xCoAx 2015
Computation
Communication
Aesthetics
and X

Glasgow
Scotland
2015.xCoAx.org

The Emergence of Complex Behavior as an Organizational Paradigm for Concatenative Sound Synthesis

Peter Beyls
CITAR - UCP Porto, Portugal
pbeyls@porto.ucp.pt

Gilberto Bernardes
INESC TEC, Porto, Portugal
gbernardes@inescporto.pt

Marcelo Caetano
INESC TEC, Porto, Portugal
mcaetano@inescporto.pt

Keywords: Audiovisual system, distributed agent system, emergence, concatenative sound synthesis, interactive musical improvisation.

Multi-agent systems commonly exhibit complex behavior resulting from multiple interactions among agents that follow simple rules. In turn, complexity has been used as a generative and organizational paradigm in audiovisual works, exploiting features such as behavioral and morphological complexity with artistic purposes. In this work, we propose to use the Actor model of social interactions to control a concatenative synthesis engine called earGram in real time. The Actor model was originally developed to explore the emergence of visual patterns. On the other hand, earGram was originally developed to facilitate the creative exploration of concatenative sound synthesis. The proposed integration results in the emergence of complex behavior from the Actor model acting as an organizational paradigm for concatenative sound synthesis.

1 Introduction

Natural systems such as insect swarms, the immune system, neural networks, and even chemical reactions (Bak 1995, Kauffman 1995, Camazine 2003) are widely considered to exhibit complex behavior arising from multiple local interactions among agents following simple rules. The self-organizing behavior of social animals (Reynolds 1987) has been used to explain certain social interactions, including in human society (Ulanowicz 1979). Interestingly, the emergence of complex behavior in computer simulations of natural systems has been explored aesthetically as organizational paradigm in artistic settings such as dance (Tidemann 2007), audiovisual installations (Beyls 2012), sound and music (Miranda 1994, Blackwell 2002, Caetano 2007), and sculpture (Todd 1992), among others.

In contrast to top-down design in most cultural artifacts, natural systems exhibit patterns arising from multiple local interactions among individuals or entities that do not exhibit the patterns themselves. From the stripes of zebras to snowflakes and termite mounds, pattern at the global level emerges solely from interactions among lower-level components (Camazine 2003). Much research in the discipline of artificial life studies life-like emergence in forms of synthetic biology (Langton 1997). Recent work in artificial chemistry (Dittrich et al. 2001) offers a wealth of models for constructing emergent behavior. For example, the idea of molecular interaction may successfully underpin complex musical human-machine interaction (Beyls 2005). Various music systems were built exploiting swarming behavior (Blackwell and Bentley 2002) – a model first formalized in the original flocking algorithm (Reynolds 1987). Miranda (1994), in turn, proposes to use the patterns that emerge from cellular automata in music composition. Caetano (2007) exploits the self-organizing dynamics of different algorithms inspired by biological systems to obtain trajectories that drive sound transformations.

In this work, we propose to use the complex behavior that emerges from a multi-agent system called the Actor model to drive earGram, a concatenative synthesis engine, in real time. The Actor model of social interactions uses the concepts of affinity and to iteratively displace the agents, called actors, to different settings of social stress. The self-organizing nature of the Actor model results in intricate visual trajectories followed by the actors. These trajectories, in turn, are used as input to earGram, a concatenative sound synthesis engine. EarGram organizes a collection of sounds in the plane according to their intrinsic perceptual qualities, such that neighboring sounds are more similar than sounds that are far apart. Therefore, spatial trajectories result in sonic trajectories that become gradual transformations along the perceptual dimensions used to organize the sounds. The user can choose the sound features corresponding to the dimensions of the space, which results in different configurations of the sounds in the plane. Consequently, the same trajectory can have several different sonic results.

Our goal is to build a system supporting non-trivial rewarding human-machine interaction. In contrast to conventional linear mapping, the user interacts with the Actor model indirectly by changing the affinity and sensitivity values, which results in different dynamic configurations. The system dynamics becomes the organizational paradigm followed when exploring the conceptual space of sonic results. The actors behave autonomously from the specification of simple local instructions, yet the system is open to disturbance by an external human performer (HP), offering fascinating aesthetic potential for human-machine interaction. Then, a perception of life-like qualities becomes apparent, one interacts with a quasi-unpredictable system while the structural integrity of that system remains. Such a work suggests critical consideration of the notions of interactivity, intricacy, participation and unpredictability.

This paper is further structured as follows, firstly we explain the Actor model and its behavioral scope, then we address concatenative sound synthesis in earGram. Finally, the implementation of a functional bridge between both components is presented. We discuss the visual and sonic components of the system, including aesthetic considerations and user interaction.

2 An Emergent Organizational Paradigm

Linear top-down planning and design suffers from a knowledge acquisition bottleneck. In contrast, collective behavior commonly presents self-organizing properties whereby pattern at the global level emerges solely from interactions among lower-level components. Remarkably, even very complex structures result from the iteration of surprisingly simple behaviors performed by individuals relying only on local information.

2.1 The Party Planner Model

Our implementation is inspired by the Party Planner Model (PPM), developed by Rich Gold and documented in his seminal book *The Plenitude* (Gold 2007). Imagine a party where each individual aims to be physically close to people one likes and as far away as possible from people one dislikes. An individual's level of unhappiness is the perceived social stress impinging at a particular location

$$S_A = \sum_{n=0}^{N-2} |D_i - D_a|$$

eq. 1

in physical space. Formally, given N actors, the level of unhappiness of actor A at index i is expressed in eq. (1) as the sum S_A of absolute values of the differences in ideal distance D_i minus actual distance D_a for all (N-2) actors. An actor does not express any social opinion towards itself, thus N-2 evaluations take place starting from index 0.

Every person aims to minimize his/her level of unhappiness by moving in space, to a neighboring spatial location, a few steps away from the current location, potentially offering less social stress. As a result, a person will relocate to his/her ideal distance from every other person thus minimizing the total perceived level of unhappiness.

In every process cycle, all actors consider eight alternative directions to move, as depicted in Figure 1. A list of different social tensions is computed from the observation of the grand sums of impinging stress. Finally, the algorithm favors the direction to move implying the least stress of all eight directions. All actors proceed according to the same logic. However, actions by individual actors only observe local social concerns i.e. the evaluation of stress towards the closest neighbor. As a result, the process proceeds as an animated sequence of globally complex spatial configurations. In addition, conflicting requirements may contribute to highly non-linear behavior. For example, actor A may prefer to be close to actor B while actor B aims to be far away from actor A. Merging this local concern with impact from neighboring actors, complex following or push-pull oscillatory behavior might emerge. One may think of PPM as a complex dynamical system that, according to the specification of particular social preferences, will produce spatiotemporal patterns of considerable intricacy.

2.2 The Actor Model

An extended version of the PPM called *Actors* has been used to simulate collective musical improvisation (Beyls 2010). In this work, we propose to use the Actor model as *organizational paradigm* in concatenative sound synthesis. In the Actor model, the user does not control the system. Instead, the user influences the outcome of an otherwise self-organizing social system. In other words, we interfere with the system's innate behavior in two possible ways. First, a HP virtually present interacts with the actors who, in turn, acknowledge the HP's social preferences. The HP interacts with the system via a control interface (e.g., MS's Kinect or Nintendo's Wii) that maps the actions into the space. The system is influenced only locally but might entail the emergence of complex patterns. The other possibility is to have the HP conceptually outside the actor society but able to adjust global parameter settings. So the interaction happens via the parameter settings of the system. The current implementation, reported here, documents the second method.

Fig. 1a Representation of the affinity matrix.

	A_1	A_2	A_n	A_N
A_1	0	$a_{(1,2)}$	$a_{(1,n)}$	$a_{(1,N)}$
A_2	$a_{(2,1)}$	0	$a_{(2,n)}$	$a_{(2,N)}$
A_n	$a_{(n,1)}$	$a_{(n,2)}$	0	$a_{(n,N)}$
A_N	$a_{(N,1)}$	$a_{(N,2)}$	$a_{(N,n)}$	0

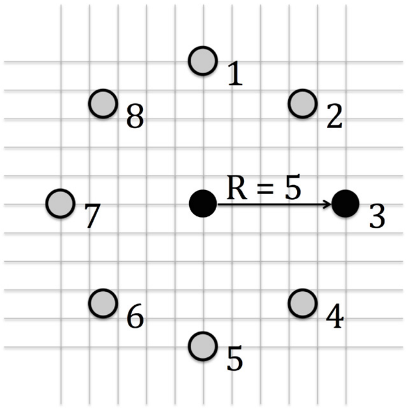


Fig. 1b Representation of moving actor.

The dynamic scope of the system is conditioned by two parameters; (1) the *affinities-matrix* (Figure 1a) specifying the ideal distances of every actor towards every other actor and (2) a *sensitivity parameter*, a value local to every actor. Sensitivity level is a single scalar value, private to the actor. It specifies the actor's sensitivity to any other actor, irrespective of the sensitivity of the other actor. Sensitivity conditions a distance threshold that in turn conditions interactions with a list of temporary neighbors.

Intuitively, it is easy to see a connection between the range of values and their diversity of values in the matrixes and the complexity of the ensuing spatiotemporal behavior. For example, given roughly equal values in the *affinities-matrix*, all actors will relocate to be at equal distances.

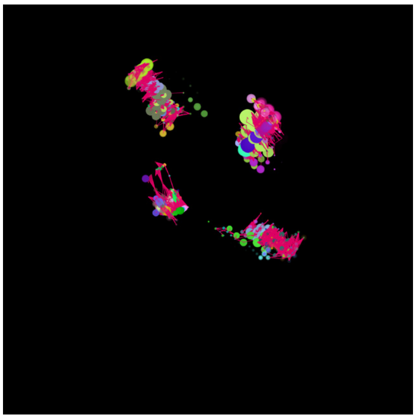
In every process step, the affinities-matrix is consulted to compute a list of new potential positions according to figure 1b; eight potential locations, at a 5-pixels radial distance, are considered relative to the position of the actor. However, only actors that are close enough are considered neighbors, that is, when their distance is within the sensitivity range currently expressed in the sensitivity value of the perceiving actor.

A wide range of spatiotemporal phenomena is generated from the specification of individual matrix values. Such a control structure is aesthetically attractive because the HP has the impression of interacting with an intricate system, whose behavior is only partially understood. The causal link between matrix and behavior is non-trivial, however it is perfectly coherent and offers structural integrity. Although individual actor behavior is unpredictable, the system nevertheless offers a strong overall impression of coherent performance.

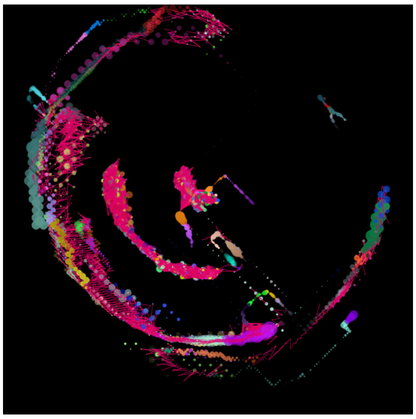
2.3 Mapping between human performer and system dynamics

Mapping aims to create specific functional relationships between gestural information and musical responses. Conventional approaches to mapping are deterministic yielding predictable results. Mapping typically generates musical responses as selected from a user-designed palette of options. For example, circular gestures always map to loudness changes. Given an aesthetic orientation favoring unpredictability and surprise, the concept of deterministic mapping is problematic. The Actor model suggests an alternative; the human performer interferes with the parameters affecting system dynamics - not unlike Sal Marirano commenting on him playing the SALMAR Construction: "it was like driving a bus" (Chadabe 1997).

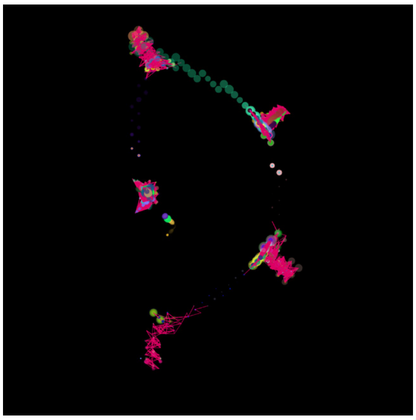
Fig. 2 Spatial configurations resulting from different parameter settings for the Actor model



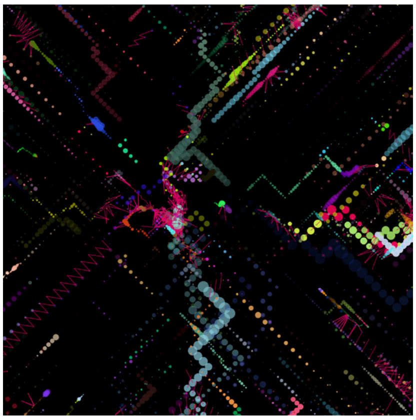
a) Local oscillations



b) Circular movement



c) Basins of attraction



d) Radial configuration

Figure 2 displays a collection of 4 snapshots, momentary spatial configurations as captured in a continuous animated process, window size is 1000 by 1000 pixels. Social affinities are set at random in the range of 50 to 500 pixels, whereas actor sensitivity is fixed (at a radius R = 5 pixels) in this experiment. Each image displays the configuration after 100 iterations (typically more), illustrating how different complex spatial configurations might emerge from different affinities. Only the affinities-matrix is occasionally slightly modified while the process is running.

In Figure 2a, all actors coalesce into four specific locally oscillating configurations. The effect of the forces of attraction and repulsion merges into a stable spatial pattern. Circular movement is clearly seen in Figure 2b with parallel trajectories showing evidence of attraction and repulsion cancelling out. Five major islands of activity emerge in Figure 2c while a spatial explosion occurs in Figure 2d.

Figure 3 illustrates the spatiotemporal behavior of 10 actors in a total agency of 50. The vertical axis shows the position along the X-axis and the horizontal axis shows iterations. We explore the behavioral scope of the system through interactive modification of the affinities-matrix. As the actors interact their trajectories oscillate between quasi-periodic and irregular. We end up having a control structure of high *plasticity,* its spatiotemporal complexity morphologically blending in the sound application to be discussed in the next section.

The Actor system implies two presentation modes; as a large-scale, projected real-time audiovisual installation and as a machine-mediated solo performance. It is implemented in two concurrent processes, agency behavior, parametric control and visualization is written in JAVA. A second process handling sound synthesis receives control data from the Actors via Open Sound Control (Schmeder et al. 2010).

3 Musical Application

In this section, we briefly introduce concatenative sound synthesis and earGram, the application used in this work. The discussion covers how the sound features capture perceptual qualities of the sounds, how to create sound spaces using the features as dimensions, and how different spatial configurations of sounds result from different dimensions.

3.1 Concatenative Sound Synthesis

Historically, concatenative sound synthesis (CSS) can be grouped with other sample-based techniques such as micromontage and granular synthesis which originated from the early *musique concrète* experiments. Briefly, CSS creates “musical streams by selecting and concatenating source segments from a large audio database using methods from music information retrieval” (Casey 2009). To a certain extent, CSS can be understood as being an extension of micromontage and granular synthesis towards a higher degree of automation.

What's unique about CSS in relation to other sample-based techniques is the annotation layer of the segments database, which not only provides the user with a good description of the audio source content, but also allows him/her to adjust, organize and re-synthesize the temporal dimension of the source in refined ways. Segment annotations include features automatically extracted and grouped into a single vector with the help of low-level audio descriptors, in a similar fashion as the audio-annotation layer of the MPEG-7 standard (Kim et al. 2005).

The sonic output of CSS systems depends on the audio descriptors used to organize the audio segments. The descriptor values define the spatial configuration of the segments, defining neighborhood relations and relative distances. For example, two segments might present similar loudness values at different pitches, which would place them close together along the loudness dimension but far apart along the pitch dimension.

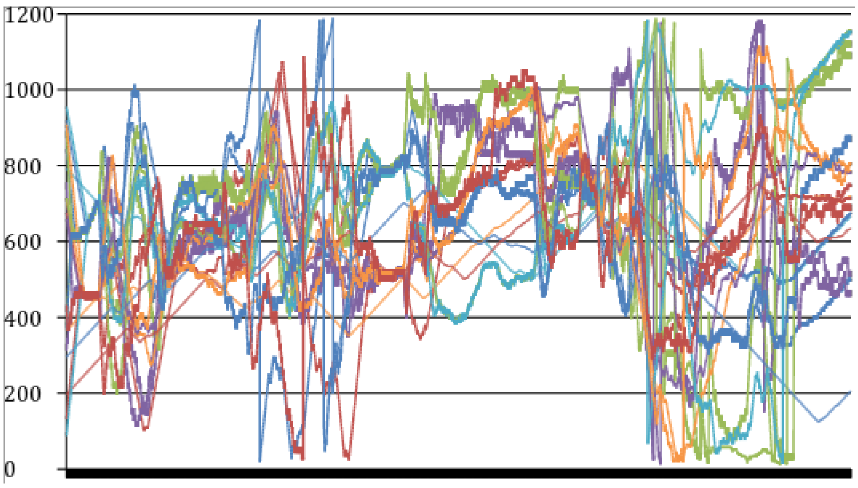


Fig. 3 Illustration of the spatiotemporal behavior of the Actor model. The vertical axis represents the position of each agent, while the horizontal axis represents the iterations. The curves illustrate how the trajectories can oscillate between quasi-periodic and chaotic paths, revealing intricate patterns.

3.2 EarGram

EarGram (Bernardes 2013, 2014) is an open-source and freely available application created in Pure Data for the real-time creative exploration of concatenative sound synthesis (CSS).[1] EarGram extends CSS, first attributed to Schwarz (2000), with new possibilities for generative audio by adopting strategies from both algorithmic-assisted composition and music information retrieval (MIR). The latter strategies are responsible for (i) segmenting an audio stream into elementary units, (ii) describing the most relevant features of the segments, and (iii) extracting patterns from the resulting collection of segments. Additionally, the system unpacks MIR terminology and concepts to a more adapted usability for musicians by relying on musicological and psychoacoustic

1 The software along with its documentation and many sound examples are available at: https://sites.google.com/site/eargram/.

theories, and presents most of processing stages of the systems in an intuitive manner, mainly through visualizations. The set of MIR tools adopted in earGram constitutes a valuable aid for decision-making during performance by revealing musical patterns and temporal organizations of the database, which are then used to represent audio in common algorithmic-assisted composition techniques.

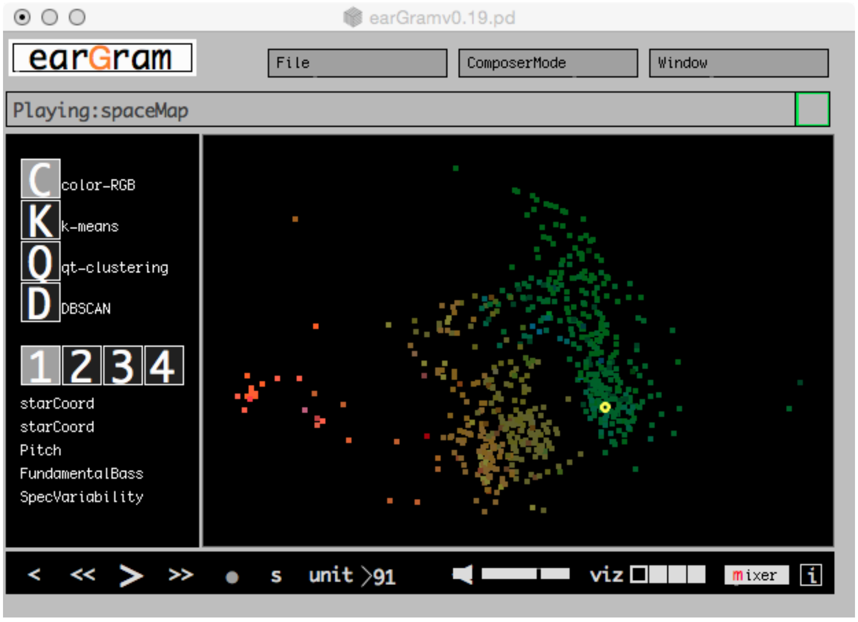


Fig. 4 2D plot of the speech sound database used in musical component of the system.

EarGram includes four generative modes: spaceMap, soundscapeMap, infiniteMode, shuffMeter, which cover a wide range of musical applications, such as the automatic generation of soundscapes, remixes, and mashups, to cite a few. Of interest here is the spaceMap mode, which is used to interact with the Actor model adding a sonic layer that offers musical functionality. The interface of spaceMap is shown in Figure 4 as a plane whose axes can be assigned to single audio descriptors or linear combinations of them. For example, the vertical axis might be loudness and the horizontal axis might be pitch. Each sound segment is represented by a (square) point in space, and their spatial organization is defined by their sound qualities (as measured by the descriptors). The visual representation of the database is used to play sound segments in the descriptor space as spatial trajectories. Hovering the mouse pointer (round point) plays the sound that is closest in the space. So, in the example, sliding the pointer vertically upward would play sounds that are louder and horizontally to the right would play sounds higher in pitch. Diagonal upward right-hand movement would play sounds with increasing pitch and loudness. While small movements synthesize similar sounding segments, larger movements pick sounds with greater sonic differences. SpaceMap allows the creation of highly controllable sonic textures driven by the user.

4 Using the Actor Model to Drive earGram

4.1 Speech Sounds as Metaphor for Social Interaction

Following the concept behind Gold's PPM's and the Actors model, we chose to work upon a database of multi-linguistic speech sounds to better express the idea of social interaction in the musical component of our system. Our aim was to represent social interaction, and particularly the affinity among individuals, by the perceptual proximity of speech sounds. Meaning that if the Actors' "society" reaches a stable configuration, the sonic response of the system should reduce the amount of variation to a minimum and, on the contrary, highly deviating configurations should result in a high level of sonic variation. Between these two poles there is a map that regulates the degree of variation in the speech sounds.

The speech sounds were retrieved from the UCLA Phonetics Lab Archive,[2] which includes both native female and male speakers of different languages, such as Bulgarian, Dutch, Estonian, Javanese, Nepali, Portuguese, Zulu, among others. After some basic sound editing to improve the sound file quality, including filters, equalization and noise removal, earGram automatically segmented the collection of speech sounds into short snippets of 200 ms each. Then, we solved the most crucial pre-processing stage of our database creation: the selection of a set of audio descriptors to represent our segments in the system.

The analysis of the database segments comprised two main tasks. First, we manually restricted the set of available audio descriptors to a sub-set of audio features that included: noisiness, pitch, brightness, spectral width, and sensory dissonance. Then, we weighted the set of selected audio descriptors to adjust their contribution in the feature space. Weights were automatically assigned according to the computed variance of each of the selected descriptors. By reducing the number of audio descriptors and weighting their contribution, we not only discarded redundant information for the analysis of speech segments, but also enhanced the computation of their perceptual similarity, which consequently improves their visual representation on the interface.

In order to plot the segments represented by their multidimensional vectors in a 2-D space, allowing to physically navigate their representation, or for the purposes of this work, to map to the 2-D visual representation of the Actor model to our space, we reduced the dimensionality of the segments' features vector to two

2 http://archive.phonetics.ucla.edu, last access on 7 January 2015.

dimensions using the algorithm star coordinates, first proposed by Kandogan (2000) and used in the scope of CSS by Bernardes et al. (2013). Figure 4 shows a 2-D plot visualization of the database whose axes are a linear combination of the aforementioned audio descriptors.

4.2 Integrating the Systems

After the database creation, we tackled the mappings between the Actor model and earGram's spaceMap, i.e. the visual and musical components of our system. In spaceMap, synthesis is typically controlled by defining trajectories in earGram's interface with the mouse. EarGram then retrieves the closest unit to the mouse position and synthesizes the selected segment with a Gaussian amplitude envelope. In our work, we replaced the mouse control by the position of each Actor in the space, given by its X and Y coordinates. This rather simple mapping strategy is effective in the sense that the segments plotted in the spaceMap interface are organized according to their perceptual distance. Therefore, Actors with high affinity values are close together in the descriptor space, so they will trigger similar sounding units, resulting in affinity being related to perceptual similarity. A video with an example of the final system is available at https://vimeo.com/118500562. In this example, there are 50 actors, the affinity matrix was initialized with low values, and the sensitivities are initialized all at 5 pixels.

Given the large amount of data sent to earGram via the OSC protocol, we ran into both technical and aesthetic problems related to the rate of transmitted data. Not only did the network block information in unclear ways, but also large amounts of data were incompatible with our sample-based technique, resulting in a constant over saturated mass of synthesized grains. To minimize this problem, we adjusted both the video frame rate to 25 frames per second (the same rate at which the location of Actors is computed) and imposed a different clock to control the rate of data sent over the network (every 800 ms).

Given the large number of data points received every 800 ms, we decided to store the Actors' location in memory and sequentially read them at equidistant time intervals within the 800 ms. Therefore, we hear a new segment every n ms, which equals the total number of Actors divided by 800 ms. Finally, we added an extra processing layer that looks at the overall stability of the Actors in the space, computed by measuring the flux of information at every received package of information, and mapped the value to the wet-dry parameter of a spectral freeze audio effect based on Paul Nasca's Extreme Sound Stretch algorithm[3] in earGram.

5 Discussion

A user engages with the proposed system mainly via modification of the parameters (global affinity matrix and local sensitivity values) in the Actor model, exerting influence on the dynamic behavior of the system. This indirect method for influencing the system behavior has implication on both the visual and sonic components of the system along two conceptual dimensions, level and extent of activity. The level of activity is related to the displacement of the actors, ranging from stationary to highly dynamic. The extent of activity refers to the distribution of the actors in the plane, which can vary between highly concentrated and spread out.

However, other decisions also influence the sonic outcome, such as selection and pre-processing of the sound source material, selection of the features used to define the dimensions of the space in earGram, and the affinities and sensitivities for the Actor model. In general terms, the source material determines the range of sonic possibilities. Speech sounds will produce a different outcome than instrumental, environmental, or synthetic sounds. The features have a direct impact on the distribution of the sounds in the plane in earGram. Changing the features will reorganize the same sounds according to different perceptual similarities, such that the same trajectory will generate a different sonic outcome. In this section, we will discuss the impact that each decision has in the aesthetic result.

In general terms, the spatiotemporal behavior of the system is determined by the level of social stress, which, in turn, depends on the magnitude and homogeneity of the affinities and sensitivities. High affinities result in strong attraction between actors, while low affinities generate repulsive forces. Homogeneity in the affinity matrix also impacts the global dynamic behavior. Whenever the user sets all the affinities to the same value, the level of activity decreases resulting in point attractor behavior. Heterogeneous affinity values entails complex dynamic behavior. The sensitivity also plays an important role in the dynamic behavior of the actors because it determines the radius R of influence of the affinity values $a_{(n,m)}$ between Actors A_n and A_m (see fig. 1a). High sensitivities force the actors to consider distant neighbors, while low sensitivities cause the actors to only interact with nearby neighbors.

For example, high magnitude homogeneous affinity values with high sensitivity will likely result in all actors clustered in a point because they are all highly attracted to one another. Low magnitude homogeneous affinity with low sensitivity will likely result in a uniformly spread out configuration across the plane because all actors are equally repulsed by their nearby neighbors. Notice

3 http://hypermammut.sourceforge.net/paulstretch/, last access on 7 January 2015.

that both scenarios result in low levels of activity because the examples suppose a homogenous affinity matrix and nearly equal sensitivity values. Highly dynamic complex behavior is commonly achieved through heterogeneous affinities and sensitivities.

The sonic response depends on the level and extent of activity as well. On the one hand, the level of activity is responsible for the dynamic response of the system. Each spatial trajectory results in a sonic trajectory that translates as temporal variation of the corresponding sound texture. On the other hand, the extent of activity influences the diversity of the sonic response by exploring different regions of the sound space.

The level of social stress drives the visual and sonic components of the system in symbiosis. The more complex chaotic oscillatory behavior of the actors, the more heterogeneous is its sonic response. Stable configurations result in sound textures with little variation. In other words, the actors' dispersion is related to the variability or "spreadness" of the sound segments selection, which equate with the level of coherence of the resulting texture, due to the organization of the segments on earGram's feature space. In between the two poles a wide and virtually endless range of possibilities exists.

Another interesting feature of the matrix-based control structure is the synthesis of smooth trajectories when one of more values fluctuates in the sensitivities-matrix. Since actors move through the consideration of a step-by-step evaluation process, changes gradually accumulate towards a spatial niche of lower social stress—the pull towards the basin of minimum stress decreases as a function of the distance of the actor from that location. In addition, considering one actor, since all its neighboring actors are all engaged in the same process, global behavior crystallizes into trajectories of considerable plasticity—the system produces smooth waves of spatiotemporal patterns. These smooth trajectories of actors in the visual domain are then mapped to the pointer position responsible for selecting audio segments in earGram's organized database visualization. The resulting sonic feedback matches the dispersion/cohesiveness and continuity of both the overall visual representation and the trajectories of individual actors. Furthermore, stable spatial configurations of the Actors society is further distilled into a blurred sonic texture obtained through spectral "smoothing" and filtering. The longer the actors' inactivity, the blurrier the texture becomes and the fewer spectral peaks are synthesized, thus reinforcing in the sonic domain the spatial configuration of the visual component of the system.

A fundamental contribution of this work in relation to its previous version, or for this matter any related work in auditory display and sonification, is the use of concatenative sound synthesis, an innovative sample-based synthesis technique, at the core of the software earGram. The integration of earGram with the Actor's model not only offered us more plastic and expressive sonic results in relation to related approaches—which tend to focus on additive, subtractive or physical synthesis models—but also allowed us to better match the conceptual basis of the system through the synthesis of speech sounds. By adopting a fixed database configuration in earGram, we favored one robust solution over a myriad of possibilities offered by the system. However, the current integration of both systems allows a user to easily experiment with different audio sources or even different feature spaces (i.e. database organization in the interface), while maintaining the same structural mapping, interactive behaviour, and to a certain extend the aesthetic basis. While adopting a different audio source has a greater impact on the sonic result, changing the feature space that organizes the audio segments database will offer a lower degree of variability, which equates in musical terms to the creation of variation of the same musical material. Ultimately, the positive outcome of this work spurs experimentation on sample-based techniques driven by a-life behaviour.

6 Conclusion and Future Work

Multi-agent systems commonly exhibit complex behavior after multiple local interactions following simple rules. The dynamics of self-organizing systems has been extensively explored aesthetically in artistic settings. Here, we use the Actor model of social interactions to control a concatenative synthesis engine called earGram in real time. The self-organizing behavior of the Actor model was designed to be aesthetically interesting visually, exploring the space as complexity emerges from the interactions. This visual complexity is used to aesthetically explore the feature space in earGram, whereby spatial trajectories become gradually evolving sonic textures. Trajectories are a powerful way to control earGram creatively because the spatial configuration reflects perceptual relationships among the sounds. The Actor model provides multiple trajectories, each controlling a sound texture in parallel, which result in an intricate and ever-evolving sonic tapestry.

User interaction is essential to explore the sonic result. Currently, the user interacts with the system by changing the parameters affinity and sensitivity that control the dynamic behavior of the Actor model. We plan to enhance the interactive feedback loop with a gestural device, such as MS Kinect. The gestures can be used to change parameter values in real-time. The sonic feedback would be used as system response to the interferences. The performer affects the visual and sonic output indirectly since the gestures do not control the system configuration, only the system parameters. More interestingly, the human performer can use a virtual presence device to interact directly with the actors. In this case, the human performer becomes the external perturbation that continuously upsets the states of equilibrium of the system driven by aesthetic judgments.

Acknowledgements

Research reported here is supported by MAT, The Media Arts and Technologies Project, NORTE-07-0124-FEDER-000061, and financed by the North Portugal Regional Operational Programme (ON.2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

References

Bak, P. *How Nature Works: The Science of Self-Organized Criticality*. New York: Springer, 1996.

Bernardes, G. *Composing Music by Selection: Content-Based Algorithmic-Assisted Audio Composition*. PhD dissertation, University of Porto, 2014.

Bernardes, G., Guedes, C. and Pennycook, B. Eargram: An application for interactive exploration of concatenative sound synthesis in Pure Data. In M. Aramaki, M. Barthet, R. Kronland-Martinet and S. Ystad, (eds), *From Sounds to Music and Emotions*, *LNCS,* 7900, 110-129. Berlin-Heidelberg: Springer, 2013.

Beyls, P. A Molecular Collision Model of Musical Interaction. In C. Soddu (ed.), *Proceedings of the Generative Arts Conference*, Milan, Italy, 2005.

Beyls, P. Structural Coupling in a Musical Agency. In E. Miranda (ed.), *Artificial Life and Music*, Evanston, WI: A-R Editions, 2010.

Beyls, P. Autonomy, Influence and Emergence in an Audiovisual Ecosystem. In C. Soddu (ed.), *Proceedings of the Generative Arts Conference*, Rome, Italy, 2012.

Blackwell, T., Bentley, P. Improvised music with swarms. In *Proceedings of the 2002 Congress on Evolutionary Computation*, 2, 1462-1467, 2002.

Caetano, M., Manzolli, J. Von Zuben, F. Self-Organizing Bio-Inspired Sound Transformation. Applications of Evolutionary Computing. *LNCS*, 4448, 477-487, 2007.

Camazine, S., Deneubourg, J-L, Franks, N., Sneyd, J., Theraulaz, G., Bonabeau, E. Self-Organization in Biological Systems. *Princeton Studies in Complexity*, 2003.

Casey, M. Soundspotting: A new kind of process? In R. Dean (ed.), *The Oxford Handbook of Computer Music*. New York, NY: Oxford University Press, 2009.

Chadabe, J. *Electric Sound, The Past and promise of Electronic Music*. Upper Saddle River, NJ: Prentice-Hall, 1996.

Dittrich, Ziegler & Banzhaf. Artificial chemistries, A review. *Artificial Life*, *7*(3): 225-275, 2001.

Gold, R. *The Plenitude*. Cambridge, MA: MIT Press, 2007.

Kandogan, E. Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In *Proceedings of the IEEE Information Visualization Symposium*, 2000.

Kaufmann, S. *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity*. Oxford University Press, 1995.

Kim, H.-G., Moreau, N., & Sikora, T. *MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval*. Chichester, UK: John Wiley & Sons, 2005.

Langton, C. *Artificial Life: An Overview*. Cambridge, MA: The MIT Press, 1997.

Miranda, E. Music composition using cellular automata. *Languages of Design*, 2, 1994.

Reynolds, G. Flocks, Herds and Schools: A Distributed Behavioral Model. In *SIGGRAPH 87 Conference Proceedings*, Anaheim, CA, 1987.

Schmeder, A., Freed, A., and Wessel, D. Best Practices for Open Sound Control. In *Linux Audio Conference*, Utrecht, Holland, 2010.

Schwarz, D. A system for data-driven concatenative sound synthesis. In *Proceedings of the International Conference on Digital Audio Effects*, 97-102, 2000.

Tidemann, A., Ozturk, P. Self-organizing Multiple Models for Imitation: Teaching a Robot to Dance the YMCA. *New Trends in Applied Artificial Intelligence, LNCS*, 291-302, 2007.

Todd, S., Latham, W. *Evolutionary Art and Computers*. Academic Press, 1992.

Ulanowicz, R. Complexity, Stability, and Self-Organization in Natural Communities. *Oecologia* (Berl.), 43: 295-298, 1979.