ALTER EGO: A GENERATIVE MUSIC CREATION SYSTEM

Jan Trützschler von Falkenstein

University of Birmingham
Music Department
Edgbaston
Birmingham, B15 2TT, UK
info@sampleAndHold.org

Tom Tlalim

Independent Artist,
Composer and Researcher
The Hague,
The Netherlands
info@tomtlalim.com

ABSTRACT

AlterEgo is a generative music creation system based on Franciscco López's "Sonic Alter Ego". This system attempts to learn the preferences of the user. The application is currently under development, and a first prototype is ready to use and explore. The goal is that the system would be trained to play in a manner that reflects the preference of the composer and eventually sound very similar to its 'teacher's' composing style. To achieve this, a combination of Genetic Algorithms and Markov Chains has been implemented. The training of the application is done through simple interaction by rating the material, which has been proposed by the system. The system was initially written by commission from sound artist Francisco López [1] for his project "Sonic Alter Ego", which is one of the winners of the VIDA 10.0 awards for Art and Artificial life by Fundación Telefónica (Spain, 2007).

1. APPROACH AND GOAL

López asked us to implement an alter ego of himself, as he proposed in his original concept of a "Sonic Alter Ego" for the VIDA 10.0 award. in his own words:

"In music there is a traditional sharp distinction between tools (instruments, software, sound materials, methods) and pieces (composed, random, improvised). My concept of "Sonic Alter Ego" is something in between these two general categories. It is a generative system that contains tools and creates pieces. In that sense, it is less open than a tool and more open than a piece. My interest in developing such a system, in fact, does not arise from a need of compositional tools, but it is rather an experiment in human-machine integration in the realm of sonic creation. From my perspective, in this project what is usually called "composition" should be the result of the collaboration between a human being (myself)and a virtual machine (the "Sonic Alter Ego"). ..."

We started with a set of field recording, processed by López and a rough sketch of the system. The application consists of two modes: a training mode, in which the system learns the preferences of the user in audio processing and compositional decisions and a presentation mode, which acts as a generative music system. [2]

After analysing López's compositional approach we decided to start with a simple implementation of a generative system which would select a range inside a sound file, and play it with a volume envelope. To come closer to López's style of creating music compositions from manipulated field recordings [3], we began to introduce other parameters, including dynamic graphic equalization and panning. The goal at this stage was not to create a tool for music composition or a generative music system which resembles electro-acoustic style, but rather to try to create a musical clone of López. However, the current implementation of the system is open for further expansion, whereby it could be turned into a more general real-time composition tool or a trainable clone for composers in the future.

2. INTRODUCING ALTER EGO

The system has been realized in the SuperCollider programming language (SC), and can be run in two modes: A Training mode, and a Playback mode. The Training mode is meant for the development and training the algorithm, while the Playback mode can be used for presentation as an installation or in other formats. The application is built as a standalone, so that users who are not familiar with SC can interact with it without problems. It is built upon TeaTracks – a Multi Track Audio Sequencer [4], which is controlled by a set of Genetic Algorithms and First order Markov Chains. Similar techniques have been deployed before by others. [5] [6] [7] [8] Though most of the previous investigations in this field focuses on pitch and rhythm oriented music rather than music based on field recordings or musique concrète.

Using these algorithms, the system proposes sequences of audio material to the user, which he can rate, modify, and tweak by means of simple and direct feedback, while the system is running. Currently, the following parameters are generated by the artificial brain: Sound file transitions, order, timing, graphic equalizer amplitudes, fade in and out, amplitude, and panning. The sound material is chosen from a pool of sound files, which is provided by the user in the beginning of the session. The user can save and recall training sessions, allowing exploration of training methods.

3. OVERVIEW

The core of the evolutionary part in of system is a pool of Genomes. A Genome has a chromosome, which is a collection of parameters that define the playback parameters of a sound file. Each sound name is associated with a GenomePool, which is a group of potential Genomes for its playback. The Genomes in GenomePool are bred regularly by a Breeder, and then evaluated by the user while listening. When the user gives a positive feedback to a certain Genome, it is added to a separate pool called a LovePool. The Genomes in this pool are used as models in the fitness function. Genomes that are more similar to the LovePool have a higher chance to survive. There are thus two pools of Genomes; One for breeding and one for comparison while breeding.

Each sound file name is also associated with two Markov Chains, one of which determines the sound files that may be played in succession to the current, while the other determines the wait time between succeeding sound files. In the initial phase any sound file may be followed by any other. As the system runs and listens to feedback, it increases the likelihood of certain favorite sound file combinations to be chosen over others by reinforcing their link in the Markov Chain. This process of reinforcement of the Markov chain happens also for the wait times in between consequent sound files.

3.1. Implementation

The program has been written in SuperCollider, using an object oriented approach. The 'brain' of the application consists of a Genome class, a Breeder, a GenomePool and a Markov Chain. The playback and sequencer has been integrated into TeaTracks.

The implementation of the Genetic Algorithm is based on classes by Fredrik Olofsson, and we used Julian Rohrhuber's implementation of a first-order Markov Set as super class for our own Markov Chains.

The values held in the Genome currently include, Amplitude, Attack time, Release time, Fade curves, Graphic EQ Amps, Panning, Start time in the file and Duration of the segment of the sound file to be played. The Markov Chains hold all of the sound files in the session, and a list of numbers from 1 to 100 that expresses the duration of the sound file in percentage, at which the next file may be played. The Breeder object breeds the Genomes in the GenomePool. It does so by taking pairs of Genomes, rating each Genome's fitness in comparison to the a LovePool Genome, and then creating two new instances, by swapping a certain percentage of the items in their chromosome, and adding a percentage of random deviation to the values. The breeding process happens routinely in the background, and can be paused and continued by the user. The user can also control the breeding rate, crossover rate, mutation rate, and survival percentage.

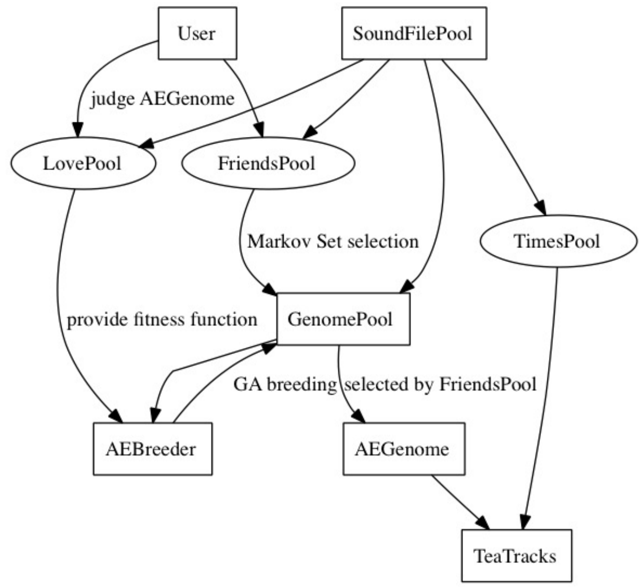

Figure 1. Overview of classes and their relationships to the User.

We tried to create a generative 'Composer', which functions as a primitive brain. It carries out a routine where it selects a sound file, associates it with a Genome and makes the transition to the following sound file and Genome, based on the best candidates from the GenomePool. The 'Composer' then starts a FilePlayer for the sound file with the parameters of the Genome. The player streams these files from disk with the proper sound parameters.

3.2. User Interface

The user interface consists of a MultiTracker view which shows the chosen sound files as they are being played, their times, current position of the playback head, and many other features such as play/stop zoom, load, save, etc. (figure 2).

By double-clicking on a sound object the user opens a window showing the Genome parameters (figure 3). The user can choose to Love, Skip, or Ban the selected Genome by clicking on the corresponding buttons, as well as tweak their parameters. Training and playback sessions may be saved to disc and then recalled, so the user can develop different personae or moods, using different training methods.

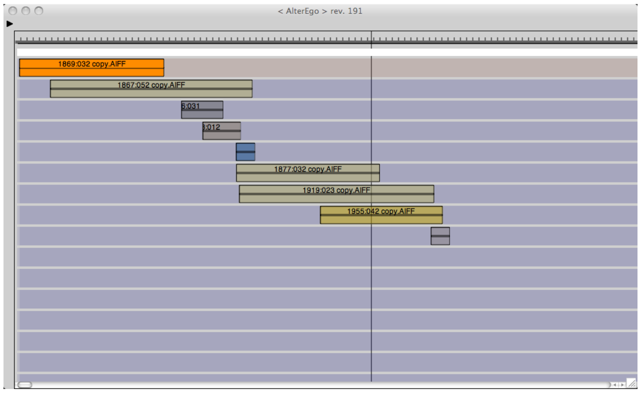

Figure 2. The Multi Tracker (TeaTracks) view.

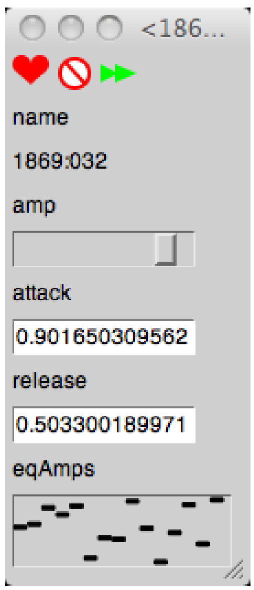

Figure 3. Genome feedback.

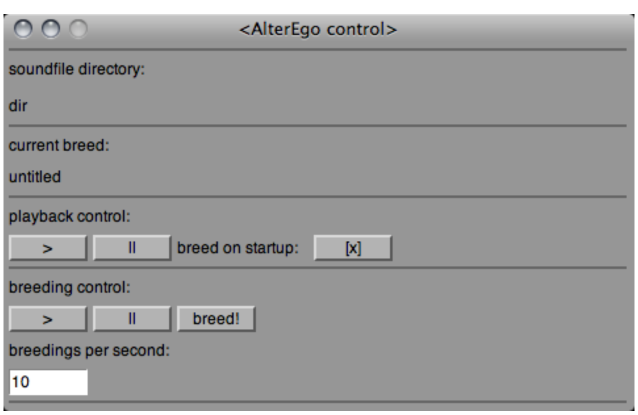

Figure 4. AlterEgo control panel.

The AlterEgo control GUI (figure 4) allows the user to control the overall playback and breeding properties, as well as breed the system manually.

4. EVALUATION

We have implemented an application which tries to adopt a certain style of composition by training. Though we haven't gone into quantitative analysis of the sonic results at this stage, evaluation during development and comments by López indicate that given appropriate sound files and training, the system would generate music which sounds very close to music by López. For proper evaluation, the results would have to be tested on people who are familiar with his style of music.

The choice to use Markov Sets for selecting sequences of sound files proved to be good. The layout of the sound files in time, which we also based on Markov Sets was a good starting point, but could definitely improved. We are currently looking into other solutions, which sets the timing in relation to other parameters, length and other processing parameters of preceding sound files. Another clear limitation of using a Markov Chain to determine time is being confined to a fixed list of times, which means that files are placed on a grid.

The approach of using various ranges and durations of sound file playback proved quite meaningful as it has a great impact on the complexity of musical results. However, in order to create a real musical Alter Ego some more advance sound processing must be integrated, and it might be worthwhile to look into alternative methods of introducing data which is similar to the tendency of the preference of the user in order to obtain more varied results.

5. FUTURE DEVELOPMENTS

As said earlier, extending the sound processing capabilities of the system is on top of our agenda for future developments. This would enhance the set of the parameters that Genomes hold, making the relationships between them, the playback functionality and the user's evaluation methods more complex. User interaction and training methods must be developed empirically based on user experience and feedback from composers such as López as well as others.

The Integration of Machine Listening possibilities (based on sonic analysis) has good potential in influencing crucial choices such as EQ settings, Panning, Volume as well as the selection of succeeding sound files. Such listening techniques may include statistical analysis of sonic activity over time, Object recognition, noisiness measurement and the like. Moreover, the implementation of timbre transformations envelopes that change over time could increase the liveliness of the outcome, bringing it closer to becoming a valid electro-acoustic composition system.

The integration of semantic associations (tagging) as well as a meta databases of the different pools could be a way to abstract the whole concept making it more universal.

We are also interested in how far the system can learn our own style of composing.

6. REFERENCES

[1] López, F. Pers. Comm. www.franciscolopez.net

[2] Collins, N.,"The Analysis of Generative Music Programs", in *Organised Sound*, Cambridge, UK, 2008

[3] López. F. "Profound Listening and Environmental Sound Matter", in *Audio Culture. Readings in Modern Music*, Cox, C & Warner, D. Eds., Continuum International Publishing Group, USA, 2005.

[4] D'Escriván, J, Trützschler, J. ' "TeaTracks: A Demonstration of an Auditory Display of a Multitrack Sound Sequencer" , in *Proceedings of the International Computer Music Conference*, Belfast, UK, 2008.

[5] Burton, A.R., Vladimirova, T. "Generation of Musical Sequences with Genetic Techniques", in *Computer Music Journal*, 23:4, pp. 59–73, Massachusetts, USA, 1999.

[6] McAuley, T., Hingston, P., Algorithmic Composition in Contrasting Music Styles, in *Proceedings of the International Computer Music Conference*, Singapore, 2003

[7] Biles, J. A. "GenJam in Perspective: A Tentative Taxonomy for GA Music and Art Systems", in *Leonardo*, Vol. 36, pp. 43-45, Massachusetts, USA, 2003.

[8] Collins, N. "INFNO: Generating Synth Pop Electronic Dance Music on Demand", in *Proceedings of the International Computer Music Conference*, Belfast, UK, 2008.