2008－MUS－76 (6)
2008／8／7

Ludo: A Collective Sound Sculpting Game Over The Network

Fabio Furlanete[†‡] Jonatas Manzoli[‡] Kenji Mase[§]
[†]Universidade Estadual de Londrina, Music and Theater Department
Campus Universitário, 86051-990 Londrina, BRAZIL
[‡]Universidade de Campinas, Interdisciplinary Center for Sound Communication
Cidade Universitária "Zeferino Vaz", 13083-970 Campinas, BRAZIL
[§]Nagoya University, Information Technology Center
1 Furo-cho, Chikusa-ku, 464-8601 Nagoya, JAPAN
ffurlanete@uel.br, jonatas@nics.unicamp.br, mase@nagoya-u.jp

ABSTRACT

This paper presents a network-based sound sculpting game. Game design strategies are employed to allow artistically meaningful interactions between autonomous agents and human players. We present our model for musical information representation, a multi-agent system to process this information and interact with human players and a set of game rules to produce a collective sound-sculpting experience. Preliminary results and the first prototype are presented.

1 INTRODUCTION

We present Ludo: a system for collective sound sculpting that uses digital game design as a model for musical interaction between artificial agents and human players over networks. Our work is aimed at the interaction rules and how they can be designed to allow artistically meaningful interactions without excessively constraining the creative autonomy of the players. These rules should be applicable both in the context of art education and performance. We believe that the know-how from the area of network game design is useful for the design of such rules.

We build a system in which artificial agents – autonomous or user controlled – enforce the rules, allow the teacher or the artist (both called designer henceforth) to interfere in the process and produce a richer interaction experience. The representation of sound information is a key to enable agents to exchange and evaluate the sound fragments to generate good sounds. We employ a spectral sound representation which agents can process easily in a similar conceptual style of human composers and performers in Electronic Music and Sonic Art.

In this paper we discuss the motivations and the background for our purpose and compare the approach of some recent works in network music interaction with contributions from the game design area. Also, we show our model for representation of musical information, and a multi-agent system to process this information during the interaction with human players. Finally, we present a testing set of game rules aimed to produce a collective sound-sculpting experience and the results obtained from our first prototypes.

2 BACKGROUND

The introduction of the computer as a platform for creative work made explicit the dynamics of interaction between the sound artist and its tools when it allowed the recurrent aural valuation and the reformulation of the work [4]. At the same time the artist could build his work based on experimentation with sound material and give to the performers the means to participate in this experimentation. Sometimes, it became difficult to differentiate composition from improvisation. The work's identity as a whole, being a product of only one creative mind, gives place to a work projected as a set of marks left over by a collective and continuous creative process. The man-machine interaction allowed the composer to confront automation and autonomy problems. In such context, the artist's role changed and his creative process began to move from the sound design to the compositional process design [8] or, as in our purpose, to the design of interactive processes.

Moreover, the rising of sound interactions over the network offered the possibility to explore processes that could allow the emergence of complex collective behaviors. This led us to the study the Theory of Self-organized Systems. In particular, the definition of self-organization made by Debrun [3] seems to fit perfectly with the interaction process that might happen in a successful collective musical improvisation. The idea of self-organization as paradigm for musical composition has been addressed by Manzolli [10].

In the same way that the concept of self-organization fits with a successful collective musical improvisation, it also fits with a great number of collective games – Debrun himself [3] made reference to soccer as an example of a self-organizing process. Demands raised by musical interaction described here are similar to those faced today by the game industry.

Given this association between games and musical interaction, it is possible to regard the designer's role as the same as a "game master" of Role Playing Games (RPG). He is an agent within an interactive process, but an agent with special powers: he should be able to create the game initial rules – space, time, levels of autonomy – invite other agents to play and, eventually, interfere in the development of the process according to his aesthetic orientation. But how can a designer define rules to control the sound material without destroying the autonomy of the players and freezing the possibilities of complex behaviour emergence?

3 Related Work

3.1 Collaborative music interfaces

An example of work in this direction is the "Public Sound Objects" [1], a system for Internet collaborative music performance with a server side synthesis engine using CLAM C++ library controlled by the users through a web-based graphical interface. Another project, the "Auracle" [11], is an interactive, distributed musical instrument that uses data from the analysis of the user's voice to control a sound synthesis engine through the Internet. It has been developed for the Java™ platform with the Jsyn and TransJam libraries.

Moreover, Blaine and Fels [2] developed also collaborative interfaces for musical interactions. They addressed issues such as balancing complexity and expressiveness in designing interfaces to allow the engagement of novices without exclude the experts.

However, while these projects are centered in the production of tools and interfaces for interaction they don't address the issues of the interaction process itself. We found studies about this topic in the digital game design filed.

3.2 The game design contribution

Game industry and researchers have worked for awhile in issues of interactive processes design, since the pioneer Multiple-User Dungeon (MUD) until the modern Massively Multi-player Online Role Playing Games (MMORPG). Interactions offered by those games are getting closer in level of complexity to that offered by musical improvisation. In MMORPG, the universe openness level and the number of possible actions offered to players is comparable to most sophisticated musical interaction processes. As defined previously, the designer should consider issues of memory, indetermination, context and delineation of time and space for beginning the process. He states initial game rules. He has tools to interfere in the process. But he cannot define, deterministically, the results from that process.

There exists, in the framework of collaborative games, research aiming to formulate strategies to handle complexity of game interactions. Jull [7] distinguished them as *Games of Emergence* and *Games of Progression*, where *a small number of rules that combine and yield large numbers of game variations, which the players then design strategies for dealing with.*

Manninen [9] develops a typology of interactions organized according to three variables: frequency (how often you could interact), range (how many choices were available), and significance (how much the choices really affected matters. He title this *Rich Interaction*, and it is easily transposed to context of musical interactions. Moreover, Zagal et al. [12] developed an ontological language for game analysis in Game Ontology Project (GOP), where they aimed to *describe the design space of games*. Their points are very close to ours: *'How can we understand interactivity in games?' 'How is game play regulated over the progress of a game?'*. Their ontology includes control devices and how they transmit information – a topic deeply explored in the musical interaction area – to semantic issues of gesture and role of space in interaction.

As Jull showed, he identified *emergent situations that were the result of many rules together, without having to resort to a list of the particular rules that caused these emergent situations; we called these, rules synergies* [7] .

4 SYSTEM DESIGN AND IMPLEMENTATION

To test these concepts from the area of game design in sound interaction we have build a multi-agent prototype system that implemented a set of game rules aimed to produce a collective sound-sculpting experience. Our main objectives for this system were:

1. Allow sound manipulation by a group of peers over the network.
2. Enforce the subjective and aesthetic evaluation of the manipulations made by the peers.
3. Allow a rich and distributed interaction process and yet give tools to the designer to

interfere on it.

The system is organized in 3 main modules: the multi-agent system (the game itself and its rules), the synthesis engine and the graphical interface for both the designer and the user controlled agents. It should be able to use different sound engines depending on the type of musical material desired for the game. Also, the system should be modular enough to use different GUI's depending on the kind of user interaction we want in the game. In this paper our focus is on the multi-agent system.

From the user's point-of-view there are 2 kinds of user:

- The Designer: by selecting the sound engine, he/she selects the sound material to the game and the variables used to describe these sounds. She/he defines values for the game ontology's variables and the capabilities of the different roles in the game and starts the game.
- The Player: chooses an interface agent (avatar) from a list previously defined by the designer, each one with different capabilities to interfere in the game's sound environment and chooses an entry point in the topography of the game's network.

From the implementation point of view, the multi agent system has 2 main components:

- Agent collection based on the JADE Framework [6]. It contains the non playing characters (NPC) of the game and avatars for human players. NPC's and human players have the same status in the game. Each kind of avatar represents a role that human player can play in the game with different capabilities to interfere in the musical stream. The NPC's should be able to generate, evaluate and process musical information.
- An agent ontology for communication of trajectories and the rules to create and transform them.

4.1 Describing musical ideas as trajectories

In this system we represent musical ideas as temporal trajectories in a parameter space where each axis corresponds to a variable associated to a qualitative property of sound that should be translated to sound synthesis parameters.

Let α be an individual musical process parameter. For $\forall \alpha_i, i \in \mathbb{Z}$ and $0 \leq \alpha_i \leq 1$. The vector

$$\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n] \quad (1)$$

defines the n-dimensional parameter space.

Each point in this space is a vector of parameters and the musical idea as a whole is the trajectory interpolating these points.

A trajectory is a tuple of time-tags denoted by

$$\Phi = (B_1, B_2, \ldots, B_m) \quad (2)$$

where m is the number of time-tags in a particular trajectory. A time-tag B_j is defined as $B_j = (t_i, \boldsymbol{a}_i)$, where t_i is a discrete time after t_0 , and $\boldsymbol{a}_i$ is a vector in the parameter space $\boldsymbol{\alpha}$.

4.2 Musical interactions with multi-agent systems

All interaction should happen by the production, evaluation and processing of the these trajectories. For such was designed a multi-agent system that models the exploration of regions of space through transforming and interpolation of trajectories. Motif variation strategies used in traditional musical composition are implemented in the agents behaviors for trajectory transforming. This approach with multi-agent systems was successfully used for rhythm patterns creation and processing by Gimenes et. al. [5].

4.2.1 Impact of a trajectory

Given the memory of the past interactions the agents can identify two kinds of trajectories:

- Stable trajectories: permanence in global context (long time memory) and strong connection with local context (recent memory).
- Variation trajectories: fracture in both global and local context.

As in traditional music composition, variation trajectories should be connected in global and/or local context established by stable trajectories to make sense. More different from the context the variation trajectory is and more stable that context is when the variation trajectory is played, greater its impact to the listener.

4.2.2 Choice of peer agent

When entering the game, after selecting some peers to connect, the player (artificial or not) can choose one of the trajectories played by his peers and qualify it as a variation or stable trajectory. The peers that had their trajectories chosen have their score increased as a good stable or variation trajectory generator. In such way, they will became more specialized during the game. High score peers are connected to other peers, increasing their range of action. Peers that aren't good variation generators nor good stable generators are disconnected. Some agents are more permissive in peers evaluation than others, given it's affection factor. The objective of the player is to became as influent as possible being connected to all players in the net. If the player lose all connections, the game is over.

4.2.3 Contextualization of the trajectory

If the player chooses a peer's trajectory as a variation one, he will progressively approximate this trajectory to the context given by stable ones. If the chosen trajectory was a stable, it will became the reference for a new variation trajectory as different as possible from the reference. Then the generated trajectory will be evaluated by her/his peers, and so on.

Trajectory evaluation has two components: the first is a set of preferences where each agent has a score to different areas of the parameter space. The second one is the measuring the Euclidean distance between two trajectories point by point. In case of two trajectories with different sizes, the smaller one will have its points interpolated. An agent trying to generate a variation trajectory will take as reference the nearest trajectory to the average (point by point) of the trajectories in the memory. Then it will produce the most distant trajectory in the space regarding the reference trajectory. An agent trying to generate a stable trajectory will search for the most distant trajectory from the mean and take it as reference. Then it will generate a sequence of trajectories that progressively approximate to the average accordingly its preferences.

4.2.4 Memory and time control

Attention and good memory is very important to the game. The only reference to the global context is through the memory of the players and a trajectory is never played twice. Better they can capture the trajectories being played and more they can remember from the global context, better trajectories they can generate. Because of this, each agent has it's own memory buffer and are designed with attention and memory erosion factors to mimic as much as possible the behavior of human players.

Time control is also very important. More time the context remains stable, greater the impact of a variation trajectory. The players will try to wait as much as possible to play a variation trajectory, but to not wait more then the other players and lose the opportunity. To mimic this behavior the agents have an anxiety factor that defines how much it can wait for the best moment.

Each trajectory played is put in the peer's short-time memory buffer and suffer some level of corruption given by the attention factor. Then it is transferred to the long-time memory buffer with tags indicating the owner of the idea and the time of generation. At each run the memory buffer suffers some corruption given by erosion factor. The longer the time of a trajectory in the buffer, the greater the corruption. Reference trajectory are chosen from the corrupted memory buffer.

4.3 Agent system implementation

The agent's initialization properties are denoted by:

$$Q_k(\beta, \delta, \gamma, \omega, \boldsymbol{f}) \qquad (3)$$

where $k=1,2,\ldots,o$ is the player number; β is the attention factor; δ is the memory erosion factor; γ is the affection factor; ω is the anxiety factor; $\boldsymbol{f}$ is the set of preferences and $0 \leq \beta, \delta, \gamma, \omega \leq 1$.

The state of a player agent in the game is denoted by:

$$P_{k,t}(s, \boldsymbol{r}, \boldsymbol{g}, \boldsymbol{m}, \Phi_{0,} \Phi_i, \Phi_l) \qquad (4)$$

where $k=1,2,\ldots$ is the player number and $t=0,1,\ldots$ is time; s is the player score; $\boldsymbol{r}$ is the peers list; $\boldsymbol{g}$ is the listening buffer; $\boldsymbol{m}$ is the memory buffer; Φ_0 is the reference trajectory (the chosen trajectory); Φ_i is the actual trajectory being played and Φ_l is the target of a trajectory transformation process.

Trajectories can be listened only locally by the human players by rendering trajectories played by the peers of an specific player. Different points on the network will produce different music. The only way to listen to all the music produced by the whole game is to became the most influent player and to be connected to all players in the game.

The system is implemented for the Java™ platform running in a Linux system. We are using the JADE Agents Framework to implement the agent system and the game ontology. For sound processing and synthesis we have developed our own library called Acusmata, also in Java™, based in the Loris C++ library [13] implementation of the Reassigned Bandwidth-Enhanced Additive Model [14].

5 SYSTEM'S TESTS

Two sets of tests were made with prototype implementations of two different parts of the system: the multi agent system and the user interface for trajectory manipulation.

5.1 The multi-agent system's test

A simple FM/AM synthesis engine was built to test the system. For this engine we selected 6 parameters with clear and direct relation with easily perceivable sound qualities: a) loudness, b) granulation frequency, c) granulation depth, d) brightness, e) spectral harmonicity and f) pitch. In the actual implementation the variation of trajectory's dimensions are independent. Because of this we will show the test's data reduced to a 2D representation of pitch over time. The tests were made with a small agent network using a random agent to start the interaction process. The agents were able to open communication with other agents, capture their trajectories, evaluate, chose a reference and generate a new trajectory after that reference. Figure 1 shows the pitch/time data of an example trajectory generated by an agent A and it's corrupted version in the short-time memory of agent B.

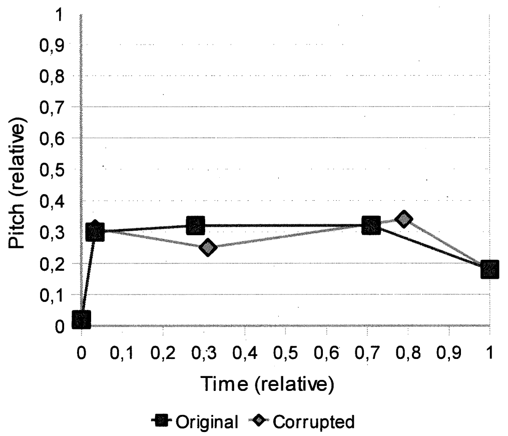

Figure 1: Original and corrupted trajectories

5.1.1. Memory and trajectory selection: memory capacity and corruption factors, in both short and long-time memories, were found to have a great impact on the selection of the reference trajectories for further trajectories selection and variation. As a general rule, a bigger memory with a small corruption factor seems to lock the reference trajectories in the center of the space as the interaction time passes becoming progressively less dependent of the context. A shorter long-time memory will make the reference trajectory change quickly accordingly to the context. A large corruption factor in the short-time memory gives to the reference random-like behavior.

5.1.2. Time control: the anxiety factor determines how much should an agent wait for the best moment to generate a new trajectory (a moment with low information density). It has impact on the global density of the sound texture generated by the network. Figure 2 shows the average density over the anxiety factor. The density values are in number of simultaneous trajectories being played for a network with 8 agents.

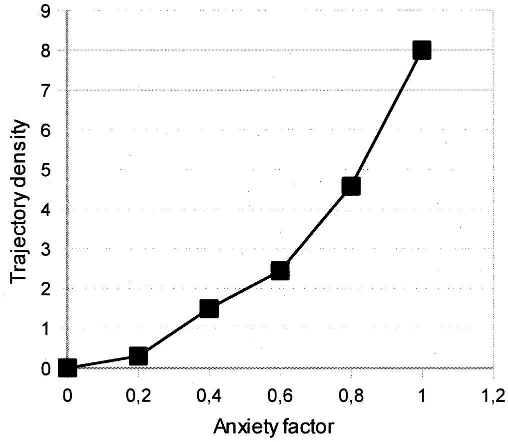

Figure 2: Impact of the anxiety factor

3. Trajectory transformation: in the first tests, four kinds of motif variation were adapted from traditional music: inversion, retrogradation, scaling (in both time and pitch) and interpolation. For the interpolation variation segments of the original trajectory where inserted at a random point generating a rhythm pattern effect. The variations were applied over the corrupted versions of the original trajectories in the short time memory. Figure 3 shows the four variations applied to the corrupted trajectory taken as reference.

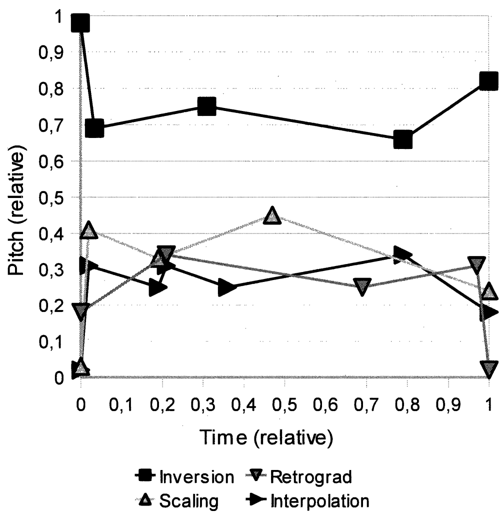

Figure 3: four variations over the reference

The aural evaluation of the agent interactions made clear the relation between the variation technique applied by the agent to the trajectories and the sound produced. The implementation of agents with more variation techniques should produce a more diverse and fluid sound stream.

5.2 The user interface's test

The interface for trajectory creation was given to public experimentation in a demo session during the Meidaisai Festival. It allowed Adults and children to play with trajectories over a parameter plane that controlled a frequency-domain sound morphing synthesis engine.

The test was satisfactory regarding the user interaction aspect. Also it showed some points that should be improved in the interface for a better interaction and integration with the multi-agent system.

6 CONCLUSIONS AND FURTHER WORK

From the 3 objectives previously stated we have already achieved the first one: our agents allow sound manipulation by a group of peers over the network.

For the second objective we have already implemented the tools for its achievement. The enforcement of subjective and aesthetic evaluation of the manipulations made by the peers is ceded in the agents and in the graphical interface, but its effectiveness needs more tests in real artistic and pedagogic situations. In the Londrina State University are groups of musicians and music teachers ready for start these tests both in the stage and in classes of public schools.

Although it is possible already for the designer to build the rules and the context for the interaction we doesn't have any graphical tool to help in this task yet. After the tests of the game we should focus on the development of this graphical interface for the designer.

Acknowledgments

Authors would like to thanks Brazilian agencies CAPES and CNPq for the financial support. Furlanete is supported by CAPES and Manzolli is supported by CNPq.

References

[1] Barbosa, A.; Kaltenbrunner, M. Public Sound Objects: a Shared Musical Space on the Web. In *Proceedings of Second International Conference on Web Delivering of Music*, pp. 9-16, 2002.

[2] Blaine, T.; Fels, S. Context of Collaborative Musical Experiences. In *Proceedings of New Interfaces for Musical Expression*, pp. 129-134, 2003

[3] Debrun, M. Auto organização. In *Auto organização: estudos interdisciplinares em filosofia, ciencias naturais e humanas, e artes*, Campinas: UNICAMP. Coleção CLE, vol. 18, pp. 3-8, 1996.

[4] Di Scipio, A. "Inseparable Models of Materials and of Musical Design", *Journal of New Music Research*, vol. 24, pp. 34-50, 1995.

[5] Gimenes, M.; Miranda, E. R.; Johnson, C. A. "A Memetic Approach to the Evolution of Rhythms in a Society of Software Agents", *Proceedings of 10th Brazilian Symposium on Computer Music*, Belo Horizonte, 2005.

[6] JADE: Java Agent Development Framework. <http://jade.tilab.com/>. Acessed in 01/15/2008.

[7] Jull, J. The Open and the Closed: Games of Emergence and Games of Progression. In *Proceedings of Computer Game and Digital Cultures Conference*, pp. 323-329, 2002.

[8] Laske, O. "Toward an Epistemology of Composition", *Interface*, vol. 20, pp. 235-269, 1991.

[9] Manninen, T. Towards Communicative, Collaborative and Constructive Multiplayer Games. In *Proceedings of Computer Game and Digital Cultures Conference*, pp. 155-169, 2002.

[10] Manzolli, J. Auto-organização: um paradigma composicional. In *Auto organização: estudos interdisciplinares em filosofia, ciencias naturais e humanas, e artes*, Campinas: UNICAMP. Coleção CLE, vol. 18, pp. 417-435, 1996.

[11] Ramakrishnan, C; Freeman, J.; Varnik, K. The Architecture of Auracle: Real-time Distributed Collaborative Instrument, In *Proceedings of the 2004 conference on New interfaces for musical expression*, pp. 100-103, 2004.

[12] Zagal, J. P.; Mateas, M.; Fernández-Vara, C.; Hochhalter, B.; Lichti, N. Towards an Ontological Language for Game Analysis. In *Proceedings of DiGRA 2005 Conference*, 2005.

[13] Loris Software for Sound Modeling, Morphing, and Manipulation. <http://www.cerlsoundgroup.org/Loris/>. Acessed in 01/15/2008.

[14] Fitz, K.; Lippold Haken, L.; Lefvert, S.; O'Donnel, M. Sound Morphing using Loris and the Reassigned Bandwdith-Enhanced Additive Sound Model: Practice and Applications. *International Computer Music Conference*. Gotenborg: Sweden, 2002.