Chapter 1: Musical Algorithms as Tools, Languages and Partners: a perspective

Alex McLean, Roger T. Dean

1 Introduction

In this chapter we introduce the landscape of algorithmic music, and point to some of its burning issues and future possibilities. We also use the chapter to provide some guidance as to how we have organised the book, and where major topics are discussed: we summarise the structure in the next paragraph, and comments on individual articles in the book are made throughout this Chapter. The book has been arranged to provide contrasting views on core topics, on the one hand from theorists and analysts, and on the other, from practitioners. Happily for us, in many cases our authors pursue both types of involvement. But we have asked another group of authors (who contribute the Perspectives on Practice chapters) to foreground their own thoughts about algorithmic music and how they make it. At the same time we have encouraged all authors to give specific musical examples of what they discuss, and to feel free to mention their own work.

Our volume brings together a diverse range of authors to explore algorithmic music in the large. We engage with meta- and post-human perspectives – pointing to the question of what new musics are now being found through algorithmic means which humans could not otherwise have made. In reciprocation to this, we also explore cultural aspects -- how is algorithmic music being assimilated back into human culture, and what is its social function or meaning? Over the chapters we will gradually widen our scope, first grounding the topic and introducing its terms by exploring its artefacts, philosophies, and histories. We then survey the range (so far!) of technical approaches to composing algorithmic music, and the metaphors used that seek to install those approaches in human understanding. Then practical aspects are explored in some depth: the role of the algorithm as co-performer, and in supporting musical coordination between human performers. Finally we explore wider cultural aspects, such as the role of algorithmic music in society, education and commerce.

Perspectives on Practice sections (PoPs) are interspersed throughout the book as short interjections outside the main flow. But they provide prime value to the reader, connecting issues in the text with direct reflections on musical activity. PoPs provide introspection by an author on their own practice, as opposed to introduction to and analysis of the field provided by the other chapters. Authorship of chapters and PoPs divides roughly along the lines of researchers and practitioners, but not strictly; we include some practitioners who are independent researchers amongst chapter authors, and invite some respected researchers to reflect upon their practice. We have included ourselves amongst the PoPs in the form of a joint article at the end of the book: amongst other things this serves to indicate why we came together to catalyse this volume (supported by the encouragement and enthusiasm of our OUP Editor, Norm Hirschy).

This brief introduction may seem quite episodic, even sometimes temporarily taking surprising directions. But we hope it will sensitise readers to the wide range of topics to be addressed, and that after reading the book they will be left with both practical understanding of how algorithmic music is made, and of what makes this activity musical.

2 Background

An algorithm, essentially, is a finite sequence or structure of instructions, and we will elaborate on this terminologically in the next section. We note here that our emphasis is on algorithmic music-making, and on primarily digital computational approaches. The histories of manual and analog algorithms (mainly addressed in this chapter and in that by Collins), include experimental process music from the 60s in particular (by Brecht, Wolff, Glass, Stockhausen and others), in which a piece was constructed from a precise or vague process description. But they also include musical and musicological theory (from Ancient Greece to Hiller, Ligeti, Xenakis and beyond) and its algorithmic embodiments, including musical style modelling (Cope, Ebcioğlu and onwards).

Leigh Landy has elaborated the often-useful distinction between note-based and sound-based musics (Landy 2009). Note-based music involves a conception of discrete sequences of events, largely capable of being described in symbols (such as score notation), and most usually characterised in part by

pitches: such music permits tonal, harmonic and rhythmic hierarchies. On the other hand, sound-based music puts more emphasis on the spectral content of sounds, which may be slowly transforming, with relatively fewer discrete events and little emphasis on pitch (and usually no spectral hierarchy, but rather in depth spectral organisation). Sound-based music may also depend less on rhythmic pattern than note-based. Using this framework, we can summarise by saying that the book deals primarily with the process of creating tonal, post-tonal and sound-based music, and any other forms of music in which innovative and individual works can result from algorithmic approaches. We do not seek to provide anything more than pointers to the major algorithmic composers of the period up to about 1990, since our emphasis is on process, methods, ideas, developments. We put little focus on work directed towards recreating earlier styles. Similarly, we try to distinguish music which is mainly an overt fulfilment of the algorithm (which might be primarily of academic interest) from that which can be musically creative. Nierhaus (2009) mentions these issues in his book, and considers that most approaches he discerns are primarily about algorithmic 'imitation'. He covers this imitative/recreative aspect substantially and those concerning practical procedures. So we resurvey these aspects relatively briefly but adequately, and with a different perspective: essentially, whether and how any particular algorithmic approach might provide a path to music which is really new?

A related general question within or behind several chapters is: has there been any evaluation of the algorithm and of its products? We argue this needs to be done both by non-experts as well as by expert musicians, and that the field at large could usefully study features present in both the most and least favourably evaluated works. This relates to the question of the relevance of algorithmic generation methods to perception and cognition, and the argument that syntax is either not a relevant concept in much music; or instead, is a consistency created perhaps transiently, and in any case locally to a work.

We are now well placed in the development of algorithmic music to build upon the technical and evaluative issues to discuss cultural issues. There are a great many subcultures supporting the development of algorithmic music in diverse situations, from the close rituals of performance to the mass market activity of smartphone and tablet 'apps'. We therefore round the book off with a section rich with

viewpoints on the social function and cultural value of algorithmic music; where we use algorithms to reach for the post-human, how do cultures grow and adjust to bring the music back into the human realm?

3 Terminology and Current Usage

Defining a term such as algorithmic music is not straightforward. For one thing, it includes the word music, which stands for an unfathomable diversity of approaches, cultures, techniques, forms and activities. Alone, the word algorithm can be understood as a well-defined set of operations or rules, but this definition is not of great use in understanding algorithmic music. This is because the range of structures which can be considered algorithmic is extremely broad, encompassing all computer programs (or according to some definitions, all computer programs which eventually terminate), and perhaps even all musical scores. Indeed, we could say that all music making involves exploration of rules or procedures, applied, made or broken. Taken together though, the words algorithmic music stand for a rich field of activity, defined by the urge to explore and/or extend musical thinking through formalised abstractions. In the process of making music as (or if you prefer, via) algorithms, we express music through formal systems of notation, taking a view of music as the higher order interplay of ideas.

Over the past few decades, algorithmic music communities have formed around a number of continuing approaches, which we can arrange according to the relationship between human and algorithm. At one extreme lies the claim of the independently intelligent algorithm, in the form of computational agents which are deemed to be creative (see chapter by Wiggins and Forth, for example). At the other extreme, algorithms are treated more like musical notations, which humans work with and adapt as vehicles for their own creativity. In music, the field of live algorithms (see the chapter by Eldridge and Bown) is situated towards the former extreme, and that of live coding (cf. the chapter by Roberts and Wakefield) towards the latter. The independence of live algorithms allows them to be presented as non-human musicians, often as co-performers which incorporate machine listening in order to respond to human musicians, in live interaction. On the other hand, the live coding tradition does not give an algorithm such agency, but foregrounds the human authorship of algorithms as the fundamental musical activity at play. In live coding, the algorithm may run deterministically, but this determinism is

broken through live modifications by the human musician, who shapes the music through modifications to its code.

What the current traditions of live algorithms and live coding have in common is an emphasis on musical improvisation. However, it is important to note that algorithmic music practice extends far beyond improvisation and the performance of music. Indeed, throughout the early development of algorithmic music, real-time digital synthesis was challenging or even impossible, due to the lack of processing power in early computers. Accordingly, the algorithmic music heritage lies very much in music composition. Composers are often less visible than performers and improvisers, but we should bear in mind that the majority of algorithmic music making takes place in private. Indeed, algorithmic composition allows us to work with abstractions of musical time while not being subjected to the very real constraints imposed by a listening audience. So we recognise here that a core aim of many algorithmic composers is to produce works which are fixed and reproducible, but which may sometimes reach beyond the human imagination.

Levtov's chapter in this volume introduces alternative terminology from the perspective of an end-user who has purchased, or otherwise acquired, a musical algorithm to enjoy. In particular, he distinguishes generative algorithmic music which runs without user input, reactive algorithmic music which responds to environmental input, and interactive algorithmic music which end-users interact with directly to influence in order to influence the music. In these terms, live coding and live algorithms are both interactive uses of algorithms in music performance, where generative and reactive forms are generally listened to in a similar way to recorded music.

A recurring theme through several of the following chapters is of the affordance of algorithms. This again relates to the relationship between an algorithm and its user, and the opportunities for action that the algorithm provides or even suggests. For example affordance is core to Fiebrink and Caramiaux's chapter on machine learning algorithms, where their consideration of the musical activities suggested by a machine learning algorithms gives a practical perspective on algorithm design. Exploring the design of algorithms as a form of user interface in this way is a radical departure from the more standard purist conception of machine learning.

Issues of computational creativity, and of audience perception of the source of musical ideas in a piece are discussed in some depth. And the sociological and educational contexts in which algorithmic music is considered, for itself or incidentally, are also evaluated in the book: a diversity of attitudes continues to be present.

4 Origins

Ideas about what we now call algorithms can be found at least as early as 900 AD, and in many different cultures, from Arabic and Greek to Indian. Clearly the word has relationships to algebra, and there is a sense in which a contemporary piece of algorithmic music has access to the whole codification of mathematics, as well as programming languages. Nierhaus has again provided many useful perspectives on these technical aspects and their application (Nierhaus 2009).

We set the stage of algorithmic music with this chapter. There is then a fascinating description (Collins) of a series of machines that link the early ideas to contemporary algorithmic thought, by way of a range of automata. One of the striking things discussed in that chapter is an analysis by Riley of the 18th and 19th century 'overestimation' of the novelty of automata, notably musical ones (Riley 2009). It seems that automata have been treated in some cultures and periods with a kind of reverence, in others with the demonisation we tend to associate with Frankenstein’s monster, and in yet others with virtual indifference. Nevertheless, in the early days of computing in the middle 1950s, toy computers such as the Geniac were sold (as the name suggests) as almost magical machines capable of making music, however simple they really were musically, or in retrospect. Even when the Geniac's producers parted company amicably, one went on to make a similar machine sold as the Brainiac. Edwards (2011) puts this machine nicely into context in his article on the development of algorithmic music. Collins points out in his chapter the salutary concern that we may still be prone to aggrandise the potential of algorithmic music.

Most of the chapters in the book take off where Collins leaves the history, and elaborate on personal compositional and communal research trends. One aspect that is worth flagging here, also mentioned by Simoni, is that many works that will be discussed are not necessarily or generally

appreciated as being algorithmic. This, counterbalancing the risk of exaggerating the influence of algorithmic thinking in music, points out that in some respects we may tend to underestimate it.

5 Early Algorithmic music

We had intended that a chapter (to come immediately after this introduction) be devoted to introducing and discussing some ‘canonical’ algorithmic music, such as Hiller’s work, some minimal music of Reich, and some music by Ligeti; however, contractual difficulties precluded this. Consequently, we offer a brief general summary here, to link Collins' chapter to the more contemporary aspects of algorithmic music, reflected both in the personal PoP sections and in the research discussion chapters. Note again that the book is not intended to catalogue composers/improvisers and their musical outputs, but to address the context of ideas, processes, and developments. Indeed, reflection on this earlier idea of a ‘canon’ of algorithmic music made us question whether we are yet in a position to delineate such a canon, and even whether the idea itself is relevant. For example, the diffusion or more importantly understanding and analysis of algorithmic music from Africa is slight. Similarly, the role of women in algorithmic music is certainly underestimated, under-researched and possibly also under-developed, an issue raised in Simoni’s chapter in the preceding Oxford Handbook of Computer Music. As an example, consider some of the fascinating electronic and electroacoustic work by Daphne Oram, or Éliane Radigue : we know all too little of their use of algorithmic processes at present.

Several authors, including Collins, point out that there is an extremely long history attached to algorithmic music. Probably the earliest parts of this used algorithms which did terminate: if there was a goal, it could be achieved. Here we want to link this via the practices of the 1950s onwards to the present. The idea of an algorithm which can terminate (be fulfilled) remained relevant in some parts of the 1960s artistic turmoil ((Banes 1993)), as in some of the 'process' works, found both in music, dance, theatre, and text writing. Consider George Brecht, several of whose texts describing algorithms for (sound) events are entirely feasible, if sometimes exceedingly long. For example, his Drip Music (for single or multiple performance) requires that "A source of dripping water and an empty vessel are arranged so that the water falls into the vessel." A single performance with a finite source would clearly terminate; multiple

performances, or a performance with a natural source might continue to infinity. Likewise, Jackson Mac Low, a key interface between sound and text, has produced many significant text works using such algorithmic processes, and also provided process scores for music performers. Several of his diastics are intended for improvisatory interpretation by both musicians and text performers, usually with finite duration. Similarly LaMonte Young's *X for Henry Flynt* is a piece whose process is simple to initiate, but which may proceed for a finite time or essentially to infinity: a chosen event is to be repeated X times.

This raises the contrast between those composers who wished to initiate a process with no explicit termination condition and those expecting completion of their process. Stockhausen's Set Sail to the Sun is a text composition amongst the set From the Seven Days: each musician plays 'a tone for so long until you hear its individual vibrations', and then after listening to the others, has to 'slowly move your tone' until there is 'complete harmony' and all the sounds become 'pure, gently shimmering fire'. It is difficult to achieve this. Compare this with the procedures of Xenakis, whose computational algorithmic control of elements or complete works was more pragmatic, open to termination, and applicable within both generation of electroacoustic music in the studio, and scores for instrumental performers (Xenakis 1971, Harley 2004). On the other hand, Xenakis also routinely transformed some aspects of the outputs of his algorithms, both in the electroacoustic and instrumental domain, while still intending that their product is finite in duration.

Minimal music from the 1960s onwards, in the sense of the rhythmically repetitive work of Reich, Glass, Riley and others, is normally clearly algorithmic, though mostly manually composed and often providing a process whose completion can be identified: for example the progressive deviation and final return of two patterns which started in unison to the original state (as in Clapping Music, Piano Phase). Similarly from the pioneering works of US algorithmic music by Hiller, through those involving Laurie Spiegel, Max Mathews, James Tenney and Larry Polansky, there were often clear target states which terminate each algorithmic section. Particularly where improvisers were also involved, as sometimes with Spiegel, there were mechanisms for them to initiate an almost Schenkerian 'prolongation' of the algorithm and process. Later, post-minimalist ideas (such as those of William Duckworth) extended the range of applications of minimalist procedures, for example transforming pitch structures rather more,

and they have often been used computationally in more recent times. Ligeti, on the other hand, without using a computer, employed particularly rigorous algorithmic procedures on the pitch and rhythmic structures in some of his instrumental works such as *Continuum* for harpsichord (1968) and *Désordre* for piano (1985), permitting complex rhythmic juxtapositions and transformations. Michael Edwards provides an appealing and accessible introduction to the ideas of algorithmic music through the historical past, the works of Xenakis and Ligeti, and the computational approaches of Hiller, Koenig and later composers (Edwards 2011).

Much of the music mentioned so far has been intended for performance by instrumentalists (and vocalists in the case of Jackson Mac Low and some others). Some was still note-based, but exploiting synthetic sounds. Xenakis is probably the central figure in triggering the application of algorithmic processes to sound-based music, through the painstaking work in his crucial body of electro-acoustic works (Hoffmann 2002), and through his rigorous yet metaphoric and stimulating expositions in his book *Formalised Music* (Xenakis 1971). Most subsequent algorithmic musicians recognise a significant debt to him, and his work is duly assessed in depth in several previous books.

After Xenakis, we should mention the US League of Automatic Composers, and their outgrowths such as the network ensemble the Hub, and Voyager, George Lewis's fascinating co-improviser set of algorithms. One of the key ideas of the Hub was that their sonic material (and sometimes musical process per se) should be passed around their networked computers, sometimes emerging as a more complex shared process (Brown, Bischoff, and Perkis 1996). Lewis on the other hand was primarily concerned with his Voyager software as a partner for improvisers, single or multiple (Lewis 2009). Lewis foregrounded what we would now call machine listening, and the use by the algorithms of the attended information, and these topics emerge continually in the present book. Lewis himself also provides some typically sophisticated and challenging perspectives in his contribution here.

6 Issues in algorithmic music

Throughout the book, there is discussion of the utility of algorithmic processes both for offline composition (as mentioned, by this we mean composition in private) and for live performance (with an

audience in public). Amongst the key issues that unite these two aspects are: a) what benefit the algorithm, especially when computational (deterministic or stochastic), provides to the music creator; b) whether the algorithm can become a genuine partner in the creative process; c) whether it can make contributions which are equivalent in utility to those potentially emanating from another (human) music creator; and finally, d) whether it can provide meta-human outputs, which we ourselves currently could not achieve, but which may in the short to medium time frame become just as accessible cognitively and socially as musical outputs that we can make now. Later, such currently meta-human music may perhaps also become possible for a human creator to produce themselves (and hence no longer meta-human). Acceptability and utility (expressed in musical and social terms) are mutable aspects of any music genre, and of any innovation or retrospection, algorithmic music included. A key question is how to endow the algorithmic creation with the humanoid power of self-evaluation, and ultimately self-evaluation that can change in nature with time (see in particular the chapter by Wiggins and Forth).

At the process and functional levels of making music, there are other layers of issues. For example, consider electroacoustic music since c.1950, and particularly acousmatic music (in which there are no live acoustic instrument performers involved in the presentation of a piece, and usually not in its realisation and recording either). Here the previously quite distinct roles of composer and performer have been largely fused, and the level of control the initiating music creator can achieve is enhanced because it is not necessary to allocate elements of control to a separate performer. By extension, we can also observe algorithmic computational mechanisms taking over functions such as mixing, sound projection, sound spatialisation, all discussed to various degrees later in the book (particularly by Schacher). In other words, algorithms in principle may contribute to all stages of music making, to what historically has been a highly differentiated series of activities: composition, performance, acoustic spatialisation, recording, editing, mixing, mastering. As colleague Greg White puts it (White 2015), the overall process may tend towards 'maximal convergence' in the locale of control of the separable activities. Sometimes improvisation takes the place of the first two of that series.

We make no implication that algorithms should take over any of these steps, rather that they may do so or may contribute. To the degree that this occurs, the algorithms may be manually- or

automatically-driven. Perhaps the strongest appeal of the automatic application of such algorithms from a creative perspective is as part of an algorithmic collaborator, with whom a human creator performs (or composes). But of course from a commercial and practical perspective the use of those automatic processes may provide more economical and efficient outputs than the manual application, and hence contribute to commercial value, and to the accessibility of creative play in music.

This brings us to a critical question: how is algorithmic music appreciated and diffused? Simoni's chapter illustrates some of the main features of the uptake and perception of such music. For example, many listeners, even with musical training, are not particularly aware of the algorithmic contribution. This is probably encouraging from the perspective that is often raised, that without physical gesture on the part of a performer that is tied to sound generation, a musical event is lacking (even boring) and requires supplement. The supplement may be a display of live code (essentially, a display of the algorithm) or a complementary dynamic visual imagery sequence. Here the editors support a diversity of views within the range from a purist stance, that the code and algorithm are secondary and need not be overt in any way, to the view that we may celebrate the algorithm by making it in some way apparent, through to the idea that the algorithm in some sense is the work and should be appreciated in itself.

It seems that the main communities of algorithmic music are centred around the creators: such as the Live Algorithms in Music grouping, the Live Coding field, and its established TOPLAP community and new conference (initiated in 2015 by AM with Thor Magnusson), and several antecedent groups. Groupings of consumers of algorithmic music are sparse, with the possible exception of those who regularly participate in Algorave events (electronic dance music created by live coding and other algorithmic means) and precursors. The penetration to audiences of other varieties of algorithmic music, as illustrated by Simoni and other chapters seems to be largely as a subcomponent (overt or not) of composition and computer-interactive improvisation.

As Wiggins and Forth argue, we do not want algorithmic music to be evaluated by something akin to a Turing test, which simply asks whether an algorithmic piece seems to be plausibly a reasonable competent human creation. Rather, we want to allow for meta-human outputs, and for systems which develop their own evaluation frameworks, potentially novel. The listener may or may not transform their

perceptions of such music into a cognitive framework that corresponds to the algorithm's own methods, but in either case they may gradually assimilate the music into a meaningful whole. Similarly, a (human) co-performer working with a live algorithm can transform the prospective and retrospective meaning of a piece as a result of what they choose to play: the ultimate evaluation of such an algorithmic co-performer will always involve factors beyond those it uses itself. So it is perhaps fortuitously positive that much algorithmic music is simply assimilated in a context in which its nature and bounds are not transparent to most listeners (and sometimes, not to the creators either).

Nevertheless, we hope that educationalists' involvement in live music making including algorithmic music, will increase alongside the desirable (indeed inevitable) rise in computational literary throughout the world. The advent of the low cost Raspberry Pi computer has stimulated the diffusion of cheap and accessible computing machinery more widely around the world, and if this (ideally) eventually elicits a virtually universal basic literacy in programming, then algorithmic music can become accessible to almost everyone as both producer and consumer, since there will be minimal cost or cultural barriers.

7 Contemporary directions

We have taken care to ground this book in historical perspectives, particularly through Collins' chapter on the origins of algorithmic thinking in music. This grounding provides sure knowledge that there is nothing fundamentally new in the basic conception of algorithmic music, as we have known it for hundreds of years. However in terms of the activity of algorithmic music, everything is new; the speed of modern computation allowed by microprocessors, their plummeting cost, their proliferation in handheld devices, and social shifts too; free/open source culture, on-line social spaces, and in much of the world, an increasingly computer-literate populus. All this means that algorithmic music is now transforming from a niche activity, shared in fringe festivals and academic conferences, into a more inclusive music culture sometimes finding large audiences, end-users and communities of practice.

Linguists and computer scientists keenly point out that programming languages and natural languages are very different categories. Nonetheless, programming languages have always been designed for human use, and are now increasingly designed for human expression, supporting the rise of creative

coding as an actual career choice for many working in art and design fields. There are now many programming languages and environments designed specifically for the expression of algorithmic music and/or visual art, with the classic Music-N (e.g. C-Sound), Lisp (e.g. Symbolic Composer) and Patcher languages (e.g. Max/MSP, PureData) joined now by SuperCollider 3, Extempore, Gibber, Sonic Pi, Tidal and many more. Where refinements to programming language environments are designed for human expression, we argue that they become more like the written form of natural languages. In the following we pick out a few directions that this new expressivity is taking us.

The foray of algorithmic music into music education is well signposted in Andrew Brown's chapter. As he relates, there is a long history of bringing computational media into education, but it feels there is a surge of interest in creative computing that can bring all this research into new fruition. The recent success of the Sonic Pi environment, designed for teaching both music and computer science, as well as supporting music practice, is particularly encouraging. While the push for computer science education in schools may at times be motivated by economic and business interests, it also points to an exciting cultural experiment. What cultural shifts will algorithmic music take when our young programmers grow up, and computational literacy really takes hold? We are already seeing the growth of chiptune, a ‘retro’ digital music community celebrating early 8-bit computer sounds; is algorithmic folk music next?

The phrase “paradigm shift” has certainly been overused, but some do argue that major changes to how we think about computation and human creativity are about to take place. Bret Victor, responsible for the early user interface design of the iPad, now rejects contemporary notions of ‘touch’ interfaces, and even the notion of technology and design, instead reaching for computational media as a means to “think the unthinkable” (Victor 2013). Victor urges us to look beyond current practices of computer programming, towards a way of using computational representations to think through and communicate ideas. This will be familiar to many algorithmic musicians who compose music through a creative process of exploration through code, but Victor advocates finding far better representations for thinking about systems. This echoes the long expressed motivations of the visual programming community (Blackwell 1996), and indeed Victor draws much from the unconstrained early work from the 1960s and 1970s.

A recent development in algorithmic music has found large audiences at electronic music festivals largely outside the academic context of computer music. This could be attributed to “post-club” electronic music, which may literally be listened to after attending a nightclub, therefore taking the repetitive, timbre-focussed structures of dance music as a starting point for experiment. Autechre are a key example, generating their alien rhythms and sounds from procedures defined in software such as Max, with fans struggling to recreate patches from screenshots found in magazines. Another key example from the UK is Leafcutter John, leading club culture into unfamiliar territory through automatic remix tools and live algorithms. Elsewhere in Europe, the Viennese scene and in particular record label Mego became a strong centre for algorithmic noise and glitch, including the prolific audio/visual collective Farmers Manual (FM), who have released DVDs containing several days worth of recordings from live performances with their handmade software. In Denmark, Goodiepal has worked more explicitly in opposition to academia, developing a post-human approach of *Radical Computer Music*. More recently several artists have grouped together under the Death of Rave label, taking an often heavily process-based approach to taking apart dance music and amplifying its structure and sound to an extreme degree. The focus for all this work is the release and performance of music, and the production methods are rarely discussed, and often form only part of a range of techniques. However, Mark Fell’s work connects with this contemporary context in a multitude of ways, and he describes his approach to algorithmic music in generous detail in this volume.

For now though, and paradoxically, the excitement around algorithmic music is in how it is becoming everyday. Observing children build musical systems inside the hugely popular game Minecraft, and seeing the increasingly enthusiastic audience response to Algorave events, are amongst many experiences that indicate algorithmic music is beginning to enrich our lives in a multitude of possibly surprising, but fundamentally human ways. As often happens with technological shifts, from the invention of the piano to the harsh noises of the industrial revolution, the human response to mechanisation is to embrace it, as a jumping-off point for creating new means of human expression. Just as the often oppressive forces of industrialisation provided cultural ground and source material for

astonishing new musics, the perceived threat of software automation gives way to musical compositions which reach beyond what we could do with pen and paper (or even tape and scalpel) alone.

We should be careful however not to be seduced by the idea that the future of algorithmic music is in unimaginable complexity. Algorithms also afford simplicity, and current developments lead towards new algorithmic composition environments which are accessible to anyone with sufficient curiosity. More than anything, now is the time for algorithmic music to break from perceptions of difficulty. Yes, it gives access to a rich, unfathomable creative space, but the means of access -- the composition of words into code -- should be thrown open to all.

8 Conclusion

We have minimised the discussion of algorithmic techniques per se in this introduction. They will be detailed when appropriate later in the book, though the core concern is the ideas and musical achievements of the field. For the reader interested in details of some of the principal historic techniques, Nierhaus' book is valuable ((Nierhaus 2009). For a thorough survey and typology of techniques, with a particular emphasis on artificial intelligence, there is an extensive review (Fernández and Vico 2013). These two sources can be placed in a broad perspective by inspection of the time line of computer music history developed by Paul Doornbusch (Doornbusch 2009). This time line is maintained online at http://www.doornbusch.net/chronology/.

What we have tried to do here is to point to the many flavours of algorithmic music, and its wide ranging potential. We hope that the reader will find what follows illuminates these rather deeply.

Bibliography

Banes, S. 1993. *Greenwich Village 1963*. Durham, N.C.: Duke University Press.

Blackwell, Alan F. 1996. Metacognitive theories of visual programming: what do we think we are doing?

Paper read at In Proceedings of IEEE Symposium onVisual Languages, 1996. .

Boden, Margaret A, and Ernest A Edmonds. 2009. "What is generative art?" *Digital Creativity* no. 20 (1-2):21-46.

Brown, C., J. Bischoff, and T. Perkis. 1996. "Bringing digital music to life." *Computer Music J* no. 20/2:28-32.

Dean, R.T. 2003. *Hyperimprovisation: Computer Interactive Sound Improvisation; with CD-Rom*. Madison, WI: A-R Editions.

Doornbusch, P. 2009. "A chronology of computer music and related events." In *The Oxford Handbook of Computer Music,* edited by R.T. Dean, 557-584. New York, USA: Oxford University Press.

Edwards, M. 2011. "Algorithmic composition: computational thinking in music." *Communications of the ACM* no. 54 (7):58-67.

Fernández, Jose D, and Francisco Vico. 2013. "AI methods in algorithmic composition: A comprehensive survey." *Journal of Artificial Intelligence Research*:513-582.

Harley, J. 2004. *Xenakis: his life in music*: Routledge.

Hoffmann, Peter. 2002. "Towards an" automated art": Algorithmic processes in Xenakis' compositions." *Contemporary Music Review* no. 21 (2-3):121-131.

Ingold, Tim. 2010. "The textility of making." *Cambridge Journal of Economics* no. 34 (1):91-102.

Landy, Leigh. 2009. "Sound-based music 4 all." In *The Oxford Handbook of Computer Music*, edited by R.T. Dean, 518–535. New York, USA: Oxford University Press.

Lewis, George E. 2009. "Interactivity and improvisation." In *The Oxford Handbook of Computer Music*, edited by R.T. Dean, 457-466. New York: Oxford University Press.

Nierhaus, G. 2009. *Algorithmic composition: paradigms of automated music generation*. New York: Springer.

Pressing, J. 1988. "Improvisation : methods and models." In *Generative Processes in Music. The Psychology of Performance, Improvisation and Composition.*, edited by J. Sloboda, 298-345. Oxford: Clarendon Press.

Pressing, J. 1994. *Compositions for Improvisers : An Australian Perspective*. Melbourne: La Trobe University Press.

Riley, T. 2009. "Composing for the Machine." *European Romantic Review* no. 20 (3):367-379.

Smith, H., and R.T. Dean. 2009. "Practice-led research, research-led practice: towards the iterative cyclic web." In *Practice-led research, research-led practice in the creative arts*, edited by H. Smith and R.T. Dean, 1-38. Edinburgh: Edinburgh University Press.

Victor, Bret. 2013. "Media for thinking the unthinkable." *Beast of Burden* no. 4. http://worrydream.com/ -!/MediaForThinkingTheUnthinkable.

White, G. 2015. *Towards maximal convergence*, Newcastle, Australia.

Xenakis, Iannis. 1971. *Formalized Music*. Bloomington: Indiana University Press.