A Sonic Eco-System of Self-Organising Musical Agents

Arne Eigenfeldt[1] and Philippe Pasquier[2]

[1] School for the Contemporary Arts, Simon Fraser University
Vancouver, Canada
[2] School for Interactive Arts and Technology, Simon Fraser University,
Surrey, Canada
{arne_e,pasquier)@sfu.ca

Abstract. We present a population of autonomous agents that exist within a sonic eco-system derived from real-time analysis of live audio. In this system, entitled Coming Together: Shoals, agents search for food consisting of CataRT unit analyses, which, when found, are consumed through granulation. Individual agents are initialised with random synthesis parameters, but communicate these parameters to agents in local neighborhoods. Agents form social networks, and converge their parameters within these networks, thereby creating unified grain streams. Separate gestures thus emerge through the self-organisation of the population.

Keywords: Sonic eco-system, Artificial-Life, self-organisation.

1 Introduction

Artificial-Life (A-Life), specifically the properties of self-organisation and emergence often found within it, offers composers new paradigms for computer music composition. The temporal nature of emergence — one desirable outcome of A-Life systems — provides a parallel to the complexity and evolution of gestural interactions sought by composers of both fixed and non-fixed music. Composers of generative computer music have found A-Life to be a particularly fruitful area of investigation. As McCormack [12] proposes, a successful evolutionary music system can "enable the creative exploration of generative computational phase-spaces."

1.1 Eco-Systems versus Evolution

Martins and Miranda point out that, while A-Life offers new possibilities for computer music composition, its algorithms must be adapted to suit its musical ends. Miranda's four requirements for evolution [15] are actually extremely difficult to achieve, particularly in a real-time context: his first criteria — selection of transformations — requires a system that can overcome the fitness bottleneck of interactive evaluation [2], a task that is difficult, if not impossible, given the aesthetic problem of evaluating successful musical evolution.

Bown [4] discusses some of the failures of the traditional A-Life paradigm of the interactive genetic algorithm, and points to new approaches based within social learning, cultural dynamics, and niche construction that offer potential solutions [13].

C. Di Chio et al. (Eds.): EvoApplications 2011, Part II, LNCS 6625, pp. 283–292, 2011.

Bown suggests that an eco-system approach "might generate sonic works that continue to develop and transform indefinitely, but with consistent structural and aesthetic properties". McCormack and Bown [13] posit that an eco-system approach emphasizes the design of interaction between its components, and that it is conceptualized within its medium itself: a sonic eco-system operates in the medium of sound rather than being a sonification of a process. RiverWave [13] is an example installation that demonstrates their concept of a sonic eco-system.

We have similarly chosen to avoid mating, reproduction, and selection within our system — all traditional properties of A-Life and evolutionary computing — instead focusing upon the complex interactions of an existing population within a sonic eco-system. Like Bown, we are interested in musical evolution and self-organisation of musical agents in real-time, in which the dynamic evolution is the composition [9]. Section 2 will discuss related work, including the use of musical agents, A-Life models for sound synthesis, and the uniqueness of our research; Section 3 will present a detailed description of the system; Section 4 will offer a conclusion and future research.

2 Related Work

We build upon the research into Artificial-Life based in audio by Jon McCormack [11], Joseph Nechvatal [19], Eduardo Miranda [15], Peter Beyls [1], Oliver Bown [4], and Tim Blackwell and Michael Young [3].

2.1 Musical Agents

The promise of agent-based composition in musical real-time interactive systems has been suggested [23, 18, 15], specifically in their potential for emulating human performer interaction. The authors' own research into multi-agent rhythm systems [6, 7, 8] has generated several successful performance systems. Agents have been defined as autonomous, social, reactive, and proactive [22], similar attributes required of performers in improvisation ensembles.

Martins and Miranda [10] describe an A-Life system in which users can explore rhythms developed in a collective performance environment. This system is an evolution of earlier work [16] in which agents could be physical robots or virtual entities whose data consisted of sung melodies. In this later research, agents are identical and remain virtual, although the number of agents can vary. Agents move in 2D space, but their interaction is limited to pairs exchanging data. Although the data is limited to rhythmic representations, the resulting transformations suggest a convergence similar to the system described in this paper; however, Martins and Miranda's motivation is extra-musical: "these transformations were inspired by the dynamical systems approach to study human bimanual coordination and is based on the notion that two coupled oscillators will converge to stability points at frequencies related by integer ratios." Agents build a repertoire of rhythms that will eventually represent a population's preference; however, these rhythms are not played collectively at any time, unlike the system described in this paper, in which a population's current state is immediately audible.

Martin and Miranda's conclusion points out problems with current approaches to musical composition with A-Life: "the act of composing music seldom involves an automated selective procedure towards an ideal outcome based on a set of definite fitness criteria." In this case, the creation of rhythms may have followed a evolutionary path that utilised complex social interactions, but the musical application of this data was a simple random selection. While the system demonstrates that "social pressure steers the development of musical conventions", the pressure is unmusical: it aims toward homogeneity.

Bown describes a system [5] which "animates the sounds of a man-made environment, establishing a digital wilderness into which these sounds are 'released' and allowed to evolve interdependent relationships". The analysed acoustic environment of the installation space, which includes the projected sounds of the agents, is treated as the virtual environment in which the agents exist. A continuous sonogram is made of the acoustic space, and agents attempt to inhibit low-energy regions, thereby creating an interesting dilemma: as the agents make sound at their occupied frequency region, they deplete the available resources at that region. It is the interaction of agents competing for available resources that creates the potential for emergence, and thus provides the musical evolution.

2.2 A-Life Models for Sound Synthesis

The system described in this paper utilizes granular synthesis as its method of sound generation, with agent interactions within the eco-system determining the synthesis parameters. Miranda's ChaosSynth [14] was one of the first systems to use models other than stochastic processes for parameter control of granular synthesis: in this case, cellular automata.

Blackwell and Young [3] have investigated the potential for swarms as models of for composition and synthesis, pointing to the similarity between the self-organisation of swarms, flocks, herds, and shoals and that of improvising musicians. The authors suggest that improvising music can explore musical aspects often ignored in accepted musical representations of "the note" — namely timbre and its morphology — aspects often shared by composers working with computers. The authors extend swarm organisation for synthesis in their Swarm Granulator, which, like ChaosSynth and the system described here, uses granular synthesis.

3 Description

3.1 Audio Analysis

We use a modified version of CataRT [21] as a real-time analysis method to generate segmentation and audio descriptors of live audio. Equally segmented units are stored in an FTM data structure [20] that has been extended to include a 24-band Bark analysis [24]. CataRT plots a 2-dimensional projection of the units in the descriptor space: we have chosen spectral flatness and spectral centroid (see Figure 1).

Incoming audio is initially recorded into a gated buffer in MaxMSP — silences and low amplitudes are not recorded — which is passed to CataRT every 12.1 seconds, resulting in 50 units of 242 ms in duration per analysis pass. When analysis is complete (at faster than real-time speed), the new units immediately populate the space. CataRT tags all units within an analysis pass with an index, or `SoundFile` number, which can be used to determine the most recent units.

We also use CataRT to query agent proximity to units, a function that returns a list of units within a defined radius (see Section 3.2.2).

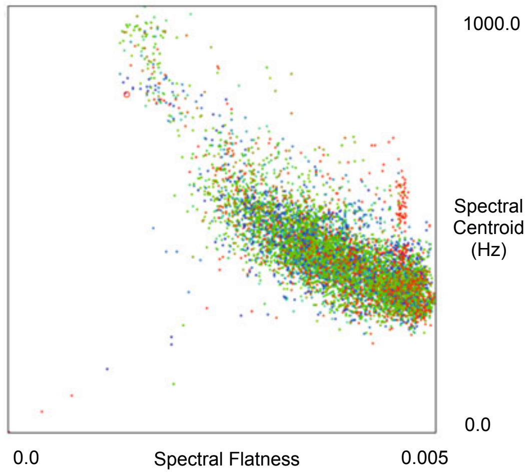

Fig. 1. CataRT display of units in 2D space, using spectral flatness (x) and spectral centroid (y)

3.2 Agents

A variable number of agents[1] exist within the space, moving freely within it, interacting with one another, and treating the CataRT units as food. When agents find food, they consume it by using its audio as source material for granular synthesis. How the agents move within the space, and how they play the audio once found, is dependent upon internal parameters that are set randomly during initialisation. As agents interact with one another, they share their parametric data, allowing for convergence over time (see Section 3.3). As the agents themselves converge in different areas within the space, their eventual parametric similarities, coupled with related spectra due to their locations, produces differentiated gestures.

3.2.1 Initialisation

When the system is initialised, agents select a random location within the space, a location which is immediately broadcast to the entire population. Agents store other agent locations, and calculate proximity to other agents after each movement step, or when finished playing a phrase. Those agents within a globally defined radius (`gNRadius`) of one another are considered to be within a neighborhood.

At initialisation, agents also select random values for their synthesis parameter ranges, from the following limits:

1. Grain duration (20-250 ms);
2. Delay between grains (5-350 ms);
3. Amplitude (0. - 1.);
4. Offset into the sample (0 to 1., where 1. is the duration of the sample less the current grain duration);
5. Phrase length (4 - 100 grains);
6. Pause between phrases (0 - 2500 ms);

---

[1] We have run as high as 48 agents on a single computer without obvious latency.

7. Phrase type (how subsequent grain delays are calculated within a phrase: stochastically using white noise, stochastically using brown noise, or exponential curves);
8. Output (number of speakers: for example, 1-8).

An agent broadcasts its parameters to the entire population if it is actively playing.

Lastly, agents select random values for their existence within the environment, including:

1. Acquiescence (the desire to stay with the same food, given a variety of nearby food sources);
2. Sociability (the desire to form social networks).

3.2.2 Movement

Agents are considered active if they have found food — the CataRT units — and are generating sound (see Section 3.2.4); only inactive agents move. Agents move at independent clock rates, calculating a time within their current pause-between-phrases range (initially two values randomly chosen between 0 and 2500 ms) from which a random selection is made. Agents move one grid location within the toroidal grid, then select a new delay time for their next movement phase.

Agents can see food outside their current grid location at a distance dependent upon a global radius: `gRadius`. When an agent sees food, it moves toward it. If no food is within view, the agent randomly selects a new location from its eight neighboring grid spaces, a selection negatively weighted by any locations the agent previously occupied. Agents keep track of their previous locations: if they have occupied a grid space in which no food was found, that location's desirability is decremented in the agent's location history. When selecting a new grid space, previously explored spaces have a lower likelihood of being selected. This allows for more efficient exploration of local space, with minimal backtracking (see Figure 2).

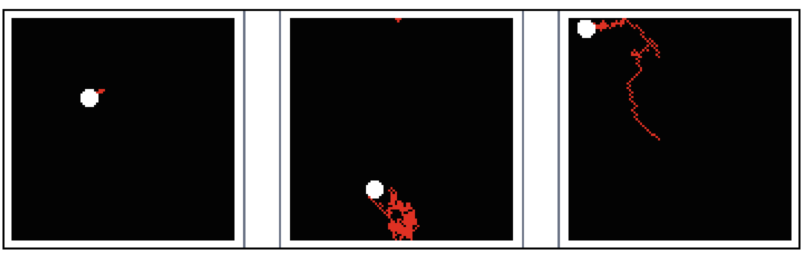

Fig. 2. Three agent's movements over time. The agent in the centre has found food after methodically exploring nearby space; the agent to the right is still looking for food; the agent to the left has recently been reincarnated, and thus has a short movement history.

When audio is recorded, the environment briefly becomes 3 dimensional, with the third dimension being time: the most recent `SoundFiles` appear at the top of the 3D cube. New audio analyses trigger an "excitement" mode, which allows the agents to move at much faster rates (using their grain delay times, rather than phrase delay) in search of new food. Any agent currently active will become inactive after its phrase is complete, and also move toward the new food. An analogy can be made of a fish-tank, with new food appearing at the surface of the tank, and the fish inside the tank swimming upwards toward it. The excited mode lasts an independent amount of time for each agent (exponentially weighted around a global variable `gExcitementTime`) after which the agents return to normal speed. The purpose of the excited mode is twofold: to allow non-active wandering agents to find food faster, and to force active agents to search for more recent audio. In terms of CataRT, during the excited mode, the most recent `SoundFile` number is included in the query.

When an agent finds food, it broadcasts this information to its neighbors. All inactive agents within that individual's neighborhood can then move toward the location in which food was discovered (see Figure 3). However, if an agent does not find food within a defined number of movement turns (a global variable that translates into a global strength, or constitution, of all agents in the population), the agent dies. When this happens, the agent broadcasts its impending death to the population, which remove that agent's data from their internal arrays. After a length of time (a global variable range between 5 and 60 seconds), the agent is reincarnated at a new location with new parameter data. The decision to use a model of reincarnation, as opposed to one of evolving generations, was made since the latter would offer little to this model – it is the complex interactions of the community over time that are of significance.[2]

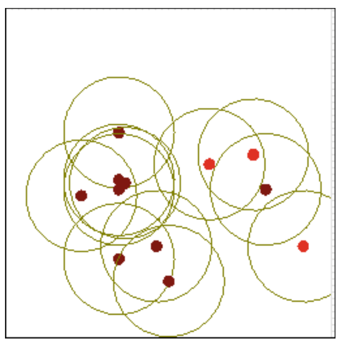

Fig. 3. Twelve agents within the space, with their individual neighborhood ranges visible. Dark agents are inactive, lighter agents are active. Inactive agents can move towards active agents within their neighborhood.

3.2.4 Sound Generation

Once an agent has moved with the CataRT's `gRadius` of a food source, the agent has access to both the unit's sample and analysis data. Using a custom granular synthesis patch created in MaxMSP, agents play a monophonic grain stream whose individual grain's amplitude, duration, delay between grains, and offset into the sample are determined by the agent's synthesis parameters. It was found that stochastic delays between grains did not significantly differentiate agent streams; therefore, over the course of an agent's phrase, the delays are chosen from the available delay range in one of three ways: white (evenly distributed throughout the

[2] On a more practical level, the number of agents cannot be dynamic within MaxMSP, as a set number need to be initialized before calculating the DSP chain.

range); brown (a random walk through the range); and curved (an exponential curve between range extremes). The specific probability distributions over the possible types are a converging parameter within the neighborhood (see Section 3.3.1).

The number of grains in a phrase is chosen from the agent's phrase range, while the length of time between grain phrases is chosen from the agent's pause range. A single agent's audio is rather meek, in that its amplitude is scaled by the number of agents within the eco-system. Interestingly, the stream itself somewhat resembles an insect-like sound due to its phrase length, discrete events, and monophonic nature[3].

The agent's spectral bandwidth is limited, as agents play their audio through a 24-band resonate filter, whose frequencies are set to those of the Bark analysis. The width of the filter is dependent upon the size of the agent's social network (see Section 3.3): agents outside a social network play the entire bandwidth, while agents within a network divide the bandwidth between them, selecting centre frequencies through negotiation (see Section 3.3). As the spectral bands of a unit are played, its Bark energy is lowered (at a rate dependent upon a globally set variable `gPersistence`); if a unit's bands are dissipated completely, the unit is removed from the environment. Agent's within small social networks therefore "use up" food faster, as their spectral range is wider; as such, an agent's fitness is dependent upon its ability to coexist with other agents.

3.3 Agent Interaction: Social Networks

Agents within a global radius of one another are considered to be within a neighborhood. Agents can be in multiple neighborhoods (see Figure 4).

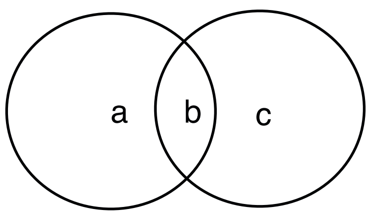

Fig. 4. Two neighborhoods, around agents a and c. Agent b is a member of both neighborhoods.

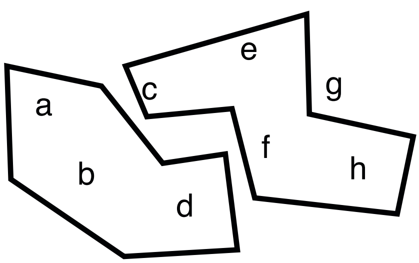

Fig. 5. Two social networks (a b d) and (c e f h). Agent g is not in a network.

[3] An example can be heard at `www.sfu.ca/~eigenfel/research.html`

After each movement cycle, or end of an audio phrase, agents can make a "friend-request" to another active agent in its neighborhood, in an effort to create a social network. The likelihood of this request being made is dependent upon the agent's sociability rating, while the target is dependent upon that agent's current social network status. If the agent has no friends in its own network, it will look for existing networks to join, favouring agents within larger networks. If an agent is already in a social network, it will look for agents outside of a network in an effort to make its own social network larger. Agents can choose to leave their network and join another, if the requesting agent's network is larger than the agent's own network.

Once agents have found food, and thus stopped moving, neighborhoods become static; however, social networks will continue to be dynamic, as agents can tempt each other away from existing networks (see Figure 5).

3.3.1 Convergence

Agents communicate their synthesis parameters within their social networks, and converge these parameters upon local medians. When an agent completes a grain stream phrase, it broadcasts which unit it consumed, along with the specific Bark bands. Next, it calculates the mean for all individual synthesis parameters it has accumulated from its social network, comparing it to its own parameters, and sets new values from the mean of the agent's previous values and the group mean.

Once values have converged for a parameter to a point that an agent's range is within 10% of the group mean range, for a period of time dependent upon a globally set variable `gDivergenceDelay`, divergence can occur, in which an agent can choose to select a new, random range. This alternation between convergence and divergence was used previously [9] in a multi-agent system to create continually varying, but related, group dynamics, and is an example of a heuristic decision that creates successful musical results. It can, however, be justified through biological models: agents can decide to leave a group if it becomes too crowded.

4 Conclusions and Future Work

Many different social networks emerge within a continuously running performance, aurally defined by their unified gestures due to their shared synthesis parameters and spectral properties. The musical goal of the eco-system is to move from independent individual granular streams into cohesive gestures, which depend upon similar spectra (arrived at through localization within the space) and similar synthesis parameters (arrived at through convergence).

Future work includes creating eco-systems across a high-speed network, in which agents act upon separate analysis created through different live audio. Related work includes sending audio generated by the eco-system across a high-speed network to an independent eco-system, complementing or replacing that eco-system's live audio.

Acknowledgments

This research was funded by a grant from the Canada Council for the Arts, and the National Sciences and Engineering Research Council of Canada.

References

1. Beyls, P.: Interaction and Self-Organisation in a Society of Musical Agents. In: Proceedings of ECAL 2007 Workshop on Music and Artificial Life, Lisbon (2007)
2. Biles, J.: Autonomous GenJam: Eliminating the Fitness Bottleneck by Eliminating Fitness. In: Proceedings of the 2001 Genetic and Evolutionary Computation Conference Workshop Program, San Francisco (2001)
3. Blackwell, T., Young, M.: Swarm Granulator. In: Raidl, G.R., Cagnoni, S., Branke, J., Corne, D.W., Drechsler, R., Jin, Y., Johnson, C.G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2004. LNCS, vol. 3005, pp. 399–408. Springer, Heidelberg (2004)
4. Bown, O.: A Framework for Eco-System-Based Generative Music. In: Proceedings of the SMC 2009, Porto, pp. 195–200 (2009)
5. Bown, O.: Eco-System Models for Real-time Generative Music: A Methodology and Framework. In: Proceedings of the ICMC 2009, Montreal, pp. 537–540 (2009)
6. Eigenfeldt, A.: Emergent Rhythms through Multi-agency in Max/MSP. In: Computer Music Modeling and Retrieval: Sense of Sounds, CMMR, pp. 368–379 (2008)
7. Eigenfeldt, A., Pasquier, P.: A Realtime Generative Music System using Autonomous Melody, Harmony, and Rhythm Agents. In: 12th Generative Art Conference Milan (2009)
8. Eigenfeldt, A.: The Evolution of Evolutionary Software: Intelligent Rhythm Generation in Kinetic Engine. In: Giacobini, M., Brabazon, A., Cagnoni, S., Di Caro, G.A., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Fink, A., Machado, P. (eds.) EvoWorkshops 2009. LNCS, vol. 5484, pp. 498–507. Springer, Heidelberg (2009)
9. Eigenfeldt, A.: Coming Together - Composition by Negotiation. In: Proceedings of ACM Multimedia, Firenze (2010)
10. Martins, J., Miranda, E.R.: Emergent rhythmic phrases in an A-Life environment. In: Proceedings of ECAL 2007 Workshop on Music and Artificial Life, Lisbon (2007)
11. McCormack, J.: Eden: An Evolutionary Sonic Ecosystem. In: Kelemen, J., Sosík, P. (eds.) ECAL 2001. LNCS (LNAI), vol. 2159, pp. 133–142. Springer, Heidelberg (2001)
12. McCormack, J.: Facing the Future: Evolutionary Possibilities for Human-Machine Creativity. In: The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music, pp. 417–451. Springer, Heidelberg (2008)
13. McCormack, J., Bown, O.: Life's what you make: Niche construction and evolutionary art. In: Giacobini, M., Brabazon, A., Cagnoni, S., Di Caro, G.A., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Fink, A., Machado, P. (eds.) EvoWorkshops 2009. LNCS, vol. 5484, pp. 528–537. Springer, Heidelberg (2009)
14. Miranda, E.R.: Granular synthesis of sounds by means of cellular automata. Leonardo 28(4), 297–300 (1995)
15. Miranda, E.R.: Evolutionary music: breaking new ground. In: Composing Music with Computers. Focal Press (2001)
16. Miranda, E.R.: At the Crossroads of Evolutionary Computation and Music. Evolutionary Computation 12(2), 137–158 (2004)
17. Miranda, E.R., Biles, A.: Evolutionary Computer Music. Springer, Heidelberg (2007)
18. Murray-Rust, D., Smaill, A., Edwards, M.: MAMA: An architecture for interactive musical agents. In: ECAI: European Conference on Artificial Intelligence, pp. 36–40 (2006)
19. Nechvatal, J.: Computer Virus Project 2.0, `http://www.eyewithwings.net/nechvatal/virus2/virus20.html` (accessed 6 October 2010)

20. Schnell, N., Borghesi, R., Schwarz, D., Bevilacqua, F., Müller, R.: FTM – Complex Data Structures for Max. In: Proceedings of the ICMC 2005, Barcelona (2005)
21. Schwarz, D.: Corpus-based concatenative synthesis. IEEE Signal Processing Magazine 24(2), 92–104 (2007)
22. Wooldridge, M.: An Introduction to multiagent systems. Wiley & Sons, Chichester (2009)
23. Wulfhorst, R., Flores, L., Alvares, L., Vicari, R.: A multiagent approach for musical interactive systems. In: Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 584–591. ACM Press, New York (2003)
24. Zwicker, E., Terhardt, E.: Analytical expressions for critical-band rate and critical bandwidth as a function of frequency. Journal of the Acoustical Society of America 68(5), 1523–1525 (1980)