132.
3. Austin, L., Cage, J., and Hiller, L. An interview with John Cage and Lejaren Hiller. *Computer Music Journal 16*, 4 (1992), 15–29.
4. Bel, B. Migrating musical concepts: An overview of the Bol processor. *Computer Music Journal 22*, 2 (1998), 56–64.
5. Bewley, J. *Lejaren A. Hiller: Computer Music Pioneer.* Music Library Exhibit, University of Buffalo, 2004; http://library.buffalo.edu/libraries/units/music/exhibits/hillerexhibitsummary.pdf
6. Boulez, P. Schönberg est mort. *Score 6* (Feb. 1952), 18–22.
7. Brümmer, L. Using a digital synthesis language in composition. *Computer Music Journal 18*, 4 (1994), 35–46.
8. Chomsky, N. *Syntactic Structures.* Mouton, The Hague, 1957.
9. Christensen, E. *The Musical Timespace, a Theory of Music Listening.* Aalborg University Press, Aalborg, Denmark, 1996.
10. Cope, D. *Experiments in Musical Intelligence.* A-R Editions, Madison, WI, 1996.
11. Ebcioglu, K. An expert system for harmonizing four-part chorales. *Computer Music Journal 12*, 3 (1988), 43–51.
12. Edwards, M. *A Pure Data implementation of Ligeti's Désordre.* Open-source music software; http://www.michaeledwards.org/software/desordre.zip
13. Edwards, M. *slippery chicken: A Specialized Algorithmic Composition Program.* Unpublished object-oriented Common Lisp software; http://www.michael-edwards.org/slippery-chicken
14. Edwards, M. *Tramontana.* Sheet music, Sumtone, 2004; http://www.sumtone.com/work.php?workid=101
15. Eisen, C. and Keefe, S.P., Eds. *The Cambridge Mozart Encyclopedia.* Cambridge University Press, Cambridge, England, 2006.
16. The Electronic Music Foundation. *HPSCHD*; http://emfnstitute.emf.org/exhibits/hpschd.html
17. Hiller, L. Computer music. *Scientific American 201*, 6 (Dec. 1959), 109–120.
18. Holmes, T. *Electronic and Experimental Music.* Taylor & Francis Ltd, London, 2003.
19. Howat, R. Architecture as drama in late Schubert. In *Schubert Studies*, B. Newbould, Ed. Ashgate Press, London, 1998, 168–192.
20. Jordanous, A. and Smaill, A. Investigating the role of score following in automatic musical accompaniment. *Journal of New Music Research 38*, 2 (2009), 197–209.
21. Kinzler, H. and Ligeti, G. Decision and automatism in *Désordre* 1er étude, premier livre. *Interface, Journal of New Music Research 20*, 2 (1991), 89–124.
22. Kirchmeyer, H. On the historical construction of rationalistic music. *Die Reihe 8* (1962), 11–29.
23. Koenig, G.M. *Project 1*; http://home.planet.nl/gkoenig/indexe.htm
24. Koenig, G.M. Aesthetic integration of computer-composer scores. *Computer Music Journal 7*, 4 (1983), 27–32.
25. Kramer, J. The Fibonacci series in 20th century music. *Journal of Music Theory 17* (1973), 111–148.
26. Kunze, T. *Désordre* (unpublished article); http://www.fictive.com/t/pbl/1999 desordre/ligeti.html
27. Lendvai, E. *Bela Bartók: An Analysis of His Music.* Kahn & Averill, London, 1971.
28. Lerdahl, F. and Jackendorff, R. *A Generative Theory of Tonal Music.* MIT Press, Cambridge, MA, 1983.
29. Lewis, G. Too many notes: Computers, complexity, and culture in Voyager. *Leonardo Music Journal 10* (2000), 33–39.
30. Ligeti, G. Über form in der neuen musik. *Darmstädter Beiträge zur neuen Musik 10* (1966), 23–35.
31. Matossian, N. *Xenakis.* Kahn & Averill, London, 1986.
32. Norden, H. Proportions in music. *Fibonacci Quarterly 2*, 3 (1964), 219–222.
33. Prusinkiewicz, P. and Lindenmayer, A. *The Algorithmic Beauty of Plants.* Springer-Verlag, New York, 1990.
34. Roads, C. *The Computer Music Tutorial.* MIT Press, Cambridge, MA, 1996.
35. Ryan, D. and Lachenmann, H. Composer in interview: Helmut Lachenmann. *Tempo 210* (1999), 20–24.
36. Sowa, J. *A Machine to Compose Music: Instruction Manual for GENIAC.* Oliver Garfield Co., New Haven, CT, 1956.
37. Steinitz, R. Music, maths & chaos. *Musical Times 137*, 1837 (Mar. 1996), 14–20.
38. Supper, M. A few remarks on algorithmic composition. *Computer Music Journal 25*, 1 (2001), 48–53.
39. Winkler, G.E. *Hybrid II: Networks.* CD recording, 2003. sumtone cd1: stryngebite; http://www.sumtone.com/recording.php?id=17
40. Xenakis, I. *Formalized Music.* Pendragon, Hillsdale, NY, 1992.

Michael Edwards (michael.edwards@ed.ac.uk) is a Reader in Music Technology in the School of Arts, Culture and Environment of the University of Edinburgh, Edinburgh, U.K.