STRIA, BY JOHN CHOWNING: ANALYSIS OF THE COMPOSITIONAL PROCESS

Matteo Meneghini
CSC-DEI Università di Padova
menego@dei.unipd.it

ABSTRACT

Stria is a piece fully generated by the computer, with the creation of all the parameters needed to play each sound, starting by a certain number of input sessions, elaborated by algorithms. This paper contains the partial synthesis of the analysis of this piece; the analysis was conducted starting from the original algorithms, the listening of the piece and a few literature documentation. Direct communication with J. Chowning was important too.

1. INTRODUCTION

Stria was composed by John Chowning, at the Stanford University (USA), in 1977, while working at the Center for Computer Research in Music and Acoustics (CCRMA). After he discovered that frequency modulation could be efficiently applied to the synthesis of sound (1967-1971), he composed several works using the results of his research: together with *Stria* we remember *Turenas* (1972) and the later *PhonŒ or Phoné* (1981). Each of those compositions is intended to give value to a specific technique he had worked on: in *Turenas* he used his studies about spatialization to define the travel of sounds in a quadraphonic space, in *Phoné* he dealt with spectral fusion between sounds, using an algorithm applying frequency modulation to the synthesis of sung vocal tones. In *Stria*, as we will see, he used computer synthesis to interrelate the small-scale sound design to the whole composition's structure.

Before going into the details of the structure of this piece, it is important to get the basic knowledge about the numeric construction which is at the base of this work: the golden mean.

2. GENERAL PROPERTIES

In this section we will analyse the general properties which characterize *Stria*, starting with some basic definitions about the golden mean, and continuing with the description of the pitch space and spectrum division, of the instrument played and of the temporal structure of the piece.

2.1. The Golden Mean

Considering the geometric and architectonic origin of the golden mean (or golden section), we start considering a segment of length z=1, and look for its part x such as the ratio between z and x is equal to the ratio between x and the remaining part of the segment itself. To do this, we can consider the equality

$$\frac{1}{x} = \frac{x}{1-x} \quad (1)$$

Solving this equation for x, we find that

$$x = \frac{1}{2}\left(-1+\sqrt{5}\right) = 0.618 \quad (2)$$

We can then extend this result to the continuous proportion

$$\frac{1-x}{x} = \frac{x}{1} = \frac{1}{1/x} = \ldots \quad (3)$$

which numerically corresponds to

$$\frac{0.382}{0.618} = \frac{0.618}{1} = \frac{1}{1.618} = \ldots \quad (4)$$

In ancient times, the important ratio we have obtained this way was considered a rule of physical perfection. It is easily recognizable in many human works (eg. in architecture) and in nature too.

In music, the golden section represents (in a good approximation) a minor sixth, in western notation. As a matter of fact, an eight-semitones space is defined by the ratio

$$\sqrt[12]{2^8} \cong 1.6$$

which is near to the golden mean.

Another important property to remember is connected to the Fibonacci succession: each of its terms, starting by 0,1,2, is obtained with the sum of the two immediately preceding terms. In particular, it can be proved that the ratio between two consecutive terms of this succession quickly tends to the golden section. From this, we can easily say that the powers of G=1.618 are ordered in accordance with the Fibonacci succession, i.e. that the equation

$$G^n = G^{n-1} + G^{n-2} \quad (5)$$

is true.

These properties are very important in reading *Stria*, and must be remembered in the following analysis.

2.2 Pitch Space And Spectrum

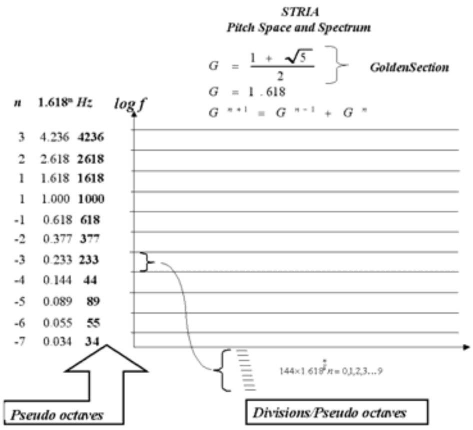
After a long series of experiments on FM synthesis, Chowning tried to discover an inharmonic ratio to re-define the concept of octave: he needed a ratio which generated FM synthesis components some of which are exactly powers of that ratio. After many tests (executed before programming) and fascinated by the sound of FM ratios $c:m=G^n:G^m$ where n and m are integer powers, he found that the golden section really had all the properties he was looking for (1974, Berlin). He re-defined the concept of octave (usually based on the ratio 1:2), using the ratio 1:G=1:1.618. Each pseudo-octave generated was then equally divided into 9 tones, by the factor

$$G^{\frac{k}{9}} \quad (6)$$

An eighteen tones division was also available, obtaining a sort of semitones. The pseudo-octaves used in *Stria* are generated around the central frequency f=1000 Hz, and the fundamentals of each octave are expressed by

$$G^{-3}f, G^{-2}f, G^{-1}f, f, Gf, G^{2}f \ldots$$

A good representation of the pitch space is given by the following figure (due to Chowning himself).



In this figure we can see the fundamental notes of each pseudo-octave, the division of each octave in 9 tones, and the properties of the golden mean. The fundamentals of each octave are connected to powers of G: the advantage due to the use of the golden mean is that this relationship is also linear.

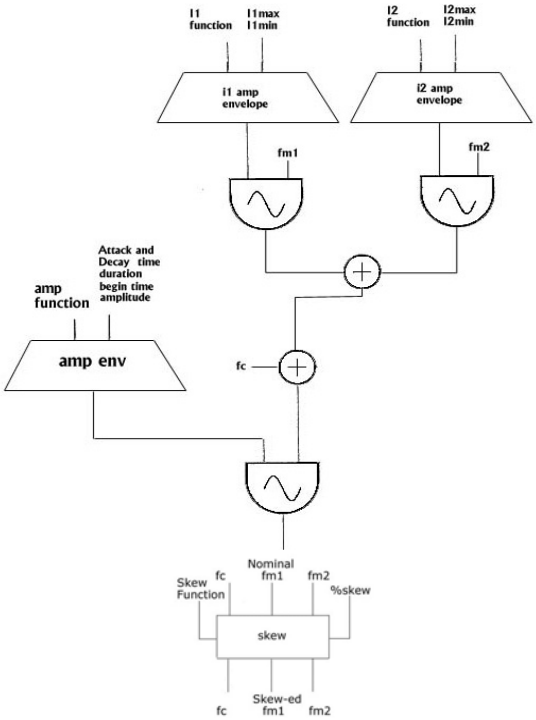
As we have seen, G is the limit of the ratio between two successive terms of a Fibonacci succession: then we can derive that by adding two following powers of G, we obtain another power of G, by the table

Power of G	Linear combination a+bG
$0.056=G^{-6}$	13-8G
$0.090=G^{-5}$	5G-8
$0.146=G^{-4}$	5-3G
$0.236=G^{-3}$	2G-3
$0.382=G^{-2}$	2-G
$0.618=G^{-1}$	G-1
$1=G^{0}$	1
$1.618=G^{1}$	G
$2.618=G^{2}$	1+G
$4.236=G^{3}$	1+2G

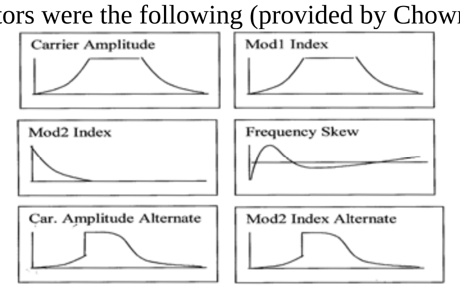
The spectral components obtained with the sum of powers of G, can be expressed by linear combinations a+bG. Chowning decided to use this property defining a carrier to modulator ratio for the FM synthesis equal to G: in this way the components generated were sums or differences between powers of G, which were also in a linear relationship a+bG with G. With this efficient mechanism, the whole pitch space was ordered in way that there was no component in discordance with the golden ratio. In *Stria* Chowning used eight pseudo-octaves, three above and five below the central frequency (f=1000 Hz): all these pseudo-octaves are used in the composition.

2.3 The Instrument

Using this efficient division of the audio spectrum, Chowning generated all the sounds with a unique instrument: starting with the input parameters, the algorithms generated 30 parameters for each instrument. The whole piece can be intended as played by a 26 instruments orchestra: each instrument generates one sound every time it is called to play; the sound played is different at every call, having different parameters used in the call itself. The basic scheme used for the instrument is the FM modulator, with double modulator: all the oscillators used were sine functions, and the modulators were summed to the carrier frequency. The sound was then shaped in time: Chowning applied envelope generators to the amplitudes of all the oscillators, thus varying the amplitude of the signals, and changing the spectral content of the sound in time. A light deviation (called skew) was added to the frequencies of both the carrier and the modulators in proportional way, to obtain a major liveliness and reality. The two modulators allowed Chowning to increase the spectral density without using large indexes. Large modulation indexes would have reduced the contribution of the carrier in the modulated sound, by reducing the zero-order Bessel function, which Chowning didn't want. The instrument can be represented by the figure



where the skew generator is represented in an equivalent way, considering in input the nominal values of the frequencies, and the parameters defining the skew; in output there are the skew-ed values of the frequencies, which will be applied to the sin oscillators. The amplitude envelopes used for the oscillators were the following (provided by Chowning)



There were two possibilities for the amplitude envelopes: the normal and the alternative one. In the normal case, the sound started as modulated by the second modulator, continued as a double modulator FM, became a FM sound modulated only by the first modulator, and finished as only the carrier. The alternative case was used at the climax of the piece, to produce a sort of *ssshBoom* effect (as Chowning called it): this effect was due to a rough variation of the second modulating index (generating a very rich spectrum), accompanied by a step variation of the carrier amplitude.

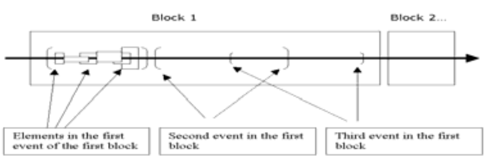
The skew function was defined by a small deviation at the attack of the sound, that becomes even smaller in a short time. It is difficult to hear this small deviation on a single sound, but it is easily recognizable in the superposition of many musical elements, as a sort of beating between components. The maximum amplitude of the skew function is determined by the frequency of the tone to play, using a low-pass function, in order to obtain a sharp deviation at low frequencies, and a small deviation at high frequencies. The human hear is more sensitive in frequency variations at high frequencies, than in low frequencies. This justifies the trend of the function defining the skew percent in frequency.

For each instrument is defined a set of three spatialization parameters: the reverb to apply to the sound, the apparent angle of the source and the apparent distance of the source. The last parameter is defined in accordance with the theory explained by Chowning in "*The Simulation of Moving Sound Sources*" (1971, [1]) and in "*Perceptual Fusion and Auditory Perspective*" (1990, [2]). In both these papers we find that the perception of the distance is based on the ratio between the direct sound intensity (varying with the distance) and the reverberated sound one (fixed with the distance).

Anyway, the spatial control used in *Stria* (no Doppler effect was used) was not intended to be precise: the sounds resulting using these parameters were similar to the ones that would be obtained in a reverberant cathedral, amorphous, big and undefined.

2.4 The Temporal Structure

Stria can be considered composed by a microstructure and a macrostructure. The whole composition is divided in blocks, a sort of input sessions, each of which is saved into three different files. Each block is composed of few events: each of these is defined by a set of input data, and is composed by a great number of elements (single sounds, played by single instruments), generated by the algorithms starting from the input values. The following picture represents the linear temporal composition of the piece:



The arrow represents the temporal evolution of *Stria*, the big rectangles are the blocks, and between brackets we can see the events; in the first event are represented also the elements composing it (by small rectangles). We can consider the elements as the atoms of which *Stria* is constructed. Each element is a single sound, and the succession and the superposition of these sounds is the whole piece.

This is the macrostructure of *Stria*: each element is defined by some microstructure parameters, defining its characteristics, as will be clear in a following section; some of these parameters are generated using the golden mean. While generating of the events the algorithms created the single sounds (elements) by the definition of their parameters:

Chowning used recursion to generate further sounds (child elements) superimposed to the original ones (parent elements); this will be discussed in a successive section. The whole piece is 17 minutes long: in the first part of *Stria* the intensity increases and after 10 minutes (number which approximately stands in the golden ratio with the total length), there is a climax, followed by a quasi-silence moment, after which the intensity grows again. The organization of the sound events in the time-frequency space is opposite to the traditional one: usually, in fact, the low pitch events are longer than the high pitch ones; in *Stria*, instead, the longer events have higher pitch, and vice versa.

The pitch of the sounds in *Stria* decreases towards the climax, and increases after this moment. Also the attack and decay time of each sound is determined by its frequency (e.g. an high pitch sound will have a slow attack).

3. THE PROGRAM

The program is written in SAIL (Stanford Artificial Intelligence Language, [3]), a language created in Stanford similar to ALGOL and PASCAL. It consists of a few procedures, called by a main program; the events are generated by the procedure EVENT2, receiving data from input, and calling other service routines to generate the frequencies (INHARM), the times (PROPORTION), the spatialization parameters (AZIM) and to write the output files (WRITE). Each call of the program defines one block, composed by few events: each of these events is defined by input parameters; the output of the program consists of three score files, one of which containing 30 parameters for each instrument to play.

3.1 The Frequency Generation

Frequency generation is managed by the procedure INHARM, called by EVENT2, when generating each element in the event. Each element created in the current event has a different value of the frequency of the note played, in accordance with the variation of a parameter (*num*) at every call of INHARM. It is the variation of this parameter that creates the melodic line of the event. In EVENT2 the frequency of each sound to play is generated by the expression

$$f = fff \cdot freq \cdot k \quad (7)$$

where *freq* is the base frequency of the event, i.e. the frequency of the fundamental note of the pseudo-octave on which the whole event is constructed (equal for every sound in the event); *fff* is the coefficient (scale frequency) that determines the note played by the current element in the event, by the product *fff*freq*, varying by element to element in the event (at every call of INHARM); *k* is a coefficient used to calculate the frequency to play on each oscillator by the product *fff*freq*k*. For example the carrier frequency is calculated from the note *fff*freq* by the expression

$$c = fff \cdot freq \cdot f_c \quad (8)$$

fff and *k* are multiples of G, and *freq* is given by an expression like G^j*1000. *fff* varies from element to element, at every call of the INHARM procedure.

For each event is defined a frequency space variable, by the expression

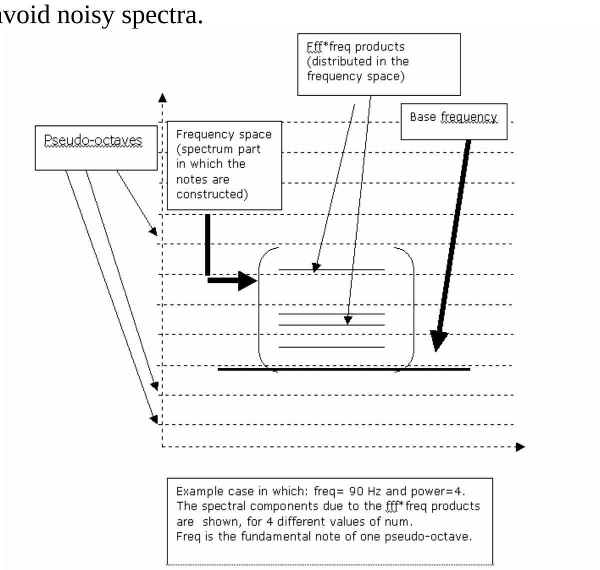
$$space = ratio^{power} \quad (9)$$

where ratio is equal to G=1.618, and power is integer (positive or negative). This variable represents the frequency space to be occupied of the event, i.e. the dimension of the spectral space occupied by the event. Power is the number of pseudo-octaves used in the current event (above or below *freq*, in accordance with its sign). Each element will play a note in one of these pseudo-octaves, that will be divided in 9 or 18 tones.

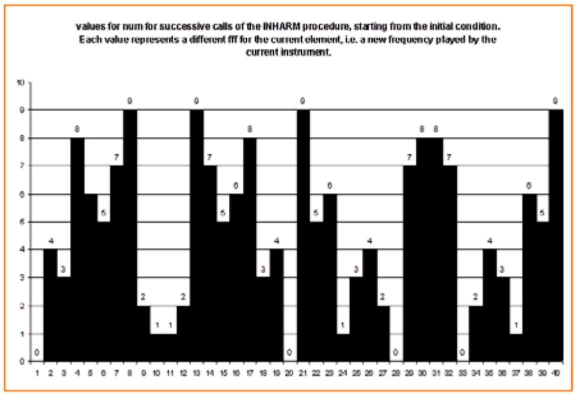
This division is done by the variable *fff*, which defines the note to play, and is calculated by the expression

$$fff = (ratio^{power})^{num / divx} = ratio^{\frac{power \cdot num}{divx}} \quad (10)$$

where *num* is different for every element generated, and is calculated by the service routine INHARM, called by EVENT2; *divx* is a variable which is equal to the number of divisions chosen for the frequency space (9 or 18). It is important to note that for $|power|=1$ Chowning didn't use the 18-notes division, as a way to avoid noisy spectra.



This figure shows the relationships between the parameters used to define the frequencies of each element in a particular case. Now we will see how *num* is generated, i.e. how the melodic line is created in *Stria*. Num is constructed by a table of 10 values, that creates a succession of number used to calculate *fff*. This succession, in the case of a 9-notes division, is periodic with repetition period 40 (if not re-initialized), meaning that the melodic line would be repeated only every 40 elements; in *Stria* there is no event with more than 40 (parent) elements; this means that the melodic line isn't repetitive (for the child element the base frequency is different from the parent's one, and no repetitive melodic line is possible, even though continuing to read the 40-period succession). The generation of the values of num in the subsequent events can continue to follow the succession (creating the periodicity) or re-initialize it from the initial values. This choice (done by input) is useful to increase the melodic variety in the piece. In the case of an 18-notes division of the frequency space occupied by the event, the table is read in a different way, generating a succession with period 20, of values of num comprised in the range 0..18; *divx* is equal to 18, in this case. In this way Chowning allowed some events to generate elements playing a kind of semitones. It is important to note that the recursive sounds (child elements) can be constructed only on a 9-division space, while the parent sounds can have *divx* equal to either 9 or 18.



The last figure represents a particular case of a melodic line (of values of num) generated starting by the initial conditions of INHARM for 40 successive elements in an event, for a 9-notes division of the frequency space. From the values of *fff* and *freq* for the current element, procedure EVENT2 calculates the carrier frequency and the second modulator by the formulas

$$c = fff \cdot freq \cdot f_c \qquad m_2 = fff \cdot freq \cdot f_{m2} \quad (11)$$

where f_c and f_{m2} are the frequency coefficient already explained. The determination of the first modulator frequency is quite different, and for this oscillator there is the possibility to maintain the same first lower (or upper) side frequency constant for all the elements in the event, besides the traditional way (which would create different components for all the elements in the event). In this case the formula used (for a constant lower side) is

$$f_{m1} = fff \cdot freq \cdot [-ratio^{\frac{9 - num \cdot power - 1}{9}} \cdot (f_c - f_{m1}) + f_c] \quad (12)$$

and remembering (10) the first lower side frequency is

$$f_{m1} = (f_c - f_{m1}) \cdot freq \cdot ratio^{8/9} \quad (13)$$

which is constant with num, i.e. for each element in the event.

3.2 Time Generation

Another important topic is time generation: each instrument has time parameters, as begin time, duration and attack and decay time. For the determination of the first two parameters, the procedure EVENT2 (generating the events) calls the service routine PROPORTION, for two reasons: to calculate a global weight factor (*sc_prop*) and to calculate the temporal weight of each element (*prop*) in respect to the total attack duration of the event. From this last parameter, knowing the attack duration of the event (*at_dur*), i.e. the part of the event in which the elements can begin to play, the begin time of the next element is calculated by the current begin time by the formula

$$nextbeg' = nextbeg + prop \cdot at\_dur \quad (14)$$

The total attack duration of the event is then partitioned between all the (parent) elements in the event. In each event the elements are numbered by a counter, *cnt*: the first instrument that begins to play is the number 1…

The duration (*el_dur*) of each element is determined by considering the remaining time from the beginning of the instrument play to the end of the event, weighted by a factor directly related to the number of the element generated, in accord with the expression

$$el\_dur = (beg + dur - el\_beg) \cdot \left(\frac{cnt}{elements}\right)^{2 \cdot ext} \quad (15)$$

where beg and dur are respectively the begin time and the duration of the event, *elements* is the number of parent elements in the event and *el_beg* is the begin time of the current element. *Ext* represents a weighting factor for an exponential interpolation, and is comprised in the range 0.8<*ext*<1.5 in *Stria*. An important situation happens when there is no overlapping between two successive elements: in this case Chowning imposed the overlapping condition on the elements, making longer the element with no overlap, by the formula

$$el\_dur = (nextbeg - el\_beg) \cdot 1.25 \quad (16)$$

where *nextbeg* represents the begin time of the next instrument, and *el_beg* the current one.

The attack time of each element in the event is determined by an exponential interpolation between two values: the attack time of the first element, and the attack time of the last element in the event. To make this interpolation a parameter *interp* is used, to calculate the position of the element in the event: this parameter is obtained by

$$\operatorname{int}erp = \frac{el\_beg - beg}{at\_dur} \quad (17)$$

which represents the distance of the beginning of the element from the beginning of the event, normalized to *at_dur* (it is comprised between 0 and 1 for all the elements in the event); by this parameter it is easy to calculate the attack time of the current element by exponential interpolation between the initial and final values (INITATT and ENDATT) with the formula

$$attack\_time = el\_dur \cdot INITATT \cdot \left(\frac{ENDATT}{INITATT}\right)^{\operatorname{int}erp} \quad (18)$$

The same for the decay time. These parameters can be determined also depending by the frequency (by an input choice) in way to obtain short attacks for low pitches and long attacks for high pitches; the inverse for the decay time.

3.3 Spatialization

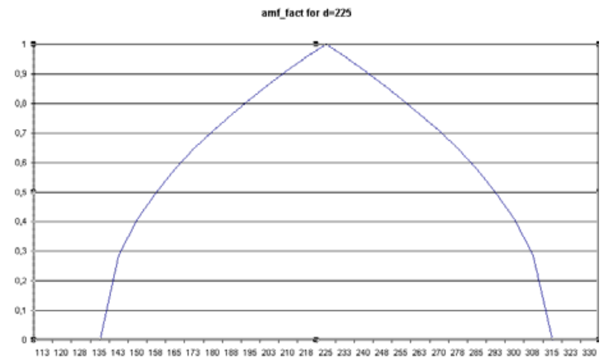
Spatialization is given by three factors: the reverb percent of all the sounds in the event, the apparent angle of the source and the apparent distance of the source of the sound.

The reverb is constant for every parent element in the event, while the apparent angle of the source is given by the quadraphonic diffusion of the sound, by calculating the four amplitude factors, for the four speakers. For the speaker positioned at the angle *d* (equal to 45, 135, 225 or 315) that has to emit a sound apparently diffusing from the angle deg, the amplitude coefficient is calculated by the formulas

$$amp\_fact = \sqrt{\frac{\deg - d + 90}{90}} \quad \text{for } d-90<\deg<d \quad (19)$$

$$amp\_fact = \sqrt{\frac{d + 90 - \deg}{90}} \quad \text{for } d<\deg<d+90 \quad (20)$$

and for the speaker positioned at d=225 degrees, the amplitude diagram is



Each event has many elements, numbered by the variable *cnt*: the reference angle *el_deg* of each parent element is calculated by the formula

$$el\_\deg = ev\_\deg + 360^{\left(\frac{cnt}{elements-1}\right)^{0.2}} \quad (21)$$

where *ev_deg* is the reference angle of the current event: this means that the elements in the event rotate around the listener, more than 360 degrees per event. Another rotation is introduced by event to event: at the end of the generation of an event (a parent or a recursive one), in fact, there is a rotation of –90 degrees: this means that a slow rotation on the events is superimposed to the one due to the elements, in the opposite direction. As we will see in the following section, the child events generally spins around the listener faster than the parent ones, generating dynamism.

The apparent distance from the source is given by varying the ratio between the intensity of the direct sound component and of the reverberated one: the second one is kept fixed in the whole generation of parent elements, while the first one can vary by element to element, in exponential way with the formula

$$dis = (cnt \cdot DIS\_SCALE)^{0.8} \quad (22)$$

where *DIS_SCALE* is the distance scale of the event, and *dis* is the distance parameter of the current element, numbered by *cnt*, obtaining an apparent movement of the source (varying its position exponentially).

3.4 Recursion

In the generation of an event a recursion may appear: after the creation of each element, procedure EVENT2 checks if two conditions are verified:

- The number of recursions done in the event is less than the maximum one imposed by input (usually one recursion per event)
- The value of the weighting factor prop for the current element is a particular one (condition which is verified on average one time every five elements)

If both these conditions are true the current element is the parent of a child event, and procedure EVENT2 calls itself with other parameters, to construct the new event, which has different characteristics from the parent one.

The child event has duration and attack duration shorter than the parent's ones, because these parameters are scaled by *prop*<1 in the recursive calls: this means that the recursive events are shorter than the original ones; in addition, being shorter the attack duration of the child event, its elements will be closer than the original ones, increasing the dynamism (shorter event with closer elements).

The number of child elements is kept minor or equal to 9 (while the number of parent ones can be major), to avoid the possibility of instrument overflow. The base frequency for the new event is given by the expression *fff*freq*, i.e. it is equal to the frequency of the note played by the parent element which generated the recursion; the space variable is chosen by imposing |power|=1, and this means that the recursive event occupies only one pseudo-octave, above or below its base frequency (above if the original value of power was <1, and vice versa). In the case in which the new base frequency is greater than 1618 Hz, power is kept equal to –1, in way to avoid frequency divergence.

The frequency space occupied by the child event is divided in 9 notes (no 18-tones division), in order to avoid too rich spectra. The child event begins in the moment in which the parent one begins, and has attack and decay times calculated by the values by input and independent by the frequency. The reverb percentage for the childs is 1.2 times the parent's one, resulting in a major reverb percent in a minor duration. The rotation around the listener is generally faster than the parent's one, depending on a smaller number of elements in a shorter event.

The generation of the parent elements continues after the creation of the child ones, starting from the moment in the score in which it was interrupted.

At this point it is easy to understand the meaning of recursion in *Stria*: in the generation of an event, one element can create a recursion, which generate another event, shorter, starting from the moment in which the parent element begins.

This new event is shorter and has closer elements, spinning around the listener faster than the original ones, has a frequency space occupation one pseudo-octave wide and a base frequency equal to the note played by the parent element; the reverb percentage is bigger, for a shorter time, to increment the acoustic weight of this event. This, in other words, means that the recursion generates an explosion of the sounds in correspondence of the recursive call, with many close and short sounds rotating fast around the listener, creating a big dynamism.

4. CONCLUSIONS

After this analysis it easy to understand that the most meaningful aspect in Stria is the formalized process which controls both the global and the low level parameters in the composition of the sounds. The recursion is used to add importance, dynamism and speed to the sounds, thus generating an acoustic explosion.

5. REFERENCES

[1] Chowning, J, "The Simulation of Moving Sound Sources," *J. Audio Eng. Soc.*, 19(1), 1-6, 1971.

[2] Chowning, J., "Perceptual Fusion and Auditory Perspective." in P. Cook (ed.), *Music, Cognition, and Computerized Sound: An Introduction to Psychoacoustics.* Cambridge, MA: MIT Press, 1999.

[3] Smith N. W., *SAIL TUTORIAL*, 1976, http://pdp-10.trailing-edge.com/decuslib20-01/01/decus/20-0002/sail.tut.html

[4] Chowning, J., Bristow, D., *FM - Theory & Applications*, Tokyo: Yamaha Music Foundation, 1988

[5] Dodge, C., Jerse, T. A, *Computer music: synthesis, composition and performance*, N.Y: Schirmer, 1985

[6] Snijders C. J, *De gulden Snede*, De Driehoek, 1969

[7] Original algorithms in SAIL, and output files

[8] Personal communication with John Chowning. Special thanks to him for his careful reading this paper.