Ambient Sound Spaces

Olivier Pasquet
Independent artist & IRCAM, 1 place Igor Stravinsky, 75004 Paris, France
olivier.pasquet@ircam.fr

Philippe Codognet
JFLI – CNRS / UPMC / University of Tokyo Information Technology Center, 2-11-16 Yayoi, Bunkyo-ku, Tokyo 113-8658, Japan
codognet@jfli.itc.u-tokyo.ac.jp

Abstract We present in this paper a media installation creating in real-time ambient electronic music from a multi-agent simulation by associating agents with sounds. More specifically, we associate various sound parameters with the internal state of an agent at a given time (position, orientation, internal variables, etc), and therefore the evolution of an agent during the simulation will modify the corresponding soundscape. The sound sources are dynamically spatialized in the actual installation space through several loudspeakers (24 loudspeakers + 1 subwoofer) and modified in real-time, providing thus an ever-changing ambient soundscape. This generative music installation uses Nature-inspired simulations to drive the musical processes, and in particular the swarm intelligence metaphor.

Keywords Ambient multimedia installation; media art; multi-agent systems; simulation; computer-based music; generative music; real-time sound processing;

1 Introduction

The basic idea of the *Sound Agents* system which is detailed in this paper is to create an immersive sound-space by relating real space and virtual sound space. It can be seen in the tradition of immersive Virtual Reality systems where video projections and computer graphics will recreate in a real space a 3D virtual world, cf. for instance the well-known CAVE system develop by University of Illinois in the early 90's. In the last two decades, advances in computer graphic rendering techniques and efficiency of specialized hardware, improved visual immersion, but to further reinforce the immersive aspects of such virtual environments, the idea of populating virtual spaces with virtual creatures or agents has been growing in the recent years a major focus of research. The motivation of the *Sound Agents* system is rooted in the development of virtual autonomous entities for immersive environments. However in *Sound Agents*, each virtual entity will not be a visual character but an invisible sound agent producing sound, which will have its own autonomous behavior. This sound will actually be localized in the actual space of the installation, much as a CAVE-like VR system. Sound agents will be like bees or butterflies flying around spectators, but you cannot see them, just hear them. This ambient media installation aims at creating in real-time electronic music from a multi-agent simulation, by relating autonomous virtual agents with sound sources and associating various sound parameters with the actual state of an agent at a given time (position, orientation, internal state, etc). The sounds are dynamically spatialized in the actual installation space and modified by the movement of the virtual agents.Therefore the ambient music generated by the multi-agent simulation produces an ever-changing music soundscape, so-called "generative music" as the music is automatically generated by a computational process.

The “Sound Agent” system itself consists in three parts:

- a multi-agent simulation engine, which will determine and compute the behaviors of many concurrent autonomous agents,
- a sound generation engine, which will transform agent parameters into sound sources in real-time,
- a sound rendering device, spatializing the sound sources in the real space of the installation, through many speakers: 24 loudspeakers organized as a 4 x 6 matrix and one subwoofer.

In this generative music system, we consider multi-agent simulations of Nature-inspired processes, in particular the so-called swarm intelligence. Swarm intelligence as been defined as follows [4]: “Swarm Intelligence is the property of a system whereby the collective behaviors of (unsophisticated) agents interacting locally with their environment cause coherent functional global patterns to emerge.” Swarm intelligence has been exhibited by colonies of ants, and termites, bees, fishes, etc, who are able to look for food and define an optimal path, create complex 2D or 3D structure or exhibit schooling behaviors by applying simple local rules for individual behaviors and no global intelligence or direct communication (communication through the environment, called Stigmergy is however possible, e.g. pheromone deposit by ants). The basic example of ant foraging is as follows: while walking and searching for food, ants may (1) deposit a pheromone on the ground, (2) follow with high probability pheromone trails that they sense on the ground. Therefore individual ants deposit pheromones from food source to nest, and foragers follow trails. Thus the key interest of swarm intelligence lies in the fact that only simple and easy to understand local rules have to be programmed, while a complex behavior will be exhibited by the overall population of agents, the so called emergent behavior. Therefore this seems to us quite intuitive and easily understandable by music composers, who could use this metaphor in order to construct their musical works.

This paper is an extended and revised version of [6] and is organized as follows. Section 2 details the general architecture of the Sound Agents system and the basic hardware. Section 3 provides the motivation and the background related to multi-agent systems and swarm intelligence for music composition, while Section 4 details the swarm intelligence simulation and the current prototype, which is based on ant foraging., To further develop high-level and interactive ambient soundscapes, Section 4 briefly presents a declarative language for describing agent behaviors, based on the notion of goal constraints and Section 5 introduces the mechanisms for interactivity with respect to the user/spectator. A short conclusion ends the paper.

2 The Sound Agents System

Figure 1 depicts the overall architecture of the *Sound Agents* system.

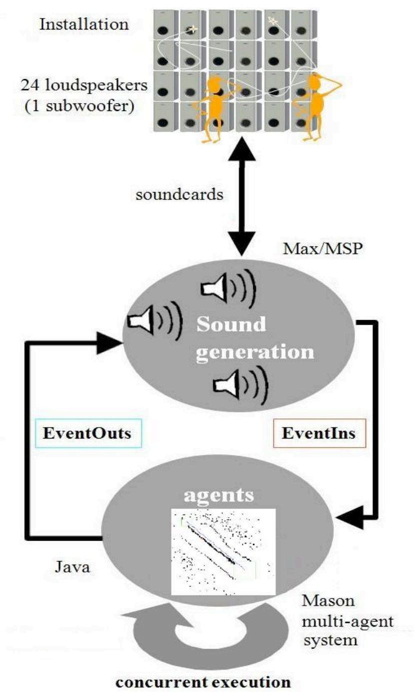

Fig. 1. System Architecture

The “Sound Agent” system consists in three parts:

1. A multi-agent simulation engine, which will determine and compute the behaviors of many concurrent autonomous agents. We use for this part the Mason multi-agent system [13].
2. A sound generation engine, which will transform agent parameters into sound sources in real-time. This is done within the Max/MSP visual programming environment and real-time sound processing environment.
3. A sound rendering device, spatializing the sound sources in the real space of the installation. We use 24 loudspeakers and one 1 subwoofer, organized as a 4 x 6 matrix, either vertically (“Sound Wall”), or horizontally on the floor. The second type of installation makes it possible for a more accurate rendering of spatialization.

The implementation of Sound Agents thus consists of integrating a Java-based multi-agent simulation engine, the Mason system [13], in the Max/MSP real-time sound generation software, both systems communicating in a bidirectional manner. Max/MSP is controlling the system parameters and is giving the timing, in order to have the system iterations synchronized with the rest of the audio engine. It is thus possible to use a time grid for giving the agents a pulse to develop their step-time behaviors and progress to the next time-step.
The hardware system configuration is composed of one Apple MacBook Pro, two M-Audio Profire 2626 soundcards (2 x 8 outputs), one Behringer ADA8000 ADAT soundcard (8 outputs), 24 loudspeakers (M-Audio Studiophile AV30), and one additional subwoofer (Tapco SW-10).

Concerning the actual rendering in real-time of the spatialized, moving sounds, the best would be to use Wave Field Synthesis (WFS) technology [19]. Such systems are commercially available but very costly. We have thus rather designed a low-cost approximation by using many small loudspeakers located in the installation space (using three 8-channel sound cards and thus 24 speakers), organized as a 6 x 4 matrix, plus one global subwoofer. Alternative configurations (e.g. a curved surface or a circle) are also possible and will be experimented in the future. Figure 2 depicts two different configurations of the audio rendering device.

(a) “Sound Wall”

(b) "On the Floor"

Fig. 2. Two different installations of the 24 loudspeakers

3 Agents, Randomness and Music Creation

With their ability to create complex structure through simple local interaction rules, multi-agent systems have attracted in the recent years, the attention of computer-music researchers. [9] briefly presents without much details a composition system based on agents who can alter a "musical space" by adding, removing or moving sound sources (not performing sound synthesis themselves), but the authors say themselves that the agent behaviors and the overall system seems quite difficult to program. A complex model of musical cognitive agents is presented in [25], and applied for example to low-level (but difficult) tasks such as beat-tracking and harmonic detection. On the other hand, agents can make use of expert knowledge and high-level representations to better perform on specialized tasks, and this is developed in [14], which proposes a multi-agent system for musical accompaniment of Latin music. A model of rhythmic agents is proposed in [16], where a collection of agent representing percussive sounds is used to evolve rhythms and recreate traditional rhythms through agent interactions. Closer to our approach, [10] presents an interactive system based on the simulation of particle swarms (e.g. like fish schooling or bird flocking). This system is interactive: the general movement of the swarm can be controlled in real-time by a musician. It is thus more an electronic musical instrument for live performances rather than a generative music system. Finally, the framework of [3] proposed a general context for self-organized music with a particular application of swarm music to improvisation performances.

However multi-agent systems are just the most recent (although arguably the most advanced) of many formalisms that has been investigated by music composers in order to give a general overall structure to their compositions. Without entering into details that are out of the scope of this paper, we can nevertheless recall that music composers have for a long time been interested in the use of complex structure that would govern the principle of their musical compositions, and indeed some of these works can be seen retrospectively as some sorts of agent systems. For instance the Greek/French composer and pioneer in electronic music Iannis Xenakis (1922-2001), has proposed in the middle of the 20th century to use various mathematical models such as set theory, game theory, statistical mechanics or stochastic processes in order to define music compositions. He developed his theory of "Formalized Music" in [26], in which he proposed in particular to consider abstractions of musical forms created by stochastic processes, the so-called "Free Stochastic Music". We do not want here to go in detail in his theory but what is worth noticing is that he uses randomness in order to define each individual notes but nevertheless keeps the overall structure of the music composition under strict rules, giving it some predefined form. The fact that he has been trained as an architect and has worked as an assistant of Le Corbusier certainly influenced his conception of music as the relation between an abstract musical form (which can be sometimes visually depicted, e.g. an hyperbolic curve) and the individual notes in the score of the partition.. He was utmost interested by the fact that “natural events such as the collision of Hail or rain with hard surfaces or the song of cicadas in summer fields (…) are made out of thousands of isolated sounds; this multitude of sounds, seen as a totality, is a new sonic event. This mass event is articulated and forms a plastic mold of time, which itself follows aleatory and stochastic laws” [26]. Of course, Xenakis was thinking within the formal models of his time (statistical mechanics or stochastic processes) but his theory of multiple isolated sounds articulated through a formal model and forming a “totality” and a new musical structure could be rephrased today with multi-agent systems.

One interesting question to further investigate is how a system such as *Sound Agents* can be used by music composers and in which respects it can be better than introducing simple randomness in music composition.

Indeed the use of randomness in music composition can be traced back to various musical games based on combinatory structures in the Baroque Era, such as for instance Mozart's *Musikalisches Würfelspiel* (musical game). In 1787, Mozart wrote a music composition game in which one can create a Minuet by assembling pre-composed measures chosen randomly among 176 possible Minuet measures and 96 possible Trio measures. A dice roll is used to select from a look-up table. In modern music composition, the use of randomness is usually attributed to John Cage (1912-1992) who was one of the key music composers in America in the 20th century. Cage is well-known for his innovative introduction of randomness in music composition, which is even often considered too drastic by critics and other composers (including Xenakis) as it limits the role of the composer. Historically, Cage started to use chance for music compositions in his work *Music of Changes* (for solo piano, 1951), which was based on the Chinese “I-Ching” (the book of changes) for randomly selecting notes in the score. Later, *Imaginary Landscape n°4* (1951) was conceived for 12 radios, with operators controlling in real-time the volume and tuning of the radio but nothing more. These experiments with randomness culminated with the multimedia performance *HPSCHD* (harpsichord), performed during four and a half hours on May 16, 1969. In this work, a computer made millions of « chance operations » in real-time in order to keep busy 7 keyboard players, 52 tape recorders, 52 film projectors and 64 slide projectors, selecting randomly what was actually played from a database of sounds and images. One can also include in this “chance music” Cage’s controversial masterpiece: *4 '33"*. It is nothing but 4 minutes and 33 seconds of silence, and was premiered and “played” by experimental pianist David Tudor on August 29, 1952. His idea was to let the spectators listen during the duration of the music piece to all the (random?) noises that can appear in a concert hall: one person coughing, a door opening, etc.
It is interesting to note that we have with Cage two ways of using randomness for music composition. One that can be called *internal chance*: randomness is used at composition time to determine the score, which is then written and closed. And another one that can be called *external chance*: the score is open to real-time chance operation and external perturbation. In that respect, *4 '33"* is certainly the best and minimal example of external chance.

Multi-agent simulations can be seen as intermediate between these two kinds: randomness is occurring in real-time during the simulation (e.g. the random walks of the ants) but within the framework of the simulation rules (e.g. limited set of behaviors for each agent). The key point in using multi-agent simulations and in particular swarm intelligence rather than simple randomness in music composition is to have an emergent property appearing after some time in the simulation. Therefore, knowing this emergent property in advance makes it possible to use it as a general movement for the music composition, even if the precise details or the exact timing of the emergent form are not known and are different for each simulation.

4 Agent Simulation and Swarm Intelligence

We are interested in multi-agent simulations with a large number of rather simple agents (about one thousand agents) interacting between themselves and with the environment. Some classes of such multi-agent systems are exhibiting an emergent behaviour; in particular what has been called *swarm intelligence*. We consider sound entities to be reactive agents receiving percepts from the environment and acting on the environment by moving and producing sounds. Reactive agents have no symbolic model of the world they live in, but rather use sensory-action control loops in order to perform tasks in a robust manner, cf. the model depicted in Figure 3 below.

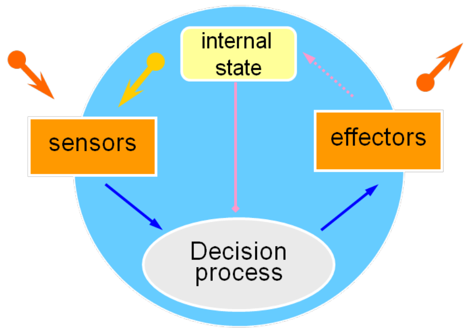

Fig. 3. Simple agent model

Indeed, we will consider multi-agent simulations with several hundreds or even thousands of agents, thus an efficient and simple agent model is prefered in order to produce sound in real-time from the simulation. In a biologically-inspired metaphor, one could consider sound agents as bees or butterflies flying around the spectators/listeners in a 3D world or ants, termites or other small animals in a 2D world, using the *swarm intelligence* metaphor. This makes it possible for the music composer to intuitively envision the emergent behavior of the agents, even if the movements of each agent cannot be fully controled in detail (autonomous behavior). In the current implementation, we are considering well-known swarm intelligence simulations such as for instance ant foraging, but we use it in a musical context. Indeed, we associate a sound generator with each agent and the movements of the agent and its position can modify both the spatialization of the sound in the actual installation space and also some parameters of the real-time sound generation process (e.g. pitch, oscillator type, reverb FX, etc). The ant foraging simulation will generate a multitude of sounds that first seem to disperse randomly and then organize on a path linking the ant nest to the food source, as ants are attracted on shortest path through deposit of pheromones. Therefore the emergent behavior of the swarm ("convergence", "optimal path", etc) is used to create the macroscopic structure of the musical composition, while the agents themselves and their movements are used for the microscopic structure. Then a crucial point lies in defining the relationship between the observation of the simulation and what the composer wants to be observed from the musical point of view (for instance the convergence towards an harmonic or a rhythmic structure).

We use for the first work a simple swarm intelligence simulation: ant foraging. This is a classical example that has been studied by entomologists and then simulated in the computer since the early 90's [2]. The basic idea is that the ants deposit pheromone on the paths that they cover and as ants going on the shortest path are getting back to the nest quicker and will go again for more food searching, therefore the shortest path will contain more pheromone than others. This results in the building of an optimal path. As ants are attracted by the pheromone, they will then more and more be attracted by the optimal path and therefore an optimal emerging behaviour will be constructed, without the ants having any model of the optimality per se. Such a simulation has been implemented in the Mason system [18] with a particular system of pheromone deposit, although other models have been proposed, such as [22], based on observations of real ants. We could use alternative implementations, but we are indeed not so much focused to the "reality" of the simulation: the key point for our installation is that the simulation exhibits the emergent property that ants converge towards the optimal path at some point.

Figure 4 depicts an example of the ant foraging simulation. The top left square represent the ant nest and the botton right green square the food location. The blue line between the two represent the optimal path. Ants are represented by black dots and some red dots, which are the ants that are linked to sound sources ("audible agents").

In (a), ants are getting out of their nest and start exploring randomly. In (b) further ants and exploration is performed, and a certain number of red "audible agents" can be seen. In (c) on path, which is not the optimal one but nearly parallel to it is created and will attract more ants. In (d) this path is strengthen (thicker, with more ants on it) but another sub-optimal path is created on the right of it.

The whole idea of using multi-agent simulation and swarm intelligence for generating music is to be able to associate to each agent some sound parameters that will be sumitted to variations depending on the position of the agent. Then the emergent property will ensure that agents will eventually converge towards some optimal path and therefore that some musical movement can be achieved, even if the actual timing or exact value the position of each agent of the swarm cannot be precisely defined in advance. This is why it was important to work on this project not only technically but also musically and we needed the help of not only a sound engineer but a sound artist. It was crucial for us to have a stand-alone sound-based installation, not only a working system which is a mere « demonstration » of music generated by swarm intelligence. Of course, there are some limitations in using an ant foraging simulation as a basis for musical pieces. This characteristic makes it possible to develop a variety of music forms and spatial movements evolving from something wide to something narrow. The evolution of the individual paths taken by the agents goes from random when the ants are looking for food to non-random when the ants have found the best path and when they go back and forth between the ant nest and the food. Therefore this is in the mapping between the swarm agents and the sound parameters that the musical aspects have to be defined and should fit with the overall evolution that will be guided by the emergent property of ant foraging simulation, that is, the fact that all ants will eventually converge to the optimal path.

Although best appreciated in an exhibition space with 24+1 loudspeakers, the resulting ambient music can be appreciated by looking at the following website, which include mp3 recordings and videos: http://webia.lip6.fr/~codognet/SA/

Figure 5 depicts a snapshop of the visualization window of *Sound* Agents during a running simulation. The ant simulation is depicted in the center, with the top left square being the ant nest and the bottom right larger green square being the food location. Thus the diagonal line is blue represents the optimal path. Black dots are ants, and red dots are ants which are acting as sound sources.

Therefore it can be seen that many ants follow a nearly optimal path, while two other nearly parallel paths have also been created. On the right side of the screenshot, VU-meters corresponding to the output levels of the 24 loudspeakers are depicted.

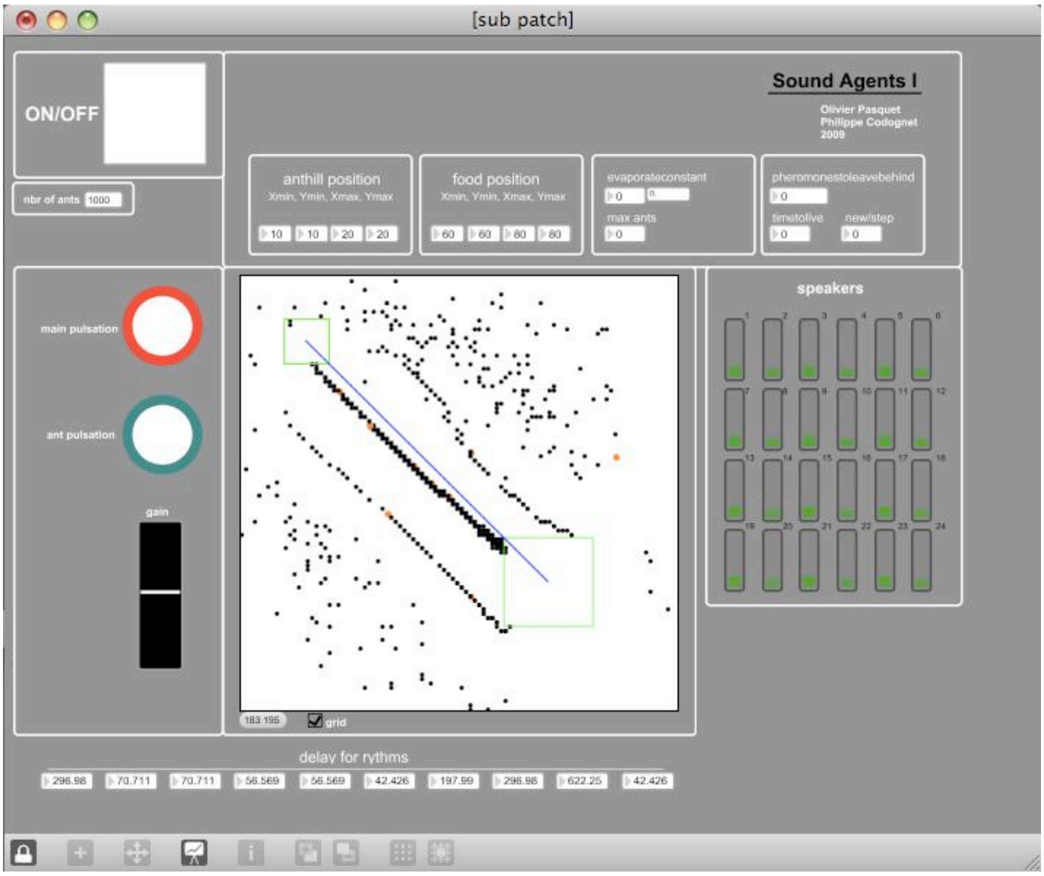

Fig. 5. Visualization window with ant simulation and loudspeakers VU-meters

We will now detail the musical choices and the structure of the musical piece developed in *Sound Agents* in the rest this section.

In the first version of the *Sound Agents* installation, the ant foraging simulation is used to produce generative ambient music in real time. About one thousand agents are used in the simulation but it would be too costly in computation time to associate one sound source to each agent (as the sound has to be generated in real-time), we randomly choose about 25 agents among 1000 that will be associated to sound sources. These will be synthetized sounds created by simple oscillators whose parameters will be given by the position and orientation of the agents. We used simple oscillators for performance reasons, but more complex treatments are of course possible with Max/MSP and will be experimented if we can have extra computing power. It is interesting to note that in order to have better performances, we also had to go down to the parallel processing aspects of the dual core processor of the Powerbook Pro and explicitely program Mason and Max/MSP to run on different cores and communicate via the main memory. This has greatly improved the performances of the system. Observe however that, in a simulation with 1000 agents, the 975 « non-audible » agents are nevertheless useful because they interact and will influence the behavior or the 25 audible agents through stymergy (pheromone deposit). We can also associate to the pheromone deposit itself a musical meaning and a specific sound generation. Concerning the sound spatialization, any mapping could be done between the 2D simulation space and 2D matrix of loudspeakers, but we prefered in this first version to do a naive one and map the simulation directly onto the loudspeaker grid. The overall movement of the simulation, which is that all ants first walk randomly in the 2D space and eventually converge to a path close to the optimal (i.e. the line linking the nest to the food, as we have no obstacle in this simulation), will be musically reflected by the fact that all audible agents will eventually converge to a common rhythm, with minor variations. As stated above, we also decided to make the pheromone deposit audible and we add an extra sound source which is some kind of continuous bass chords, whose volume depends on the level of pheromone, and this sound is further modified by a distortion effect whose parameters depends on the location of the pheromone deposit.

An important issue in music is time, and in *Sound Agent* it lies in the way both the agent simulation and the sound generation are scheduled and synchronized. In a general setting, the scheduler running the agent simulation is not obviously linked with the timing of the music piece. Everything depends on the amount of epochs needed for the system to converge and on the length of convergence in the music, as the way time is calculated and its segmentation may differ. Time for music could be directly "wired" to the time of the simulation or it could be driven by any event happening in the simulation. For instance,

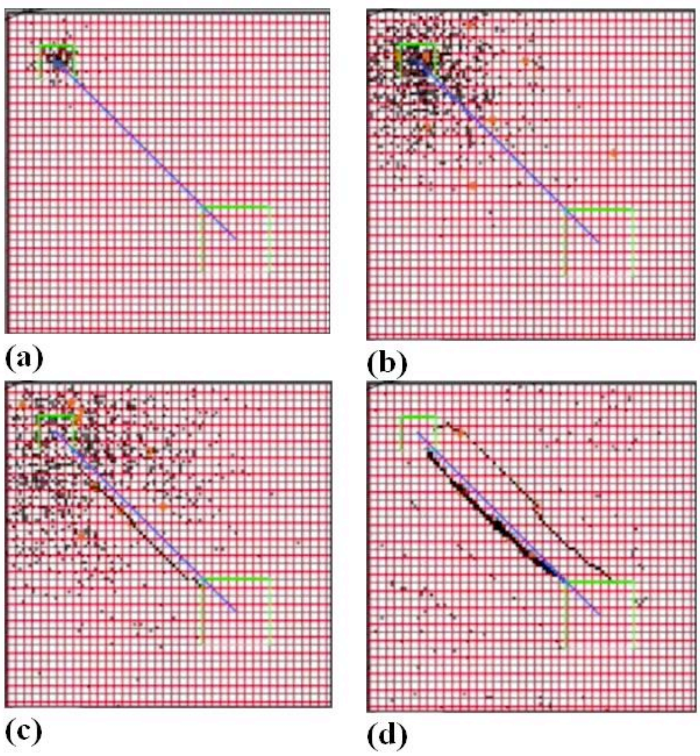

Fig. 4. four steps in the ant foraging simulation

every time an ant enters a new segment of space, the musical time could make one more step. However for simplicity reasons the first and direct approach has been chosen in the current version of *Sound Agents*; thus the time needed for the simulation to converge is also the time of the musical piece. A master clock is used, which is external to the multi agent simulation and drive both Mason and Max/MSP. This way the amounts of time are the same for both but their segmentation can vary thus both the music and the agents can run independently as many events as needed. At regular ticks, the clock manager reads the position for each active ant (depicted in orange in Fig. 4) and the amount of pheromones in the plane, in order to get a pulsation for music. Using a pulsation is not an obligation but rhythm is important from a perceptive point of view. This information will be used to generate the sound in Max/MSP. The path of each active ant is individually tracked and each has a specific sound or pitch. However, in order to keep a global homogeneity, the sound differences between ants are quite subtle and amounts mainly in differences in rhythmic patterns. Indeed, the overall music composition is divided in four layers, or tracks, detailed as follows.

One layer is concentrating on the rhythm. Several classes are stored in a tank. Using rhythmic classes is often convenient because it gives an intelligible redundancy and a better beat synchronization between one another. Each active ant is triggering its sound with the rhythm corresponding to its position. Also, in the system setup, each active ant is positioned at the same coordinates as the one in the multi-agent system. Therefore, high-pitched percussive sounds has been chosen in order to get a precise feeling of positions in the space. Theses tiny pitches are also in aesthetic opposition with the other sounds. This layer could thus be see as the lead. The closer the ants are from the optimal path, the simpler are the rhythms. This way, the rhythmic evolution goes from something quite chaotic to something almost minimalist and repetitive. The food path is roughly known before the start because the position of the anthill and the food are decided by the user before-ward.

The second layer is dealing with the quantity of pheromones and its distribution in the space. It is a continuous accumulative ingredient over time. One radical and efficient way to describe it is distortion. The more pheromone there are the more distortion there is on a continuous base sound. A pretty harmonic and smooth continuous sound has been chosen, in opposition with the bright and tight rhythm sounds. Harmonics are continuously evolving and getting more and more distorted with a tube overdrive digital model. Pheromones are not distributed equally in the space since they are on the paths taken by the ants, thus this second layer is also spatialized on the 24 loudspeakers. The amount of pheromone left in the (simulation) space is directly linked with the amount of distorted sound diffused in the (music) space from the speaker array.

The two other layers are dictated by purely musical concerns and have no concrete links with the agents. But, as their presence is musically important, they perceptively and meaningly enhance the layers produced by the agent system.

The third layer is a bass drum. The complexity of the first rhythmical layer needs a bass drum in order to segment time and better perceive rhythmical phrases. This bass drum is giving a pulsation in the same way indian music would do with a triangle and a mridangam. It's triggering is probabilistic so it does not always appear and it is not too obvious. Thes "right" amount, empirically chosen by taking into account the very complex beginning of the process and the more minimalist ending of the high pitched rhythmic track, is aimed to be constant. Also the opposed second track, going from harmonic to distorted has to be taken into account. This bass sound does not need spatialization in the space.

The last layer is made for linking everything together. The listener does not have to concentrate on this one but its presence it rather important. The synthesis for this track is made of 8 chaotic oscillators. This allows an expressive and continuous evolution of thin and precise sounds. The muddle evolution, continuous and pretty fast, could be felt as a voice, since its tessiture is right in the middle of the basses and high pitched rhythms.

5 A Declarative Language for Agent Behaviors

In the first version of *Sound Agents*, we use the Mason multi-agent system [13] in order to drive the simulation. Although a powerful and efficient system, Mason nevertheless requires serious programming skills for describing agent behaviors and it is thus difficult for a composer to directly modify the simulation code, or to experiment a new type of multi-agent simulation by himself. Therefore we are currently developping for the second version of the system a high-level language to describe agent behaviors that could be easily understood by someone with limited programming skills. Simple but interesting life-like behaviors with emergent properties should be easily implemented, such as those described in [5]. In computer graphics and animation systems, the most common formalism for representing behaviors of high-level agents, such as virtual humans is some extension of finite state automaton (FSA) [15,27] or more complex hierarchical models [23,12]. For low-level agents, such as the swarm agents in flocks or herds and reactive agents, two basic approaches are classically used:

1. Steering behaviors, where the different low-level goals (such as grouping or escaping) are stated as forces that are then added to produce the actual behavior of the agent in a time-step manner. This approach has been pioneered by Reynolds since the late 80's [20,21], but it still active now and various extensions have been proposed [17,23,8].
2. Particle systems [24] or potential fields [11] treating the swarm as a complex physical system.

We are obviously closer to the first approach, but we propose to use the formalism of CSP (Constraint Satisfaction Problems) as a general behavior description language. Constraints are used to state goals, or more exactly partial goals, that the agent has to achieve. This can be seen as an extension of the steering behavior approach where constraints are solved logically instead of forces added numerically. One interesting point however is that the constraint formalism is naturally nondeterministic, as opposed to any force-based formalism such as steering behaviors, which is intrinsically deterministic. Indeed we find here again the classical dichotomy between declarative and procedural languages. We believe that a declarative, nondeterministic formalism such as that of goal constraint is more powerful and easier to use than a procedural one.

As we are mainly concerned in the *Sound Agent* project with the motion of agents in a 2D or 3D space, we should focus on the specific goal constraints related to navigation. Indeed we want to generate the trajectory of an agent as the (iterative improvement) solving of the goal constraints at each time-step, which will generate the actual movement of the agent. Therefore we can simply consider a single variable for each agent, which is its position in the virtual space. As we are in a combinatorial search setting, we will consider that the domains of the variables are representing some discrete approximation of the real space. The key idea is that the agent will look at the possible positions in his neighborhood, check the combination of errors of his own goal constraints and choose the position minimizing the error. Moreover the use of a special mechanism makes it possible to forbid some areas for a given period of time (inspired from Tabu search) and thus helps to prevent being trapped in a local minimum (e.g. a spatial dead-end) or short-term oscillating behaviors. More details about the implementation and the local search algorithm for solving goals are described in [7]. Indeed comparing to the steering behavior paradigm, we have here non-deterministic behaviors as we do not try to define a combination of forces that will bring the agent to the desired position but just check possible locations and choose the best one, which can be done efficiently in the local search approach. Also observe that we can cope with dynamically changing priorities between behaviors, as the error functions are reevaluated at each time steps.
Below are some examples of declarative navigation goals for reactive agents.

Constraint	**Declarative meaning**
In(Region)	Stay within the zone define by *Region*
out(Region)	Stay outside the zone define by *Region*
go(Object)	move towards the location of *Object*
away(Object)	move away from the location of *Object*
Attraction(Stimulus)	Move towards source of stimulus
Repulsion(Stimulus)	Move away from tsource of stimulus

Table 1. Examples of Goal Constraints

6 Ambient Spaces and Interactivity

We are considering constructing in the new version of the *Sound Agents* system a sound-based installation space that can combine the multi-agent simulation of swarms (generating sounds and ambient music) with the tracking of the location of spectators in the real space to make the sound evolve while visitors are entering and moving within the installation space. For instance if we map the real space of the installation into the 2D space of the "ant world" and consider spectators as obstacles, then ants will have to adapt and find different optimal paths for food. Of course, as spectators are moving in the installation space, obstacles in the "ant world" are moving, and routes followed by ants are changing, generating therefore different sounds. We thus have an interaction between the spectators/users inside the real space of the installation and the virtual agents (ants) in the simulated world, who interact in a simple but interesting way.

It is worth noticing that, in the first implementation with ant foraging, the communication is only unidirectional: the Mason system drives the simulation and communicates events to the Max/MSP system which generate sounds that are transmitted to the loudspeakers through the soundcards. There is indeed some communication from Max/MSP to Mason, as Max/MSP is driving the simulation ticks, but none from the installation space to the simulation space. However we intent in the future to include sensors in the installation space and communicate back information from the installation space to Mason through Max/MSP. For instance, we intend to place some position sensors or an omni-directional camera to track the positions of spectators and include this information as obstacles in the ant system simulation, influencing therefore the core simulation process as a feedback and the whole music generated. This is the final goal of the *Sound Agents* installation, even if this feedback loop through position sensor is not implemented yet. One important remark is that the perception of the sounds in space by humans is not very precise. Most of the people have a precision of about 10%, meaning that they cannot differentiate sound source below 10% of the distance between the auditor and the sound source, e.g. below 10 cm for a sound source at one meter distance [1]. This is of course even worth for sound reproduced between several loudspeakers. This is not really a problem for our installation because we aim at producing an ambient spatialized music, not a precise localization of sounds in space (we would need a WFS system for that). But this also means that we do not need to have very precise position sensors (e.g. no need to have an accuracy below 10 cm) in order to have an interesting feedback by tracking the position of the spectators and introducing it in the simulation, as obstacles in the 2D Ant World.

7 Conclusion

We presented a media installation relating real space and virtual sound space. It follows previous work on virtual agents and autonomous characters in 3D immersive spaces. However in *Sound Agents*, each virtual entity will not be a visual character but an invisible sound agent producing music, which will have its own autonomous behavior. This sound is to be actually spatialized in the real 3D space of the installation, through several loudspeakers (24 loudspeakers + 1 subwoofer). Sound agents are like butterflies flying around spectators, but people cannot see them, just hear them. Therefore the ambient music is dynamically spatialized in the actual installation space and modified by the movement of the virtual agents, providing thus an ever-changing music soundscape. Each sound agent is autonomous and his behavior is described by simple behavior rules. For the first prototype implementation, we used an ant foraging simulation which was generating music in real-time. We are currently developping a more simple and flexible agent-based system in order to make musical composition and experiments easier. Also, this can be combined with the tracking of the location of spectators in the real space to make the sound evolve while visitors are entering and moving within the installation space, creating then an interactive installation.

Acknowledgement

We would like to thank the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT) for partly supporting this research in 2008 at Keio University Research Institute for Digital Media and Content.

References

1. D. H. Ashmead, D. Leroy, and R. D. Odom, "Perception of the relative distances of nearby sound sources", *Perception & Psychophysic*s, volume 47 issue 4 (1990), pp 326-331.
2. R. Beckers, J.L. Deneubourg, and S. Goss, "Trail laying behaviour during food recruitment in the ant Lasius niger (L)", *Insectes Sociaux*, vol. 39, pp 59-72, 1992.
3. T. Blackwell and M. Young, "Self-organized music", *Organised Sound*, vol. 9, no. 2, Cambridge University Press 2004, pp 123-136.
4. E. Bonabeau, M. Dorigo and G. Theraulaz*, Swarm Intelligence: From Natural to Artificial Systems,* Oxford University Press 1999.
5. V. Braitenberg, Vehicles - Experiments in Synthetic Psychology, MIT Press, 1984.
6. P. Codognet and O. Pasquet, "Swarm Intelligence for Generative Music", *proceedings of ISM2009,* IEEE International Symposium on Multimedia, San Diego, IEEE Press 2009.
7. P. Codognet and O. Pasquet, "Sound Agents", proceedings of EvoApplications 2010, C. Di Chio et al. (Eds.), LNCS 6025, Springer Verlag 2010.
8. I. Couzin, J. Krause, R. James, G. D. Ruxton, and N.R. Franks, Collective Memory and Spatial Sorting in Animal Groups, *Journal of Theoretical Biology*, vol. 218, no. 1, Pages 1-11.
9. P. Dahlstedt and P. McBurney, "Musical Agents: Towards Computer-aided Music Composition using Autonomous Software Agents", *Leonardo*, vol. 39 no. 5, 2006, pp 469-470.
10. T. Davis and O. Karamanlis, "Gestural Control of Sonic Swarms: Composing with Grouped Sound Objects", In: *proc. SMC'07, 4th Sound and Music Conference*, Lefkada, Greece, 2007.
11. X. Jin, C. C. L. Wang, S. Huang and J. Xu, Interactive control of real-time crowd navigation in virtual environment, *Proc. 2007 ACM symposium on Virtual Reality Software and Technology*, pp 109-112, ACM Press 2007.
12. F. Lamarche and S. Donikian, Crowd of Virtual Humans: a New Approach for Real Time Navigation in Complex and Structured Environments, *Computer Graphics Forum* vol. 23 no. 3, 2004, pp. 509-518.
13. Sean Luke, Claudio Cioffi-Revilla, Liviu Panait, and Keith Sullivan, "MASON: A New Multi-Agent Simulation Toolkit", *proceedings of SwarmFest04, 8th Swarm Users/Researchers Conference*, Ann Arbor, USA, 2004.
14. D. Murray-Rust, A. Smaill and M. Contreras Maya, "VirtuaLatin - Towards a Musical Multi-Agent System", In: *proceedings of ICCIMA'05, 6th International Conference on Computational Intelligence and Multimedia Applications*, IEEE Press 2005.
15. T. Noma, L. Zhao and N. I. Badler, "Design of a Virtual Human Presenter", *IEEE Computer Graphics and Applications*, vol. 20, no. 4, 2000, pp. 79-85.
16. F. Pachet, "Rhythm as emerging structure", In: *proceedings of ICMC2000*, *International Conference on Computer Music*, Berlin, Germany, 2000.
17. C. Pedica and H. Vilhjalmsson , Social Perception and Steering for Online Avatars, *In: Proc. IVA'08, 8th international conference on Intelligent Virtual Agents*, Tokyo, Japan, LNAI 5208, Springer Verlag 2008, pp 104 – 116.

18. L. Panait and S. Luke, “Ant Foraging Revisited”, In: proceedings of ALIFE9, 9th International Conference on the Simulation and Synthesis of Living Systems, Boston, USA, 2004.

19. G. Theile, “Wave Field Synthesis – A Promising Spatial Audio Rendering Concept”, *Proceedings of DAFX’04, 7th Int. Conference on Digital Audio Effects,* Naples, Italy 2004.

20. C. W. Reynolds, Flocks, Herds, and Schools, A Distributed Behavioral Model, *Computer Graphics,* vol. 21 no. 4 (SIGGRAPH '87 Conference Proceedings), 1987, pp. 25-34.

21. C. W. Reynolds, Steering behaviors for autonomous characters, *In: Proc. of 1999 Game Developers Conference*, San Francisco, Miller Freeman Group 1999, pp. 763-782.

22. E. J. H. Robinson, D. E. Jackson, M. Holcombe and F. L. W. Ratnieks, “Insect communication: 'No entry' signal in ant foraging”, *Nature* 438 (24 November 2005), pp 442.

23. W. Shao and D. Terzopoulos, Autonomous pedestrians. *In: Proc. of the ACM SIGGRAPH’2005, Symposium on Computer Animation*, ACM Publishing 2005.

24. A. Treuille, S. Cooper and Z. Popovic, Continuum crowds, *ACM Transactions on Computer Graphics*, vol. 25 no. 3 (SIGGRAPH 2006 conference proceedings*),* 2006, pp. 1160-1168.

25. R. Wulfhorst, L. Nakayama and R. Vicari, “A multiagent approach for musical interactive systems”, In: *proceedings of AAMAS'03, 2nd international joint conference on Autonomous agents and multiagent systems*, Melbourne, Australia, ACM Press 2003, pp 584-591.

26. I. Xenakis, *Formalized Music*: *Thought and Mathematics in Composition*, Indiana University Press 1971 (original edition in French, *Musiques formelles*, 1963).

27. H. Xiao and C. He, Behavioral Animation Model for Real-Time Crowd Simulation," *In: Proc. of 2009 International Conference on Advanced Computer Control*, IEEE Press 2009.