FEEDBACK IMPLEMENTATION WITHIN A COMPLEX EVENT GENERATION SYSTEM FOR EMERGENT SONIC STRUCTURES

Sinan Bökesoy

CICM, University of Paris VIII
Paris, France
sinan@sonic-disorder.com

ABSTRACT

This paper discusses the implementation of a complex event generation model with a simple feedback loop and its sound synthesis results while investigating the overall system behaviour. The system is based on the *Cosmos* model, which is a self similar structure, and distributes events on different time-scales with certain interdependency. The user intervenes with the system in real-time by inputting a live sound source and interacting with the user interface by controlling the parameters for the time scales *macro*, *meso* and *micro*. Because of the complex dynamic behaviour and modulation scheme, it is possible to create a timbre space of unique textures.

1. INTRODUCTION

Complex systems become increasingly important in synthesizing sonic structures. The understanding and controlling the phase space of such systems is a challenge. We have explored before the use of stochastic formulae inspired by Xenakis's models[1] for waveform synthesis, granular synthesis [1], other linear mapping models for organizing sonic atoms [2], and complex mapping models [3]. Applications using non-standard synthesis models with iterative functions [4] and interdependency in their structure [5] introduce non-linear mapping systems with dynamic behavior resulting to emergent phenomena. The interaction design is based on controlling their parameter space which controls the inner processes. By constraining the synthesis parameter space of dynamic systems for obtaining perceptually meaningful results, it becomes possible to produce a sound palette with desired specifications for compositional purposes. My perspective is gained by considering a macro-sound object as an evolving system of micro structures. The granular representations are especially useful in the analysis and perceptual modeling of dynamical events of peculiar complexity. [6] By introducing a hierarchy of multiple time scales for the event distribution process and modulation sources constructed with stochastic/deterministic functions, it is possible to form sonic creations that cannot be handled by traditional synthesis methods. The integration of gained experiences through different approaches into a coherent structure is my basic motivation.

First I will introduce the *Cosmos* model [7] shortly and follow with the structural facts such as how it gains the property of an iterative function by adding the recursive element becoming an interactive signal processing tool [8].

[1]Xenakis' works such as Achorripsis, AnalogiqueA/B, Gendy3.

1.1. About the *Cosmos* model

The *Cosmos* model is a real-time dynamic event distribution system, which generates sonic textures with an the event distribution process on multiple time scales. The discrete events of certain density are distributed in a time space with their onset time and duration parameter, which are calculated with stochastic/deterministic functions (Figure 1).

Each event in the macro space defines the duration of a meso space, and the sub events are distributed inside this space length. The same organization is also true for the events of meso space and micro space. As new spaces are created, new events and therefore new sub spaces are distributed while constructing the interdependency between layers.

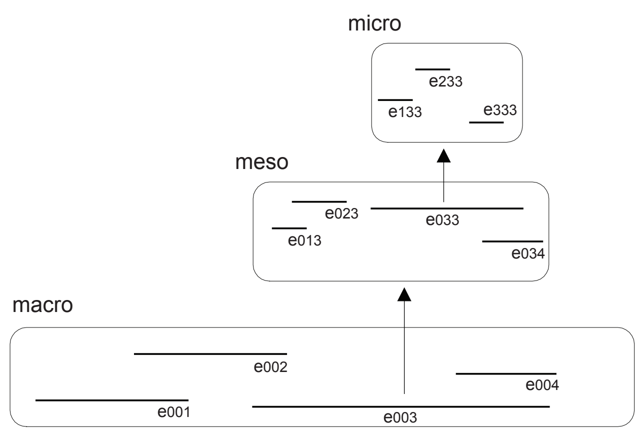

Figure 1: *Self similarity in the structure of the event generation mechanism of* Cosmos.

The modulation generators in 'Cosmos' are the stochastic modulation sources addressing the synthesis parameters, which are assigned to sonic entities defined within the micro event space. There are four different curve generators for each of the macro, meso and micro space events. Finally, we can combine together different modulation sources belonging to three different time scales, from macroscopic to microscopic, to obtain higher complexity. Each curve generator can be assigned to a unique destination, and if there are more than one modulation source for one destination, they will be superimposed as one modulation source with the complexity of representing the evolution of multiple time scales. The original version of *Cosmos* is based on the idea of assigning a waveform, sonic grain, to the micro events, where the meso and macro level distribution forms the sonic texture on multiple layers with various modulation schemes. The version presented in this paper will be explained in the next step.

2. ADDING THE RECURSIVE ELEMENT ON THE *COSMOS* MODEL

My purpose here is to convert the *Cosmos* model into a system, which transforms the waveform assigned for the sonification of micro-events with its inner dynamics, where it is able to generate a chain of actions without the intervening of human action. The only necessary human action is setting the initial conditions of the system and injecting a starting waveform (also it can be a continuous flow of audio).The audio input will be delivered on a delay line into the system (see Figure 3), which will be tapped with different delay times according to the equation (1).

$$X(n) = \sum_{l=1}^{macro_d} w_m(n) \cdot \sum_{m=1}^{meso_d} w_{mm}(n) \cdot \sum_{l=1}^{micro_d} f(n - D_{l,m,n}) \cdot w_{mmm}(n) \qquad (1)$$

$f(n)$ represents the input waveform as a series of sampled values. $macro_d$, $meso_d$ and $micro_d$ are event space densities defined by the user or by stochastic functions. The scalar $D_{l,m,n}$ representing the delay time in samples in equation (1) is calculated at the point of each macro-space initialization. This occurs with a mapping of the onset time values calculated for the micro events as in equation (2).

$$T_{on}\ e_{x,y,z} = \sum_{m=0}^{z} \sum_{l=0}^{y} \sum_{k=0}^{x} t_{on}\ e_{k,l,m} \qquad (2)$$

$t_{on}\ e_{x,y,z}$ in equation (2) is the onset time interval between $e_{k,l,m}$ and $e_{k,l,m-1}$. It can be calculated with statistical functions like gaussian, exponential, uniform, triangular, chauchy distributions or deterministic functions like arithmetic, geometric series or constant values. The functions use the space length and density as input parameters and distribute the onset time values and durations for the events inside the particular space. $T_{on}\ e_{x,y,z}$ represents the absolute time between the initialization of the macro space and the onset time of xth micro event in yth micro-space of zth meso-space.

This value will be mapped as the delay time in equation (1) by multiplying it with the sampling frequency which is 32 kHz in this case.

The delayed input waveform is multiplied with a window function $w_{mmm}(n)$ which is a trapezoid function with the duration equal to the micro-event length having the attack and decay times of 1 ms. The meso-events will be multiplied with $w_{mm}(n)$ and the macro-events with $w_m(n)$ as in equation (1). This procedure implements a complex delay network created by the event data which is generated following a top-down approach from macro to micro structure. The bottom up organization of the delay network will be realized by sampling this delay line with the duration defined by the length of micro and meso spaces of *Cosmos* and passing the audio to the delay lines on higher levels (Figure 3). Thus, each organization of micro events produces the smallest delay network and will be passed as micro-space waveform data to the meso-space.

In Figure 2 one can notice an input waveform belonging to a xylophone sample as organized by the micro-space distribution with a space length of 100 ms. We remind here that each micro-space corresponds to a meso-event. Then the meso-events organize the meso-spaces corresponding to macro-events. Finally the macro-events (meso-spaces) together realize a macro-space. The audio output is delivered back to the micro-event with a connection of the macro-space output to the input delay line which creates a feedback loop. In Figure 2, we see a direct mapping of the input material to the micro-events forming up the meso-event.

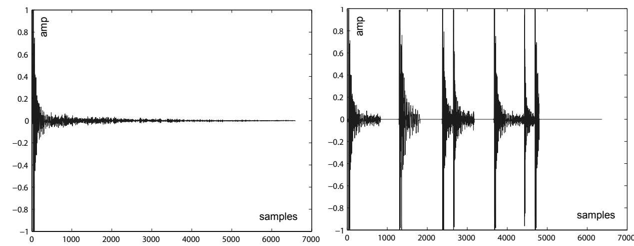

Figure 2: *The input waveform above has been mapped by the micro-space distribution onto the micro-events. The window function applies a 1 ms fade out on each event.*

This introduces local densities contributing rhythm to pitch phenomena depending on the distribution functions in the event spaces, but also the possibilities of *Cosmos* present certain modulation source generators which are capable of introducing morphological change on the waveform data.

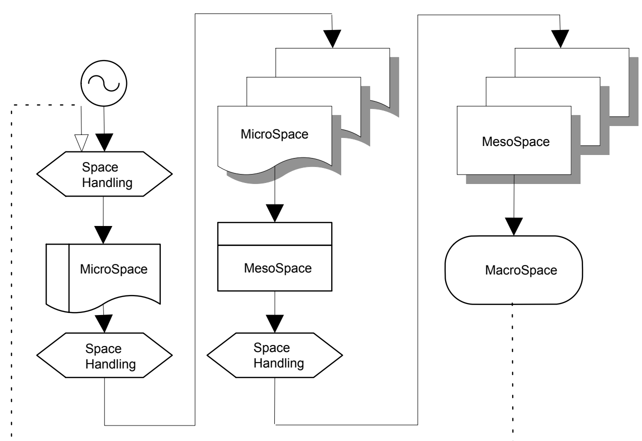

Figure 3: *This is the schematic overview of the audio routing and the event space organization inside the application. The audio enters top-left. The dotted line is the feedback line.*

The applied modulation sources on micro, meso and macro events individually are as following:

For *macro-events*:	delay tap mod., intensity mod., spectral filter spatialization
For *micro-events*:	delay tap mod.,intensity mod.,
For *meso-events*:	delay tap mod.,intensity mod., spatialization,spectral filter mod.

These are continuous modulation sources applied to each event along its duration. They are constructed with mathematical functions inspired by the *Achorripsis* model of I. Xenakis [1] (see Figure 4). Depending on the speed of the modulations applied on the audio data, the effect can be non-linear such as the in the delay tap modulation, which can produce complex pitch modulation in this case. In this version, I had to consider the CPU usage in order not to compromise the real-time aspect of interaction.

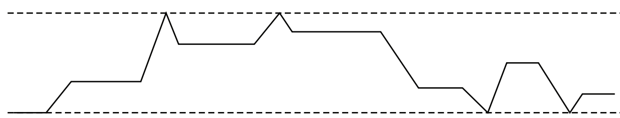

Figure 4: *A continuous modulation constructed with a stochastic function and lo-hi barriers witch fold the curve back from the limits as in the Xenakis models.*

The application of the model is implemented on Max/MSP[2]. The delay network is constructed with *tap.in/out* structure. All the duration and onset time information is calculated on Java-script. The parameters from the user interface are updating the variables in the script during the real-time operation, and the calculated event parameters are written to the memory locations, which are accessed by the MSP patch accompanying for DSP routines. The information necessary to render the audio in these functions are delivered in audio-rate to the relevant MSP objects for precise handling of the data. Each event is synchronized with at audio-rate *adsr* (AttackDecaySustainRelease) function serving as a window function with the duration equal to event duration. Those frames are captured and organized as in Figure 3.

I have limited the macro space density with 2 macro-events, the meso-space density with 4 meso-events and the micro-space density with 5 micro-events. Hence, inside one macro-space there can be $2 \times 4 \times 5 = 40$ micro events at once and the accompanying modulation sources would have the quantity equal to $2 \times 2 + 8 \times 4 + 40 \times 2 = 116$. This gives an idea about the complexity the system is offering between one waveform input and the macro sound output. We could substitute the windowing functions in the equation (1) to include the modulation functions in macro, meso and micro-space domains.

The system in Figure 3 proposes a single audio input/output structure with controllable parameters. A complex system is not *linear time-invariant*. Time invariance here means that whether we apply an input to the system now or t seconds from now, the output will be identical, except a time delay of the t seconds. We could force the model to periodic behavior. We can describe the next macro-space according to the input parameters explicitly when using only deterministic functions like geometric series or fixed values for the decision of the onset time and duration distributions which effect directly the delay time distribution and deterministic modulation sources. We would know then how the events are distributed and modulated inside one macro-space, which can be also represented as the phase space of the macro-space with regarding its input parameters and state controls.

2.1. A case study using the recursive *Cosmos* model

Now I will explicitly formulate a special deterministic case as an example and explain how the system becomes an iterative function by applying the feedback loop. The onset time decisions in macro, meso and micro events are made with arithmetic series function and the event duration is defined with the distance coefficient a of the arithmetic series of Equation (3). With the use of the discrete time domain equation, it becomes possible to watch the phase space with continuing the iteration on the equation. Consider the arithmetic series with:

$$y_n = 0 + a + 2a + 3a + \cdots + na \tag{3}$$

We define an event space with density n, where y_n equals to the event space length. The macro space length M_{Cl} and the densities macro, meso and micro spaces (d_m, d_{mm}, d_{mmm}) are deterministic input parameters for the system.

$$M_{Cl} = \frac{d_m(d_m+1)}{2} a \Longrightarrow y_{M_n} = \frac{n(n+1)M_{Cl}}{d_m(d_m+1)} \tag{4}$$

y_{m_n} in equation (4) will provide us the macro space event onset times, such as for $1 \leq n \leq d_m$:

$$e_{Mon_n} = y_{M_{n-1}} \text{ and } e_{Mdur_n} = a = \frac{2M_{Cl}}{d_m(d_m+1)} \tag{5}$$

For example a 1 sec. long macro space with 3 events would have the event onsets at 0 ms, 166 ms and 500 ms with the event durations of 166 ms, according the arithmetic series distribution function used here. The event duration of the macro events define the meso space length and the meso space event durations define the micro space length in the self similar structure of *Cosmos*. Therefore it is easy to substitute the equations to find the meso and micro level distributions. For the meso space, the onset and duration equations are:

$$e_{MMdur_i} = \frac{4M_{Cl}}{d_m(d_m+1)d_{mm}(d_{mm}+1)}; \; 1 \leq l \leq d_{mm}$$

$$y_{mm_l} = \frac{2l(l+1)M_{Cl}}{d_m(d_m+1)d_{mm}(d_{mm}+1)} \; e_{MMon_l} = y_{mm_{l-1}} \tag{6}$$

The micro, meso and macro spaces are modulated by their individual modulation sources, however in this example we discard the modulation sources and perform the iteration for an impulse input. In order to define the delay times for the micro events, we need to formulate the absolute onset time according to Eq. (2):

$$e_{Mon_n} + e_{MMon_l} + e_{MMMon_k} = e_{MMMon_{n,l,k}}(abs)$$

$$1 \leq n \leq d_m \qquad 1 \leq l \leq d_{mm} \quad 1 \leq k \leq d_{mmm} \tag{7}$$

Equation (8) represents the recursive delay network with the feedback gain G. A unity gain represents the direct connection of *Cosmos* output with its input. The experiment can follow by inserting an impulse function as an input and listen to the iterated y values.

$$f\left(x - \sum_{n=1}^{d_m} \sum_{l=1}^{d_{mm}} \sum_{k=1}^{d_{mmm}} e_{MMMon_{n,l,k}(abs)}\right) + G\, y(x-1) = y(x) \tag{8}$$

2.2. Sound examples and some interpretations

The first result is a mapping of the arithmetic series in the self-similar structure of *Cosmos* as a rhythmic pattern while $G = 0$ (*exampleA*[3]). The change in G won't destruct this pattern as new

[2]a programming environment distributed by *Cycling74*.

[3]The audio examples can be downloaded at URL: http://www.sonic-disorder.com/cosmosfdb-examples.zip

macro-spaces follow each other using the same mapping (*exampleB*). Hence, regarding the impulse response of this system, we observe the simplest case of sound morphology carried by the feedback loop to the input delay line. It is helpful to follow the transformation by ear, and do some interpretations according the hearing experience.

The audio signal is not being analyzed in order to create control signals and there are no functions which manipulate the system using this kind of analysis [7]. Any change in micro or meso-level will be reflected to the higher level which becomes the interdependency inherent in the system design. In *exampleC*, a uniformly distributed tap delay modulation has been applied to the meso events by keeping the same onset time and duration distribution and $G = 1$. The skeleton remains the same but the skin does change.

In *exampleD*, a stochastic delay time modulation is applied on the micro event level while keeping the settings in *exampleC*. Then in *exampleE*, more disorder has been created in the system by changing the macro and meso event durations with the exponential distribution. In *exampleF* a very low frequency sawtooth waveform with a single cycle has been delivered as the initial signal.

These examples show the response of the application to a short waveform input (a 100 ms xylophone sample) according to the change applied to its parameter space. After certain amount of iterations, we can hardly recognize anymore the departure sound having been injected to the system. Due to the system dynamics, emergent properties result textural sound phenomena which is evident in the *exampleC*, *D*, *E* and *F*. Such results are extremely difficult to obtain with *montage* or any other hand driven methods since the delivered complexity and the computation necessary is high.

I stressed myself to use short sounds as the input material and the model organizes a macro sound structure from its own sound output depending on its internal conditions at the moment it captures the output signal on the micro-level. It creates a skeleton of events organized on different timescales simultaneously. The overall texture becomes a complex one exhibiting various modulation sources transforming the audio material from micro to macro levels.

I heard many comments from my colleagues about how this system could successfully deliver an organic character like the sounds of unknown creatures especially even when the input material is very simple and the model transforms it quickly to various emergent forms within its dynamic structure.

3. PRELIMINARY CONCLUSIONS

With the combination of stochastic distribution mechanisms and the self similar structure of the *Cosmos* model, it is impossible to predict the precise output of any configuration of values. A tool has been developed in order to bring a periodic behaviour by freezing the macro level mechanisms but letting the system operate further on meso and micro levels.

In this case, the ear can easily follow and memorize the macro level outline of the sonic structure. It is also a quick process to add new dynamic functions or user defined rhythmic patterns to the Javascript code in order to control the onset time and event duration distribution. A new tool is also being developed, where one could morph between two distribution functions on the fly such as a mapping between determinism and indeterminism occurs. Many experiments should be tried to gain perceptual experience with such systems where the user interacts in order to explore the sonic output by controlling the parameter space externally.

I have tried to explain the implementation and the design aspects of a recursive structure based on the *Cosmos* model. The feedback of the sound output (macro level) into the system on the micro level adds a considerable gain to the emergent behaviour. In musical contexts, there are important aspects which cannot be left aside such as the question of interacting with a complex synthesis application and a powerful user interface for efficient mapping of the parameter space based on external manipulations [9]. One could transform this model into an instrument within real time implementation.

A continuous MIDI controller is very useful to assign its pots to various parameters in the application interface, as the operation with mouse is not very intuitive. As a composer, my intention is continuing this research by trying to realize compositional ideas by developing a mapping interface which translates the dynamics of acoustic sound sources/instruments to the parameter space of the model.

(The author would like to realize an augmented percussion instrument coupled with this model to process the audio/control data captured from the percussion by sensors and microphone.)

4. REFERENCES

[1] I. Xenakis, *Formulized Music*, revised ed. New York: Pendragon Press, 1992.

[2] C. Roads, "Sound composition with pulsars," *J. Audio Eng. Soc.*, vol. 49, no. 3, pp. 134–147, Mar. 2001.

[3] M. Hamman, "Mapping complex systems using granular synthesis," in *Proc. Int. Comp. Music Conf. (ICMC'91),* Montréal, Canada, 1991, pp. 475–478.

[4] A. Di Scipio, "Iterated nonlinear engines as a sound-generating engine," *Leonardo Journal*, vol. 34, no. 3, pp. 249–254, May 2001.

[5] P. Bowcott, "High level control of granular synthesis using concepts of inheritance and social interaction," in *Proc. Int. Comp. Music Conf. (ICMC'90),* Glasgow, Scotland, 1990, pp. 50–52.

[6] A. Bregman, *Auditory Scene Analysis*, 2nd ed. Cambridge: MIT Press, 1999.

[7] S. Bökesoy, "The Cosmos model, an event generation system for synthesizing emergent sonic structures," in *Proc. Int. Comp. Music Conf. (ICMC'05),* Barcelona, Spain, 2005, pp. 259–262.

[8] A. D. Scipio, "'Sound is the interface': Sketches of a constructivistic ecosystemic view of interactive signal processing," in *Proc. XIV Colloq. Musical Informatics (CIM-03)*, Firenze, Italy, 2003, pp. 259–262.

[9] I. Choi, "A chaotic oscillator as a musical signal generator in an interactive performance system," *J. New Music Research*, vol. 26, no. 1, pp. 17–47, Sept. 1997.