A MutaSynth in parameter space: interactive composition through evolution

PALLE DAHLSTEDT

Innovative Design, Chalmers University of Technology, SE-412 96 Göteborg, Sweden
E-mail: palle@design.chalmers.se

In this vision paper I will discuss a few questions concerning the use of generative processes in composition and automatic music creation. Why do I do it, and does it really work? I discuss the problems involved, focusing on the use of interactivity, and describe the use of interactive evolution as a way of introducing interactivity in composition. The installation *MutaSynth* is presented as an implementation of this idea.

1. INTRODUCTION

Through history, many musical works have been composed with the help of extra-musical structures and generative processes, such as formulas, chance, algorithms or hidden numerological messages, from Renaissance motets through serialism to the physics formulas of Xenakis (Xenakis 1971). I call these techniques formal methods, and I do not include the use of predefined musical forms, such as fugue or sonata form. Though they sometimes act as a creative limitation for the composer, I regard them more as templates to be filled with content and as rule sets to break in a moderate way, while formal methods are more about generating the content.

Why do composers use formal methods? There are several possible answers. It is certainly not true that it saves time and effort for the lazy composer, as anyone struggling with a computer program for months could confirm, spending even more time to understand the output. My experience is that it takes more time and work to create something with formal methods than with conventional composition techniques. So there must be other reasons to recommend them.

1.1. Creativity

Consider Mozart's piano sonatas. They form a set of ideas that we could think of as being inside a circular boundary. Many parts of the interior of this circle are shared among different sonatas, since the same ideas are reused several times. Other regions are unique to one work, because the ideas they represent are specific to a single sonata. Together they fill the whole circle, with no spots uncovered. From the beginning, when Mozart had only written a few sonatas, the circle was small. Every time he wrote a new one, the circle expanded, because a new sonata had to include some new ideas, or it would not have been interesting at all.

Imagine trying to construct a computer program that could imitate Mozart's sonatas. It must be based on the existing, real sonatas as a learning set. When composing in the style of Mozart, the program will take from the ideas in the circle. But it cannot add any new ideas. It will not make the circle any bigger, because there simply is no way to know what he would have invented, had he written one more sonata. We cannot extrapolate, since we do not know in which direction to go. The program, or rather the computerised musicologist, will never dare to do that, and it is also impossible per definition, since then it would not be Mozart anymore. This is a vicious circle (pardon the pun).

This expansion of the circle is, as I see it, the key to what creativity is, and also the key to the difficulties in automating it. I could stop here, being sad and disappointed, but I will not. My curiosity is too much challenged by at least trying to make some music with automated methods, although (un)consciously aware that it will always be music by me, not the program.

Still, it opens up the possibility for fascinating music that may not really be artificially created, but rather is an audification of a mathematical process, chosen by the composer. And the beauty of mathematics is undisputable.

Some of the reasons for doing it are:

- Formal methods are a tool for stepping out of the circle of what one has already created. A new work inside the circle, without expanding it, would be a mere exploitation of known ideas. To surprise myself, I need help from outside. It could be chance, a good teacher, beer, a sonata form or a formula.
- I think there is potential for music that would not have existed otherwise.
- I want to give myself a chance to observe the beauty of mathematics.
- The tools are there – do it! This is the rock climber answer, but is it not the responsibility of artists to investigate the possibilities of the available means?

A side-thought: Considering the importance of limitations, internal or external, in creative processes, I think using formal methods for some people is just another

Organised Sound **6**(2): 121–124 © 2001 Cambridge University Press. Printed in the United Kingdom.

way of giving up control, to submit oneself to something (maybe falsely) greater – the algorithm. Maybe to escape responsibility. This is dangerous, and may be a moral collapse. I am not sure, since sometimes something is coming out from it, and it is not hurting other people. It all depends on what comes out – the music.

Formal methods can be used on different levels in the compositional work. A plausible hierarchy in the compositional process could be:

- *Sound design*. Generating source material on the lowest level.
- *Material generation*. Processing the source material.
- *Structure generation*. Putting together the processed material.
- *Generation of large-scale form*. Assembling and shaping the parts of the piece.

The levels of sound design and material generation are certainly overlapping to a large extent, especially when working in the sound continuum, where one single sound could be material, structure or the whole piece. But sounds also have a structure, although more elusive.

In my experience, formal methods work best at the lower levels, for material and structure generation. This may be because the listener's expectations of coherence and conceptual consistency are so much greater at the higher levels. Material is nothing but material, and the way it is used will determine the quality of the music more than the material itself.

2. WHO IS THE COMPOSER? WHAT IS THE WORK OF ART?

When confronted with a large body of material, such as a MIDI file or a sound file coming from a program of mine, I get very mixed feelings. I have a slight feeling I did not write that music, and yet I am quite sure no one else did. I designed the algorithm, implemented it and chose the parameters, and still I feel alienated. Mentally I am just a consumer of the music, because I could not predict the results of my algorithms.

At this point, the composer has to stop for a while. If I just made a transcription (not a trivial task, I know) and published the music, what would that do to my circle of ideas? It would stay the same. In the view of the outside world (critics, audience) it would expand, since there would now be more ideas attributed to my name. But I, the composer, would have learnt nothing. Normally, formally generated musical material needs many rounds of filtering and arranging, of course, but this was a simplified example.

This feeling of alienation has to be eliminated, and the cure is to listen and listen again, until the material is assimilated by the mind, incorporated into my intuition, making my circle of imaginable music expand. This process of assimilation takes a lot of time, but it is absolutely necessary. Without it, I will not be able to assemble the material in a meaningful way, and more importantly, I will not have any moral right to put my name on it. Even if I do not change much in the generated structure before putting a title on it and publishing it, I have made it a part of me. I have become the composer by changing myself to accommodate to the result. Then I fulfil the two criteria of artistic authorship: I have created it, and it is a part of me. I must add that this does not imply that whatever result coming from a program is worthy of this attention. If the material is bad or simply boring, repeated listening will only make myself less sensible and more stupid. And the few ideas it contains would probably already be inside my circle, so it would not expand anyway.

When a piece is very much dependent on a certain algorithm, it is not clear what is the real work of art – the piece or the algorithm. If the algorithm is kept fairly secret and only the piece is published (in a broad sense), then I guess the piece is it. On the other hand, if the algorithm is published, or the composer uses it to produce several works or exposes it to the audience in interactive pieces, then maybe the algorithm really is the work of art.

What she has composed, in designing the algorithm, is really not one piece, but a parameter space or hyperspace of possible pieces. If the piece is interactive in some way, it can be regarded as a function of some input, such as a set of parameters, a chord or a theme, generating the output in the form of musical structures.

3. INTERACTIVITY

The parameter space of an algorithm can be huge, and there is no way for the composer to know all of it and predict what is going to happen for every single parameter set, but with clever design she can maximise the fraction of good results. Interactivity can be a way of exploring the potential of the algorithm. The interactive listener is helping the composer to explore the possible sound space, since his or her preferences will probably be very different from the composer's.

Human interaction is a great source of variation. In some ways it is much better than chance, so often used in music (also by me). Pure chance is blind and does not know when to slow down, zoom in or rush on. It does not create a feeling of participation, but it exhibits a nice random distribution. Human interaction, under favourable circumstances, will create a strong bond between the piece and the user/listener/explorer. She reacts in a nonlinear way to what she hears, in an interesting interplay between two very complex nonlinear functions, one's input being the other's output. By favourable circumstances, I mean when there is a correlation between

https://doi.org/10.1017/S1355771801002084 Published online by Cambridge University Press

input and output in such a way that the listener knows when she is affecting the music, in a repeatable way. It is not necessary to be able to predict what is going to be the response to a certain untested action – rather the opposite – but if roughly the same action is made again, a similar response should be expected.

Humans have brains, but are biased by personal predisposition which may prevent them from finding interesting points in space. Chance has no brain and exhibits a nice random distribution, free of bias. They should be able to work nicely together, complementing each other.

Interactivity can be implemented in different ways – everything from using a few sensors to trigger some notes to using advanced analyses of video streams to control sound synthesis. The listener's choice may be *when* to do something, *how* to do it, or both. The task is to find out *why*. Either may be regarded as an exploration of a parameter space.

If the sound space is too small, the listener will know it all after a short period of exploration and lose interest in the work. On the other hand, it will be difficult to navigate if it is too big. The paradox is that the more universal the algorithm, the bigger the space of possible results, and the lesser the good-sounding fraction of it. This means the interactivity will not work, since the listener will less probably get positive feedback and will stop exploring, except if there is a way of – when you have found an interesting point in sound space – holding on to that region.

I mentioned earlier that many current algorithms are limited to the level of material generation. In well-functioning interaction, interactive listeners will create their own large-scale form by the history of their explorations, or possibly a planned or revised recapitulation of those, when they know their way around sound space. This will make them co-creators of the piece.

4. INTERACTIVE EVOLUTION AND *MUTASYNTH*

One process that is definitely successful in exploring the space of possible music is history. By mass generation and selection by composers and audiences through time, fantastic pieces of music have emerged. The lifelong learning process of an individual composer could also be regarded as an evolutionary process, regulated by internal and external response to his works, mutated by arbitrary events. And just like its biological counterpart, the evolution of species, different surviving works are not better than each other, but different and continuously new. New not because it is necessary, but because the possibilities are so numerous, the chances are infinitely small that exactly the same combinations will ever appear again.

This metaphor of evolution of cultural items, called 'memes' – a word originally coined by Dawkins (1976) – has quickly grown to a research field – memetics. Also, the special case of evolution in music history has been investigated (Dennet 1999, Jan 2000).

Emulating this process is an excellent way of exploring a huge parameter space, such as a composition algorithm or a sound synthesis engine. In interactive evolution, where the listener selects the most fit individuals (i.e. sounds objects) based on his or her aesthetic preferences, interesting regions in parameter space can easily be located and explored. This was also an idea of Dawkins (1986), later used by Sims (1989) and others to generate beautiful graphics. An example of a musical application can be found in Johnson (1999).

Interactive evolution works like this:

(1) A random starting population is generated.
(2) The user listens to the individual sounds and selects his or her favourites.
(3) A new population is generated, based on the selected individuals, with some random variations.
(4) The process is repeated from step (2).

To let people try the process of interactive evolution on music, I have made the installation *MutaSynth*. It is based on a standalone program of mine with the same name, allowing users to evolve parameter sets for any MIDI controllable sound engine. This program is described in detail in Dahlstedt (2001).

The installation is based on collections of sound engines, called *styles*, each consisting of three sound engines, *parts*, generating different layers of the music such as drum pattern, bass line and melody. The sound engines are implemented on a Nord Modular synthesizer, which is a standalone virtual modular synthesizer, capable of a number of basic synthesis and triggering techniques.

Some styles are beat-based, techno-ish in nature, while others produce more amorphic, electroacoustic material. After a style set is selected, interactive evolution can be applied to either a single part or all three at the same time. The starting point for each part is either a random or a previously stored sound. Different genetic operators are applied to the sounds to generate variations on the parent sound.

Mutation will produce nine random variations on one parent sound. *Mating* will recombine the characteristics of two parent sounds, to form a set of nine children sounds. Everything is about repeatedly generating variations or combinations of one or two sounds and selecting the best ones. Sound examples of the different genetic operators (and other information about the project) can be found at http://www.design.chalmers.se/palle/mutasynth

When a population of nine new sounds is generated, the user can audition the individual sounds, one at a time. To simplify the evolutionary process, one can choose between evolving all parts at once or only one at

https://doi.org/10.1017/S1355771801002084 Published online by Cambridge University Press

a time. Inactive parts are muted. If plausible sounds are found, they can be stored in the gene bank for future use, or be selected for further breeding with the genetic operators. When nobody interacts, MutaSynth starts evolving sounds by itself, based on the stored sounds in the current style. The user can interrupt at any time and start breeding from there.

With this setup, widely different music can be created without knowing anything about the underlying sound engines or the different synthesis and triggering techniques used. Actually, the user is controlling about 360 synthesis and pattern generation parameters at the same time, without even knowing it.

Examples of what users have evolved can be heard in sound examples 1–3.

5. CONCLUSION AND FUTURE WORK

I made the installation of *MutaSynth* to show how interactive evolution can be used in creative processes and why it is especially well suited for sound synthesis. I also wanted to show how it offers an interesting approach to interactivity, really allowing the user to participate in the creative process. A way of constructing an interactive musical work is to present the audience with a sufficiently complex, well-balanced sound engine open for exploration in a consistent parameter space.

To conclude, returning to the discussion on automated music-making, I would like to mention a dream project I would like to realise at some time. It is partly based on the methods discussed in this paper and may be utopian, but I think it is at least in part possible to realise within the foreseeable future.

The first step is to design a flexible sound engine that could use existing sounds or parts of pieces as fitness criteria, measuring the degree of similarity in some way. The necessary shortcomings of the sound engine will make the result similar but different, and the differences may be perceived as an interpretation of the original material.

Based on this technique, combined with other learning algorithms, it could be possible to construct a program or machine that would constantly compose new music. It would learn from radio, from concerts and recordings, and adjust its compositions to what it hears. The program should also be able to re-evaluate its own works after a while, through the listening process. It would be real fun to bring this machine to a concert (without paying for its ticket) equipped with a microphone, and switch on the speaker in the foyer after the concert, playing a paraphrase.

This is not a small project, involving the design of automated, dynamic evaluation criteria and very flexible generation algorithms. It could be done either with audio synthesis or score generation, learning from sound or MIDI streams.

This is a dream project, because it is such an artistic and intellectual challenge. Someone may say it will never make good music. I say it does not exist until it makes good music. That is part of the goal.

REFERENCES

Clavia DMI AB. 1997. *Nord Modular* (a virtual modular synthesizer). Stockholm, Sweden. http://www.clavia.se

Dahlstedt, P. 2001. Creating and exploring the huge space called sound: interactive evolution as a composition tool. In *Proc. of Music without Walls, Music without Instruments Conf*. Leicester: De Montfort University.

Dawkins, R. 1976. *The Selfish Gene*. Oxford: Oxford University Press.

Dawkins, R. 1986. *The Blind Watchmaker*. Essex: Longman Scientific and Technical.

Dennet, D. C. 1999. The evolution of culture. *Edge Online Journal* **52**. http://www.edge.org/documents/archive/edge52.html

Jan, S. 2000. Replicating sonorities: towards a memetics of music. *Journal of Memetics* **4**.

Sims, K. 1991. Artificial evolution for computer graphics. *Computer Graphics* **25**: 319–28.

Xenakis, I. 1971. *Formalized Music: Thought and Mathematics in Composition*. Bloomington, IN: Indiana University Press.

SOUND EXAMPLES

1. Sound examples evolved by the audience at the 'Music without Walls?' conference in Leicester in June 2001. These textures, edited into a sequence by the author, are generated in the style *Dunk*, which is rather heavy down-beat in character (120 BPM).
2. Textures generated on the same occasion in the style *Tala*, which is based on repeated sequences of different lengths (132 BPM).
3. Textures generated on the same occasion in the style *Amorfik*, which consists of one slow, periodic part and two aperiodic, sound effects-like parts (72 BPM).

https://doi.org/10.1017/S1355771801002084 Published online by Cambridge University Press