Return to Index

Palle Dahlstedt
Innovative Design
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
+46-31-772-1000
palle@design.chalmers.se

Abstract

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.

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. It simply isn't true, 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 longer time and more work to create something with formal methods than with conventional composition techniques. So there must be something else to it.

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 conacing else to itsting e.D.ij   Tircle 0eiehm8snI 0.14336 Tw (bac 0.ntioheinside a circular) Tj   08es.

[illegible]

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 me, 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 wouldn't 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 isn't it a 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 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'm not sure, since sometimes something is coming out from it, and it isn't 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

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.

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 maximize 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 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 doesn't know when to slow down, zoom in or rush on. It doesn't 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 non-linear way to what she hears, in an interesting interplay between two very complex non-linear functions, one's input being the other's output. By favourable circumstances, I mean when there is a correlation between input and output in such a way that the listener knows when she is affecting the music, in a repeatable way. It is not

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 her aesthetic preferences, interesting regions in parameter space can easily be located and explored. This was also an idea of Dawkins (Dawkins 1986), later used by Sims (Sims 1989) and others to generate beautiful graphics. An example of a musical application is (Johnson 1999).

Interactive evolution works like this:

1. A random starting population is generated.
2. The user listens to the individual sounds and selects 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 a number of collections of sound engines, called *style sets*. One style set consists of four sound engines, *parts*, generating different layers of the music, such as drum pattern, bass line, chords 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 style sets are beat-based, technoish 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 four 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 the parent sound. The degree of variation can be selected. *Mating* will combine the characteristics of two parent sounds, to form a set of nine children sounds. *Morphing* generates linear interpolations between two parent sounds, i.e. the children will be on a straight line in parameter space, between the parents. This may sound complicated, but it is really quite simple to use – 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 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. This is made by pressing the corresponding number key, 1-9. To simplify the auditioning process, it is possible to mute individual parts. If the user finds sounds she likes, they can be stored in the gene bank for future use, or be selected for further breeding with the genetic operators.

With this set-up, widely different music can be created without knowing anything about the underlying sound engines or the different synthesis and triggering techniques used.

Conclusion and Future

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.

One 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 realize 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 realize within a 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 revaluate 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 doesn't 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 *Proceeding of Music Without Walls, Music Without Instruments Conference*. 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, 319-328.

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