# A Swarm Environment for Experimental Performance and Improvisation

Frank Mauceri and Stephen M. Majercik(✉)

Bowdoin College, Brunswick, ME 04011, USA
{fmauceri,smajerci}@bowdoin.edu

**Abstract.** This paper describes Swarm Performance and Improvisation (Swarm-PI), a real-time computer environment for music improvisation that uses swarm algorithms to control sound synthesis and to mediate interactions with a human performer. Swarm models are artificial, multi-agent systems where the organized movements of large groups are the result of simple, local rules between individuals. Swarms typically exhibit self-organization and emergent behavior. In Swarm-PI, multiple acoustic descriptors from a live audio feed generate parameters for an independent swarm among multiple swarms in the same space, and each swarm is used to synthesize a stream of sound using granular sampling. This environment demonstrates the effectiveness of using swarms to model human interactions typical to group improvisation and to generate organized patterns of synthesized sound.

**Keywords:** Interactive human/computer music improvisation · Swarm intelligence

## 1 Introduction

Real-time, interactive music generation systems often draw inspiration from the interactions of improvising performers. Group improvisation provides examples of spontaneous music generation involving player autonomy as well as reaction and responsiveness between players. Generative systems for music provide possibilities for complex, real-time, pattern generation analogous to improvisation. For the purpose of interactivity, these systems can be designed to accept human input in real-time. However, this is not a sufficient condition for creating interactivity analogous to group improvisation.

Interaction in group improvisation happens in several musical dimensions (e.g. changes of timbre, thematic material, dynamics, phrasing) and addresses different time frames (e.g. instant changes of texture or mood; short term motivic call and response; recapitulation of remembered materials). In artificial systems, the mapping between the human player and the computer needs to address multiple parameters and should have consequences over multiple time scales.

Audience recognition of the interactions among players is an essential element of the drama of performance. By witnessing interactions among players,


J. Correia et al. (Eds.): EvoMUSART 2017, LNCS 10198, pp. 190–200, 2017.
DOI: 10.1007/978-3-319-55750-2_13

the audience understands that decisions are made in the moment. Similarly, in artificial systems, the interface between player and computer should allow the audience to witness the exchange of information and discover its significance.

Swarm-PI is a real-time computer environment for music improvisation that uses swarm algorithms to control sound synthesis and to mediate interactions with a human performer. Other swarm-based interactive improvisation systems have been developed; we discuss some of these in Sect. 5. What distinguishes this environment from previous attempts to use swarms for interactive music systems is the introduction of a swarm avatar, i.e. a swarm that is analogous to acoustic features of the human performance. This swarm's parameters are continuously modulated by changes in the performer's sound. This avatar swarm, representing the performer, interacts with computer generated swarms.

In Sects. 2 and 3, we describe the goals that guided the design of Swarm-PI and discuss the nature of interactivity. We describe the details of Swarm-PI in Sect. 4. In Sect. 5, we discuss related work on swarm-based music systems. We discuss further work in Sect. 6, and we conclude in Sect. 7.

## 2 Design Goals

Our goal was to create an environment in which an improviser playing a conventional musical instrument could interact with a generative music system that responds to musical gestures, including changes in timbre. We constructed an interactive environment for improvisation to satisfy the following goals:

- The environment should facilitate the exploration of unfamiliar materials, relationships, and structures.
- The system should not presume any stylistic predisposition; the goal is not to simulate jazz (for example) or any other pre-existing music syntax.
- The observers (the performer and the audience) should be continually invited to anticipate the sequence of system states, although the system's complexity makes this sequence hard to predict.
- The performer's improvisational decisions and the environment must have meaningful consequences for each other.

## 3 Interactivity

Frequently, the creative work in a group of improvisors depends on interactions between performers who have independent, yet intersecting sets of artistic goals and strategies. In our environment, each swarm generates its own stream of sound, which develops according to the swarms unique rule set.

Improvisers' decisions often involve simple, immediate responses to one or more performers. In the course of playing together, organized behaviors emerge that are not the consequence of previously established global rules, but rather the consequence of local decisions. In this respect, the interactions of improvising performers resemble the behavior of swarm systems, where local interactions often result in surprising global patterns. For many performers, the emergence of unexpected patterns and relationships is a primary artistic goal. In improvised performance, the level of information is always in flux; artists constantly adjust their strategies in order to preserve the conditions of uncertainty and discovery.

In the introduction, it was noted that the audience's recognition of a responsive exchange between players was essential to the experience of interactivity. For that reason, a live musical performer was used for human input into Swarm-PI. The audience is thus given visual cues as well as auditory cues from the performer that are intended to facilitate the understanding of the audience. The performer exhibits listening and response in her comportment, gesture, and the expressive dimension of playing an instrument. Additionally, the authors intend that a display of the video representation of the swarms will be part of a performance, further facilitating the recognition of interaction between performer and computer. For example, an increase in the population size could be mirrored by an increase in the amplitude of the generated sound.

## 4 System Dynamics

The system dynamics of Swarm-PI arise from the interactions among a human avatar swarm and up to six computer generated swarms. A swarm's dynamics are not governed by a global controller or external force, but rather emerge from the interactions of the individual agents in the swarm (see Sect. 4.1). The self-organization and emergent behavior of the swarms make them appropriate for modeling creative production; each swarm is a virtual performer and its dynamics resemble the spontaneous autonomy of an improvised solo. The swarm can be described in terms of its location, density, velocity, and coherence; this allows for a nuanced mapping of swarm movements onto sound synthesis parameters. Swarms react to the location and movement of neighboring swarms, reflecting the exchange of information typical of group improvisation.

In Swarm-PI, the swarms interact with each other and are represented by real-time video animation in a 3-dimensional virtual space (Fig. 1). Audio from the human player is analyzed and acoustic descriptors are mapped onto a swarm avatar that interacts with the computer-generated swarms. The locations and movements of the computer-generated swarms are mapped to parameters for granular sound synthesis. The computer's reaction to the human player is mediated by the swarm representation of the performer and by the swarm rules governing the movement of swarm individuals. The performer is free to respond to the audio output of the system. Figure 2 provides an overview of the system in a configuration in which there is only one computer generated swarm. The Swarm-PI structure readily maps onto Blackwell's *PQf* architecture [5].

### 4.1 Swarm Movement

A swarm is composed of some number of virtual agents, typically between 50 and 100. The movement of individual agents in Swarm-PI is governed by rules

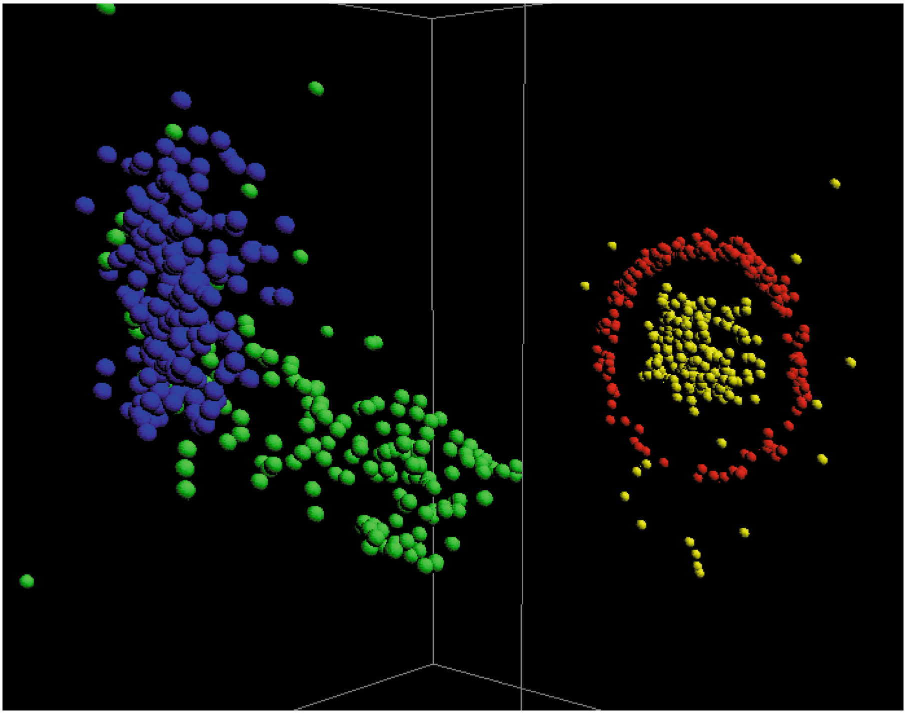

**Fig. 1.** Visual representation of swarms

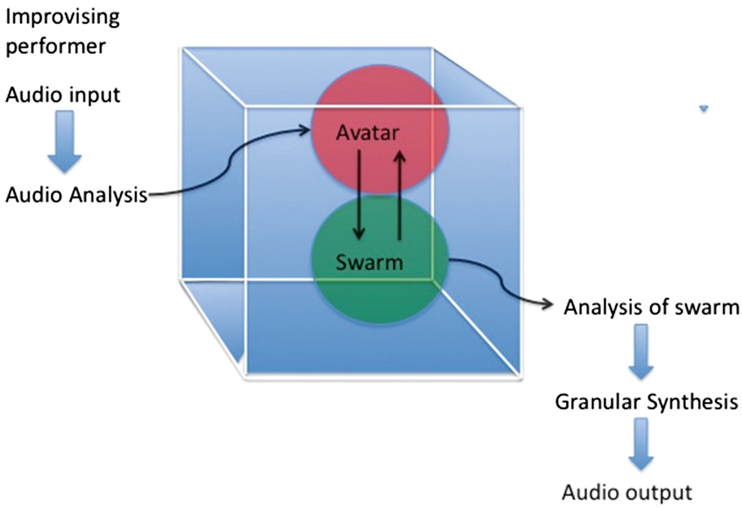


**Fig. 2.** Swarm-PI system overview

outlined in the classic bird flocking algorithm of Reynolds [10]. Individual agents move according to the location and velocity of other agents in a local neighborhood defined by a radius of perception. Three basic rules govern agent movements. An agent tries to match the average heading of its neighbors (*alignment*) and move toward the average location of its neighbors (*cohesion*), while avoiding collisions (*separation*). In our implementation, agents also have a *normal speed* and a *maximum speed*, and a degree of *random motion* can be introduced into their motions. With this basic model a wide variety of behaviors is observed. Swarms of agents can be coherent or chaotic; unexpected behaviors such as splitting and rejoining are possible. When more than one swarm is present, each with it own rules, a variety of interactions are possible including orbiting behaviors and predator/prey relationships. The swarm algorithm and video routines are based on Sayama's Swarm Chemistry [12] and written in Processing. The user interface and sound synthesis are implemented in Max/MSP.

### 4.2 Human Avatar

The human performer is introduced into the swarm space as a swarm avatar. An acoustical analysis of the performer (using Tristan Jehan's analyzer external for Max/MSP [8]) characterizes the sound of the player's improvisation. This analysis in turn defines parameters for the performer's swarm avatar. Like Rowe and Singer's stage work *A Flock of Words* [11], which uses acoustical analysis to control a swarm, the swarm avatar in Swarm-PI is not used directly for sound synthesis. It does, however, interact with the computer generated swarms, thus mediating between the human-improvised sound and the computer generated swarms, which generate their own sound (see Sect. 4.3).

Audio analysis yields values for amplitude, pitch (fundamental frequency), noisiness (spectral flatness), and brightness (spectral centroid). These values are mapped onto the parameters governing swarm behavior, in a way that is intended to confirm intuitive analogies between visuals and sound. For example, in Fig. 3, noisiness controls the cohesion and density of the swarm. Amplitude controls the creation of new swarm individuals and, thus, the size of the swarm. Pitch controls the location of the emitter (where new particles are created).

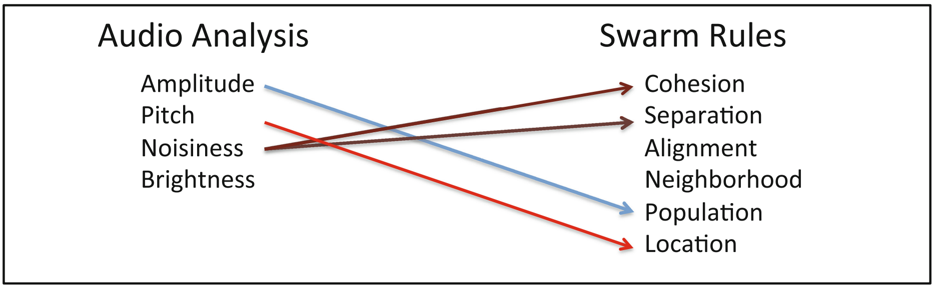


**Fig. 3.** Example mapping for swarm avatar

### 4.3 Sound Generation

The computer generated swarms, whose parameters are generated randomly, provide the basis for sound generation. It was noted in early experiments that these swarms settled into predictable patterns. Movements of the performer's avatar would perturb swarm movement and create interesting patterns, but additional changes in behavior were desired to prevent the repetition of these patterns over larger time scales. This was accomplished by cyclically modulating the swarm parameters using low frequency sine waves (e.g. 0.1 Hz).

Sonification of the swarms is accomplished via granulated audio sampling (using GMU Max externals developed at GMEM [1]). Sound granulation involves taking small audio samples (5–200 ms) and subjecting them to an amplitude envelope; constellations of these sound granules are assembled to synthesize complex timbres. Each sound grain is characterized by the location from which it is selected in a stored audio file, and by its envelope, duration, amplitude, playback speed, and pan position. The density of grain events is controlled by the inter-onset time between sound grains. Sequences of grains can be synchronous or asynchronous.

Parameter values for granular synthesis are obtained from an analysis of the swarm. Mean location, location variance, mean velocity, and variance of velocity are mapped onto the parameters of the granular synthesis engine. Again, mappings are chosen to make intuitive analogies between swarm behavior and changes in sound. For example, in Fig. 4, swarm density (location variance) correlates to granular density (inter-onset time and length), movement in the $x$-axis controls which sample is used and the pan position, movement in the $y$-axis affects transposition, and movement in the $z$-axis in conjunction with the population size affects amplitude.

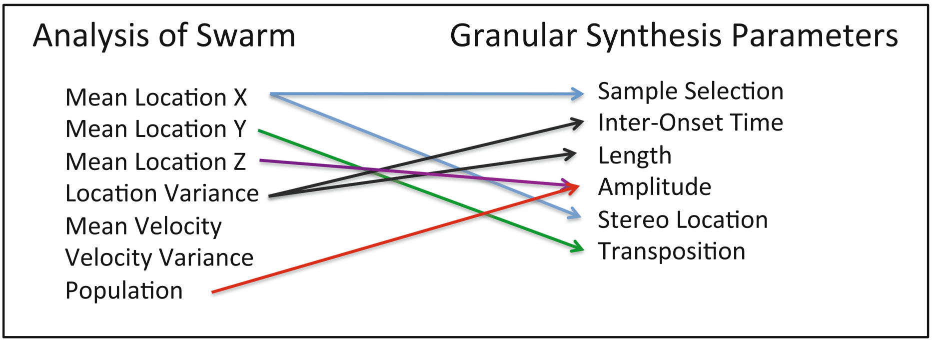


**Fig. 4.** Example mapping for granular synthesis

The parameter spaces of both the computer generated swarms and the synthesis technique are manifold, resulting in rich possibilities for mapping system dynamics onto sound. Initial performances used a live saxophone player and audio samplings from recordings of musical instruments for the granular synthesis. These performances demonstrated interesting interactions with the granulated sound generated by Swarm-PI. The performer found it engaging to control the swarm avatar and felt that the generated sound had a rich, complex, and continuously changing texture that was an interestingly stark contrast to his saxophone. He felt, however, that the system was slow to react to sudden changes or fast sequences of notes, and that the timbre of his saxophone needed to be better integrated into the ensemble of swarm improvisers. Finally, we felt that changes in the sound produced did not capture the vivid changes of organization displayed by swarms. Finding satisfying parameter mappings continues to be a challenge.

## 5 Related Work

Similarities between the dynamic of improvising musicians and that of swarms has inspired the creation of a number of swarm-based music generation systems. Three of these are Blackwell's: Swarm Music [4], Swarm Granulator [3], and Swarm Techtiles [6]. In Swarm Music, multiple swarms of particles move in a virtual 3-dimensional space, using to Reynolds' flocking algorithm [10], described in Sect. 4.1. Music from human performers influences swarm behavior by creating targets within the environment. Swarm particles are mapped into a virtual 3-dimensional music space, the axes of which represent volume, pitch, and pulse, thus providing a straightforward mapping from swarm to sound. Swarm Granulator is similar to Swarm Music, but operates at the sound grain level rather than the note level. Swarm Techtiles uses an entirely different approach. Audio samples from two channels are scaled to pixel values and the two streams are used to create a two-dimensional image with texture. Swarms search for areas of high texture, producing a granulated sound based on the image and leaving behind attractors to attract other swarms.

Swarm-PI is distinguished from Blackwell's systems in a number of respects. Blackwell's Swarm Music and Swarm Granulator both represent audio input as an attractor. Swarm-PI visualizes audio input as another swarm. In Swarm-PI, the human performer is therefore represented by this swarm avatar in the swarm space. In Blackwell's Swarm Granulator framework, the $x$, $y$, and $z$ coordinates of the attractor location are mapped onto three acoustic parameters (pitch, amplitude, and spectral centroid). The avatar in Swarm-PI maps acoustic descriptors onto a larger set of swarm parameters (e.g., pitch, amplitude, spectral centroid, and spectral variance). The avatar is more than a visual representation of the performers sound; the avatar swarm exhibits its own swarm behavior and continually interacts with other, sound-generating swarms. These sound-generating swarms and the avatar together, visually represent the interactive ensemble; the avatar preserves the metaphor of an ensemble of equals, improvising together.

Another difference between Blackwell's approach and Swarm-PI is the method for translating swarms into granulated audio. Blackwell's Swarm Granulator and Swarm Techtiles use the position of individuals in the swarm for the calculation of sound grains. Swarm-PI uses a statistical analysis of the swarm to control parameters for sound granulation. This method sacrifices the specificity of representing swarm individuals, but is less computationally expensive and more robust in real-time applications.

Two other swarm-based systems, Jones' AtomSwarm [9] and an unnamed system by Beyls [2], are not currently configured for interaction with human performers. AtomSwarm employs a flocking mechanism similar to that of Reynolds, a system of virtual hormones that affects the behavior of an individual, and a genetic component that regulates the hormone mechanism and modifies the musical behavior of an individual. The cyclic modulation of swarm rules in Swarm-PI is similar to the hormonal cycles in AtomSwarm, both serving to promote more varied behavior.

In Beyls' swarm music system, swarms operate in a 2-dimensional space associated with a MIDI player. The behavior of swarm individuals is governed by various factors, including position, energy level, level of attraction/repulsion with respect to other individuals, an activation level, and a personality. Movement is controlled by a stress factor that is similar to the cohesion/separation factors in a flocking swarm. Individuals form clusters; sound is produced by the first agent in the cluster with the most energy, its personality determining the melody's pitch-intervals, durations, and velocities. The other agents in the cluster add to the melody only if their activation level is high enough, whereas all swarm individuals in Swarm-PI contribute to the sound produced.

## 6 Further Work

Swarm-PI realizes an interactive computer system inspired by group improvisation. One of the most attractive features of this system is the performer's ability to function as if improvising in a group situation, playing an instrument and interacting with the system using only musical gestures and cues. There are many directions for further work; we describe four that we are currently pursuing.

### 6.1 The "Edge of Creativity"

In improvised performance, the artist is constantly adjusting her strategies to try to maintain a balance between uncertainty and creation [7]. There must be enough uncertainty so that there is "room for" new musical ideas, but not so much uncertainty that nothing created can be sustained.

Complex dynamical systems, such as swarms, have become an influential concept in psychology and cognitive science, especially with reference to creativity [13,14]. Such systems exhibit three types of behavior that are analogous to different aspects of creativity: (1) self-organization, a phenomenon in which the elements of the system become organized or structured in a seemingly purposeful way despite the absence of a global controller, (2) emergent behavior, i.e. behavior that is not predictable from knowledge of the system's constituent parts, but instead arises from the operation of the system, and (3) phase changes, which are relatively abrupt transitions from one behavioral regime to another very different regime. These three characteristics suggest analogies to the organization of ideas into new patterns in the creative process (self-organization), the novel and unexpected qualities of creative ideas and artifacts (emergent behavior), and the nonlinear quality of the creative process (phase changes).

In particular, phase changes provide a way to model the efforts of the performer to maintain the balance between uncertainty and creation. In the computer generated swarms in Swarm-PI, some parameter values result in a fixed state, e.g. all the swarm members are trapped in a tiny sphere centered at a fixed point. Even if that point were moving, the behavior would, in a sense, still be fixed; only the center of the sphere is moving. Changing the parameter values, even smoothly, can transform a fixed state into an apparently random state, in which swarm members are evenly distributed in the virtual space.

In between these two regimes, however, it is possible to find regimes of behavior in which swarm members form structures, such as rings (see Fig. 1), which are stable for a period of time, but then dissolve and re-form as some other type of structure, such as a sphere. Two structures can collide and produce a new structure. There is both uncertainty and creation. There is no formula that one can use to produce this regime, but information flow appears to play a key role. There needs to be sufficient information flow—swarm members need to know what other swarm members are doing—to prevent fixed behavior and enable structure formation, but not so much information flow that the structures created are destroyed by new structures as soon as they form. Our hypothesis is that an analysis of the information flow in the swarm will allow us to determine the conditions necessary to produce and maintain a “sweet spot” between the fixed and random regimes, where uncertainty and creation are in a productive balance. This could be used to create a self-regulation mechanism for the computer generated swarms in Swarm-PI.

### 6.2 Modeling Memory

Human performers not only respond to the immediate environment, but also to things that happened in the immediate past and to things that they anticipate happening in the future. The novelty of emerging patterns is always measured against expectations held in memory. The neighborhood of interactions extends in time as well as in space.

The human player’s interaction with Swarm-PI includes the player’s memory and anticipation. Ideally, the swarm players would respond not only to their immediate environment but also to a span of “remembered” events. One method to implement system memory would be to have swarms lay persistent or recurring stigmergic traces or trails in the virtual swarm space. These traces, functioning as attractors or obstacles, would imprint the topology of the solution space with a map of past events. Blackwell uses stigmergic attractors rather than a swarm avatar to represent human input [4]. We would like to retain the advantages of using an avatar and add stigmergic mechanisms for introducing responsiveness over a range of time frames.

### 6.3 Self-reflective Feedback

Many models of creativity recognize evaluation as essential to the process; human players recognize the processes they are engaged in as productive or not and adjust their responses appropriately. Similar feedback for swarm players might be implemented using measures of information flow and pattern emergence, which could be used as prompts modulating mappings and swarming rules. Feedback mechanisms could also contribute to the organization of the system over longer time scales.

### 6.4 Evolution of Swarms

Measures of information flow and pattern emergence could potentially be used as fitness measures for the evolution of the computer generated swarms, producing additional interesting dynamics. The values for the rule parameters governing a particular swarm would be that swarm's genome. Exchanging some rule parameters between two swarms would result in two child swarms, similar to [12]. Mutation would introduce slight changes in the parameters. The two parent swarms would die, via age-based mortality, and be replaced by the child swarms. The lifespan of a swarm (with a suitable lower bound to prevent an early demise) would be one of the parameters described by the genome. This evolutionary scenario would produce system dynamics for each generation that are new, yet related to the dynamics of the previous generation.

## 7 Conclusion

Swarm-PI provides a framework for explorations of swarm-based interactive performance. The swarm avatar provides a flexible mechanism for mediating interactions between the human player and the improvisation system. Avatars exist in a rich and complex space; the parameter space is very large and avatars are capable of a wide range of behaviors. This supports the mapping of a detailed characterization of the human player's output, including non-musical gestures and cues, into the virtual space of the system. The complexity of the possible mappings provides the opportunity for subtlety in the system's internal dynamics and its reaction to input, and it affords users great flexibility in configuring the environment. We have sampled only a tiny fraction of the ways human output can be mapped onto avatars and the ways avatars can interact with the computer generated swarms. Further explorations of this complex space may uncover new modes of interaction between human and computer performers.

**Acknowledgments.** The authors would like to thank Bowdoin College students John Burlinson, Nicole Erkis, Grace Handler, Octavian Neamtu, and John Truskowski for their work.

## References


1. Bascou, C., Pottier, L.: GMU, a flexible granular synthesis environment in Max/MSP. In: Proceedings of the Sound and Music Computing Conference (2005)
2. Beyls, P.: Interaction and self-organization in a society of musical agents. In: Proceedings of the ECAL 2007 Workshop on Music and Artificial Life (MusicAL 2007) (2007)
3. Blackwell, T.M.: Swarm granulation. In: Romero, J., Machado, P. (eds.) The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music, pp. 103–122. Springer, Heidelberg (2008)
4. Blackwell, T.M.: Swarm music: improvised music with multi-swarms. In: Proceeding of the 2003 AISB Symposium on Artificial Intelligence and Creativity in Arts and Science, pp. 41–49 (2003)
5. Blackwell, T.M.: Evolutionary computer music. In: Miranda, E.R., Al Biles, J. (eds.) Swarming and Music, pp. 194–217. Springer, London (2007)
6. Blackwell, T.M., Jefferies, J.: Swarm tech-tiles tim. In: Rothlauf, F., et al. (eds.) EvoWorkshops 2005. LNCS, vol. 3449, pp. 468–477. Springer, Heidelberg (2005). doi:10.1007/978-3-540-32003-6_47
7. Borgo, D.: Sync or Swarm: Improvising Music in a Complex Age. Continuum Publishing, New York (2005)
8. Jehan, T.: Max/MSP (2012). http://web.media.mit.edu/~tristan
9. Jones, D.: AtomSwarm: a framework for swarm improvisation. In: Giacobini, M., et al. (eds.) EvoWorkshops 2008. LNCS, vol. 4974, pp. 423–432. Springer, Heidelberg (2008). doi:10.1007/978-3-540-78761-7_45
10. Reynolds, C.W.: Flocks, herds, and schools: a distributed behavioral model. SIGGRAPH Comput. Graph. **21**(4), 25–34 (1987)
11. Rowe, R., Singer, E.: Two highly-related real-time music and graphics performance systems. In: Proceedings of the International Computer Music Conference, pp. 133–140 (1997)
12. Sayama, H.: Swarm chemistry. Artif. Life **15**(1), 105–114 (2009)
13. Sulis, W., Combs, A.: Nonlinear dynamics in human behavior. In: Studies of Nonlinear Phenomena in Life Science, vol. 5. World Scientific (1996)
14. Zausner, T.: Process and meaning: nonlinear dynamics and psychology in visual art. Nonlinear Dyn. Psychol. Life Sci. **11**(1), 149–165 (2007)