Reiner Kramer
MUGC 5950
Dr. David Schwarz
10.04.2007

Introduction

The analysis of electro-acoustic music, computer music and interactive computer music poses several difficult issues that represent a frontier in the field of music theory.[1] These issues demand creative solutions so that a larger picture can emerge that will allow a deeper understanding of this type of music. Unlike other bodies of musical literature such as the common practice period, the Renaissance or the Second Viennese School; electro-acoustic music does not have any prescribed analytical procedures.[2] Most analysis is done via methods such as aural transcription, spectral analysis, and musicologically philological analysis, etc. Additionally, it is common practice for composers that work in the field of EA-Music to produce analytical papers and justifications as a supplement to music that is composed. Therefore the composer her/himself becomes the theoretician of her/his own music. The lack of a unified approach to this type of music makes it difficult for music theorists to communicate within an established language on these topics.[3]

Stria is a landmark composition for computer written by John Chowning in 1977 for four channels.[4] When examining *Stria* it becomes clear that Chowning himself produced quite a tome of written materials about his own composition. But because the work holds such an important position within the development of EA-Music, many other writers have also contributed to the analysis of this composition. There have been spectral analyses, transcriptions, philological

[1] Even though electro-acoustic music, computer music and interactive computer music are distinctively different music branches I shall refer to them as EA-Music for sake of simplicity from here on out.
[2] If the existing ways of music analyses are completely commensurate, accurate and conclusive is doubtful in itself.
[3] Not that it is absolutely necessary to speak the same language at all times. Any academic discourse should include the knowledge of as many languages as possible.
[4] *Stria*, being lines like the ones found on glaciers or the human brain.

analyses, re-constructions, re-compositions and source code interpretations.[5] All of these interpretations have been produced because John Chowning did not directly leave a traditional score for analyses by musicologists and music theorists.[6]

Because many EA-Music compositions share this problem, it becomes necessary to re-evaluate what constitutes a score. Is a score only notes transcribed neatly onto a piece of paper? Or can a score be of a set of written instructions on paper? John Cage, with his compositions like 4:33, already expanded our definition of a musical score. EA-Music can and should be analyzed using the tools that have come forth along with the development of the technology of EA-Music. After all, is Schenkerian analysis not an approach that tries to describe music with the help of music itself? Through re-composition of *Stria* in the Max/MSP, Pd, or jMax environment, it will be possible to acquire new analytical knowledge. This analytical model will be capable of being applied to many other electro-acoustic pieces.

In this thesis, John Chowning's *Stria* will be closely examined through the eyes of previous researchers, composers, musicologists, music theorist and John Chowning himself. But it will also try to examine *Stria* through an electro-acoustic music tool like Max/MSP, Pd or jMax.[7] Even though *Stria*, which is a composition for four channels and its sound generation was composed via the SAIL programming language, was not directly composed with any software of the Max family, the fact that the Max family of programs has the ability to re-create most all of EA-music in a unified environment, makes the Max family a perfect candidate to be utilized for

[5] This is especially apparent in the Fall 2007 issue of the Computer Music Journal. Laura Zatta, "The Assembling of Stria by John Chowning: A Philological Investigation," *Computer Music Journal* 31, no. 3 (2007). Matteo Meneghini, "An Analysis of the Compositianal Techniques in John Chowning's Stria," *Computer Music Journal* 31, no. 3 (2007). Kevin Dahan, "Surface Tensions: Dynmics of Stria," *Computer Music Journal* 31, no. 3 (2007). Olivier Baudoin, "A Reconstruction of Stria," *Computer Music Journal* 31, no. 3 (2007).

[6] A recent re-construction of *Stria* has the closest resembly to a score. John Chowning, "Stria: Lines to Its Reconstruction," *Computer Music Journal* 31, no. 3 (2007): 24.

[7] These three programming environments will referred to as the Max family from here on out.

the analysis of *Stria*.[8] Additionally the Max family not only can represent musical example visually but also aurally, which makes it even more powerful for the interpretation of scores, since previously scores could not be sounding by themselves.[9]

Miller Puckette, Kerry Hagan and Arshia Cont developed a set of patches as part of their Pd Repertory Project, which are study patches for students wishing to learn more about the Pd environment.[10] One of these study patches is a recreation of John Chowning's "FM synthesis and non-twelve-tone musical scales, inspired by *Stria*."[11] This thesis will closely examine this study patch, expand it if needed and try to read the patch as if it would represent a score for *Stria* in hope to gain a deeper insight into the piece and create an environment in which a broader range of music theorist, musicologists and composers can talk about *Stria* using the same language in terms of its own vocabulary.[12]

What is a patch?

Before continuing a discussion how a patch can be used as a musical score an explanation of what actually a patch is becomes necessary. This document will frequently refer to a patch as an instructional, visual, sound-design, object-oriented computer program. These patches may be created with software programs as Max/MSP, Pd or jMax, all of which grew out of software developed at IRCAM in the 1980's-90's.[13] A patch is a visual representation of logic and

[8] Stanford Artificial Intelligence Language.
[9] Unless examples were sung, played in reduced form on the piano or perhaps were from a CD or any type of other media carrier. A live performance could also have been beneficial.
[10] http://crca.ucsd.edu/~msp/pdrp/latest/doc/
[11] Ibid.
[12] Since Chowning has re-composed his composition several times himself, there exist three versions of *Stria*, and since other composer recently have re-composed this piece with the help of other programming environments, a re-composition of Stria with the help of a member of the Max family can be completely justified, even if the Max-family was created for interactive computer music. The existence of three main different versions of *Stria*, there do exist more than three versions, are a result of different performances of the piece. An interactive environment is an ideal candidate if the fluidity of the performance of *Stria* needs to be maintained.
[13] Institut de Recherche et Coordination Acoustique Musique.

calculations that can affect or produce sound, MIDI data, or other musically useful inputs and outputs. The author of the patch determines what inputs are used, how they are processed, and what outputs result from the algorithm.

The Max family of computer synthesis programs in itself can be seen as a musical instrument.[14] Like all musical instruments it has the ability to change tone color, timbre, etc. but can do even more than a traditional musical instrument since it can be build in such a fashion that it can provide a model of musical intelligence. On the other hand Max can also act as a tape recorder, a digital reverberator, a synthesizer, or any of a variety of well-known sound appliances.

Another reason why Max patches are important is that they have existed now for about twenty years and are deeply rooted within electronic and computer music itself. With the Max tools it is possible to re-create any aspect of electronic or computer music from the turn of the century up to the present. Early electronic music tools and instruments can be recreated, from simple oscillators to complex synthesizers and signal processors. Therefore the Max family represents a medium that can contain all aspects of electro-acoustic music and can recreate any electro-acoustic experiments and experiences from the past.[15]

The term "patch" has its origin in early music synthesizers since the 1950's and UNIX systems.[16] In early electronic music, it was used to describe a pre-formulated sound setting for a

---

[14] From now on only referred to as Max.

[15] The history of the Max family starts with the Patcher editor written by Miller Puckette. This editor was specifically written for Philippe Manoury's composition Pluton. The main focus of creating this editor was to create a GUI (Graphic User Interface) that would be easy to use for non-programmers, mainly composers. Originally used to control IRCAM's unique 4X synthesizer, by 1989 Max had evolved into a hardware/software system using NeXT computers and ISPW signal processing cards. By 1995, the speed of processors on consumer computers was sufficient that no specialized processing hardware was required; in the succeeding years, Miller Puckette created Pd and David Zicarelli released the commercial software Max/MSP; IRCAM used the newly developed Java environment to create J-Max, the third member of the Max family. More about the history of Max can be read in Miller Puckette, "Max at Seventeen," *Computer Music Journal* 26, no. 4 (2002): 31-43.

[16] Traditional UNIX terminologies can also describe a patch as "a Unix program that updates text files according to instructions contained in a separate file. [...] The patch file (also called a patch for short) is a text file that consists of a list of differences and is produced by running the related diff program with the original and updated file as

musical synthesizer. Modular synthesizers would use patch cables to alter certain sounds by re-routing electrical signals.[17] This is what a Max patch essentially does. It re-routes electrical signals that were converted from some type of digital or analog source to any given subroutines, smaller parts of a patch program that can alter and manipulate these electrical signals and pass these signals back and forth to each other.

Figure 1 shows a simple example of a metronome patch. The patch is written in a way that is self-documenting, i.e. any musician looking at the patch will intrinsically know how to operate it. E.G.: The BPM (Beats Per Minute) slider can be used to change the beats. The VOL(ume) slider can be used to change the volume of the beats. The loudspeaker symbol, when pushed, turns the metronome either on or off. The graph is a visual representation of the electrical sound signal that has been passed from the BPM slider to a "phasor" object[18], which in turn transforms the timing into a clicking sound and then passes it to a "dac~" (Digital Audio Converter) object that enables the listener to hear the sound from the computer. The volume slider also connects to the "dac~" object in order to control the volume of the sound produced, in this case the clicking.

The lines that connect these objects are the patch cables. Patch cables themselves come in two different flavors, one for individual data and another for streams of audio. Certain calculations have to happen in order for the electrical signals / number signals coming from the VOL or BPM sliders to be translated from raw values, predetermined by the slider object, into values that are humanly comprehensible and readable. These values appear in the number boxes below the calculation boxes that are all connected via the different types of patch cables. The

---

arguments. Updating files with a patch is often referred to as applying the patch or simply patching the files." (http://en.wikipedia.org/wiki/Patch_(Unix)) This description goes to the heart of what musical patches are made of and how they are used within the Max family of software programs.

[17] http://en.wikipedia.org/wiki/Patch_%28synthesizer%29.

[18] An object here refers to any type of component that can transform or create an electrical signal. These objects can be created with a Max/MSP patch itself, via JavaScript, but can also be hard coded into the Max/MSP environment via the C/C++ or Java programming languages.

graph gives a visual representation of the frequency and amplitude of the sound created by the phasor. The button with the loudspeaker symbol turns the sound on or off, since Max/MSP is a real-time environment in which the stream of information actually does not stop, which means that the sound is continuously on.

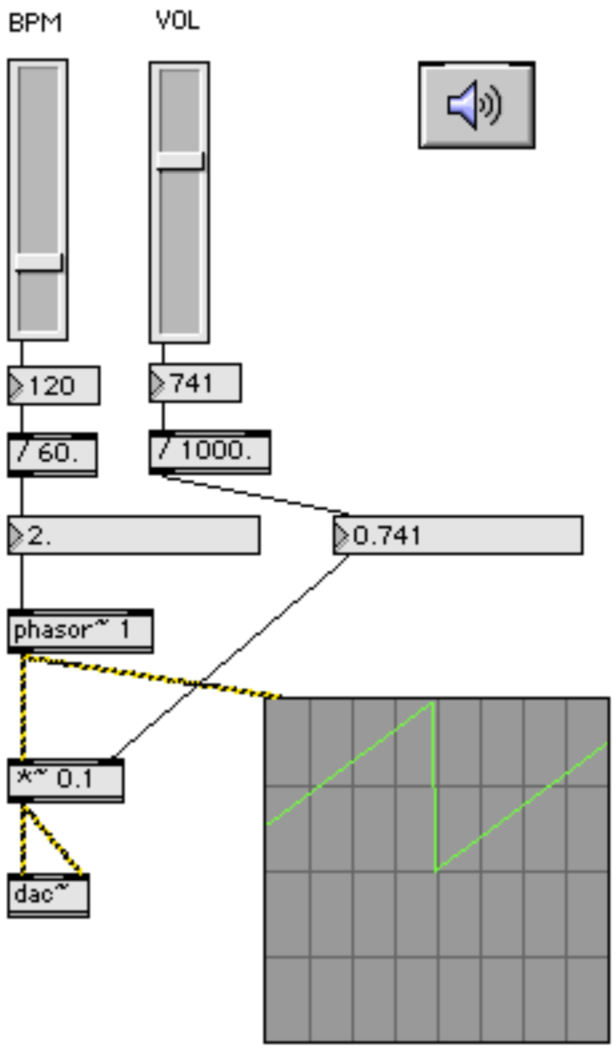


Figure 1: A simple Max/MSP metronome patch. All patches created with the Max family of programs share the same basic structure.

One of the greatest attributes of Max patches is the range of opportunities they provide for computational solutions to creative problems; these are limited only by the imagination of the composer. This is also one of the biggest problems of using Max patches: on the one hand, expert Max programmers may produce systems whose technological interest exceeds their musical sophistication, while novices may produce works using the technological means crudely. In either case, the limitations of the user become flaws in the resulting work. These are issues that yet have to be resolved when composing music with Max patches, but have been known to composer like Stravinsky who states "I shall go even further: my freedom will be so much the greater and more meaningful the more narrowly I limit my field of actions and the more I surround myself with obstacles. Whatever diminishes constraint diminishes strength. The more constraints one imposes, the more one frees one's self of the chains that shackle the spirit."[19]

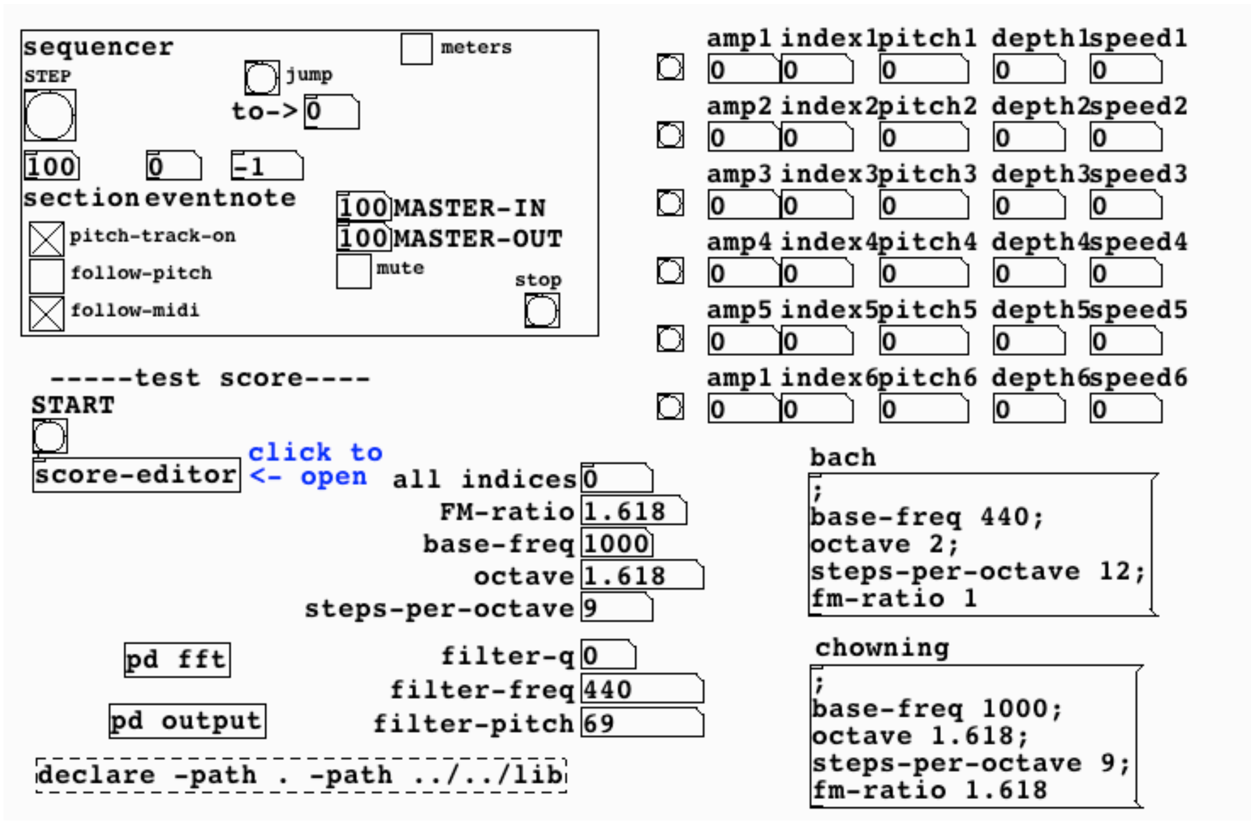
Sample Analysis

To use a re-composed patch as a score and therefore as an analytical tool, allows the theorist to observe multiple layers of information that exist within the music simultaneously.[20] These layers include dynamic and static concepts such as: pitch, timbre, volume, overtone series, division of octave, time, octave ratios, frequency modulation, musical acoustics, psycho-acoustic phenomenon, text files with descriptive information of what will happen and cue lists that describe information of when events will happen. Once the layers of analytical information are extrapolated, the theorist can then interpret and connect the information in a meaningful manner in the same way that a Max/MSP composer/programmer/coder connects objects with patch cables.

John Chowning set out to break new ground by basing his composition *Stria* "on the mathematical properties of the Golden Mean."[21] He changes the octave ratio from 1:2 to 1:ϕ,

[19] Igor Stravinsky, *The Poetics of Music*, trans. Arthur Knodel and Ingolf Dahl (Cambridge: Harvard University Press, 1993), 64-65.
[20] This has been hinted at and proposed by Puckette with the help of Pd's visual tools. Miller Puckette, "Using Pd as a Score Language," in *ICMC* (Götheborg, Sweden: 2002), 187.
[21] Meneghini: 26.

thereby creating and entirely different overtone series altogether.[22] According to Meneghini, Chowning “tried to discover an inharmonic ratio to re-define the concept of the octave.”[23] In nature this is not possible, but with the aid of FM additive synthesis it is possible to simulate this ratio on the computer.[24] These differences in the overtone series can be visually and aurally re-presented in a Pd patch. Figure 2 shows a patch with the ф as being the octave division and Figure 3 shows the “natural” octave division.


sequencer
STEP
jump
to->
meters
section
eventnote
pitch-track-on
follow-pitch
follow-midi
MASTER-IN
MASTER-OUT
mute
stop
amp1 index1 pitch1 depth1 speed1
amp2 index2 pitch2 depth2 speed2
amp3 index3 pitch3 depth3 speed3
amp4 index4 pitch4 depth4 speed4
amp5 index5 pitch5 depth5 speed5
amp1 index6 pitch6 depth6 speed6
-----test score----
START
score-editor
click to
<- open
pd fft
pd output

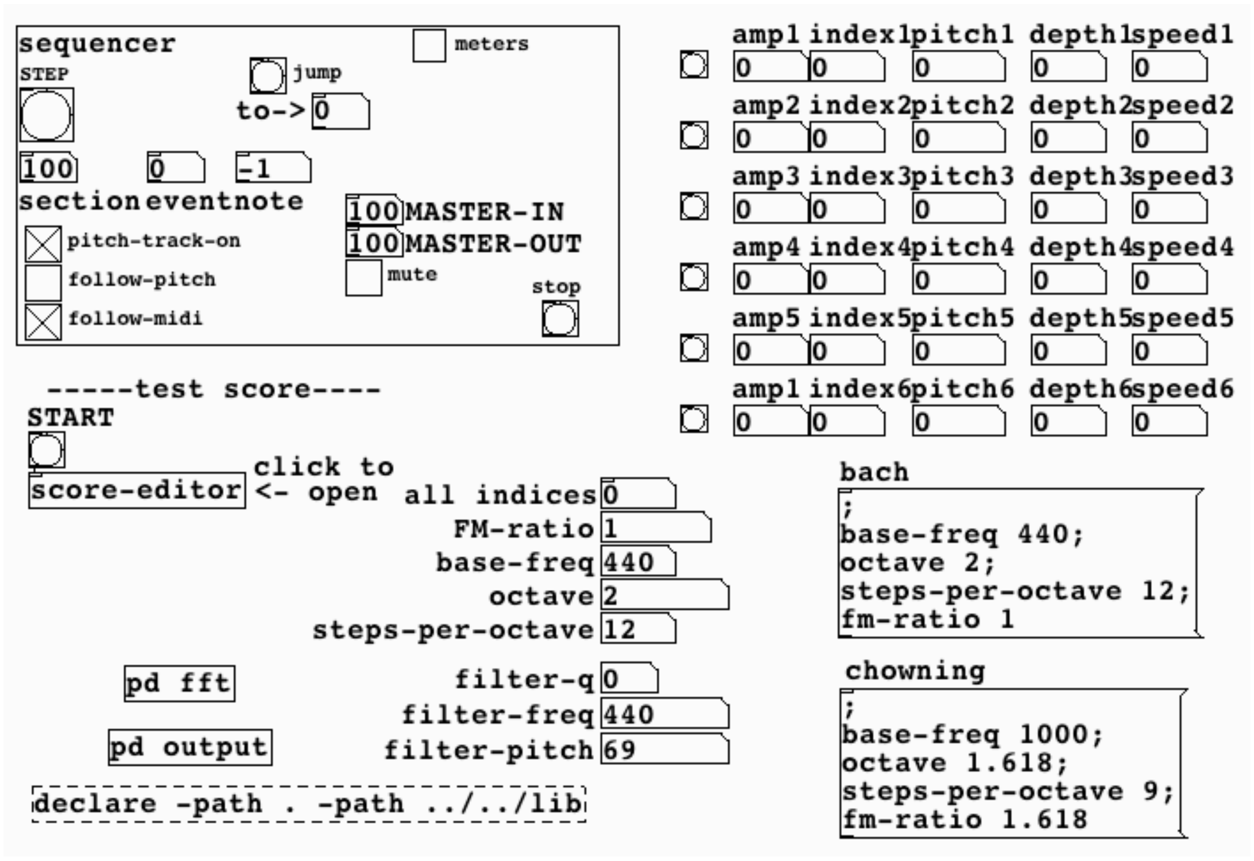
```
all indices 0
FM-ratio 1.618
base-freq 1000
octave 1.618
steps-per-octave 9
filter-q 0
filter-freq 440
filter-pitch 69
declare -path . -path ../../lib
```

bach
```
;
base-freq 440;
octave 2;
steps-per-octave 12;
fm-ratio 1
```

chowning
```
;
base-freq 1000;
octave 1.618;
steps-per-octave 9;
fm-ratio 1.618
```

Figure 2: ф as the octave ratio in Pd.

[22] The paper will refer to the Golden Mean from here on out as “ф”, which is the mathematical symbol associated with 1.618.
[23] Meneghini: 27.
[24] Ibid.



```
;
base-freq 440;
octave 2;
steps-per-octave 12;
fm-ratio 1
```

```
;
base-freq 1000;
octave 1.618;
steps-per-octave 9;
fm-ratio 1.618
```

Figure 3: The "natural" 1:2 octave ratio in Pd.

When observing the number boxes with their corresponding labels of all “indices”, “FM-ratio”, “base-freq(uency)”, “octave”, “steps-per-octave”, “filter-freq(uency)” and “filter-pitch” (in MIDI format) in both figures the differences of the traditional octave division as opposed to the ϕ octave division become immediately clear and can be immediately aurally reproduced, therefore creating the distinction by visual and aural cues. However, in order to be able to hear anything a cue list and a score list, both are text files and were explained earlier, will have to be loaded into the patch, which is done by clicking on the “score-editor” object.

The “score-editor” object then will open two file browser windows with which the two required files are to be loaded into the actual patch. The score has instructions on when the events on the cue list are fired. The cue list is a flat file database like file that contains instructions of what the events are. The cues in the event lists mirror the number boxes labeled “amp(plitude)1”, “index1”, “pitch1”, “depth1”, and “speed1”. One interesting object in the patch is called “pd fft”. This object creates a graph in real-time as the one shown in Dahan’s recent article as Figure 1 that shows the relative amplitude or volume and the associated frequencies from the first event as figure 4 shows.[25]

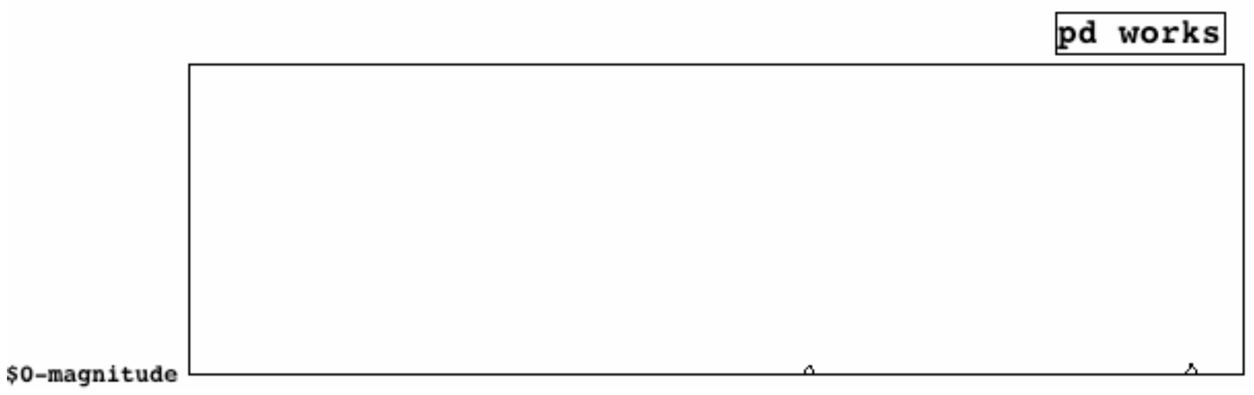
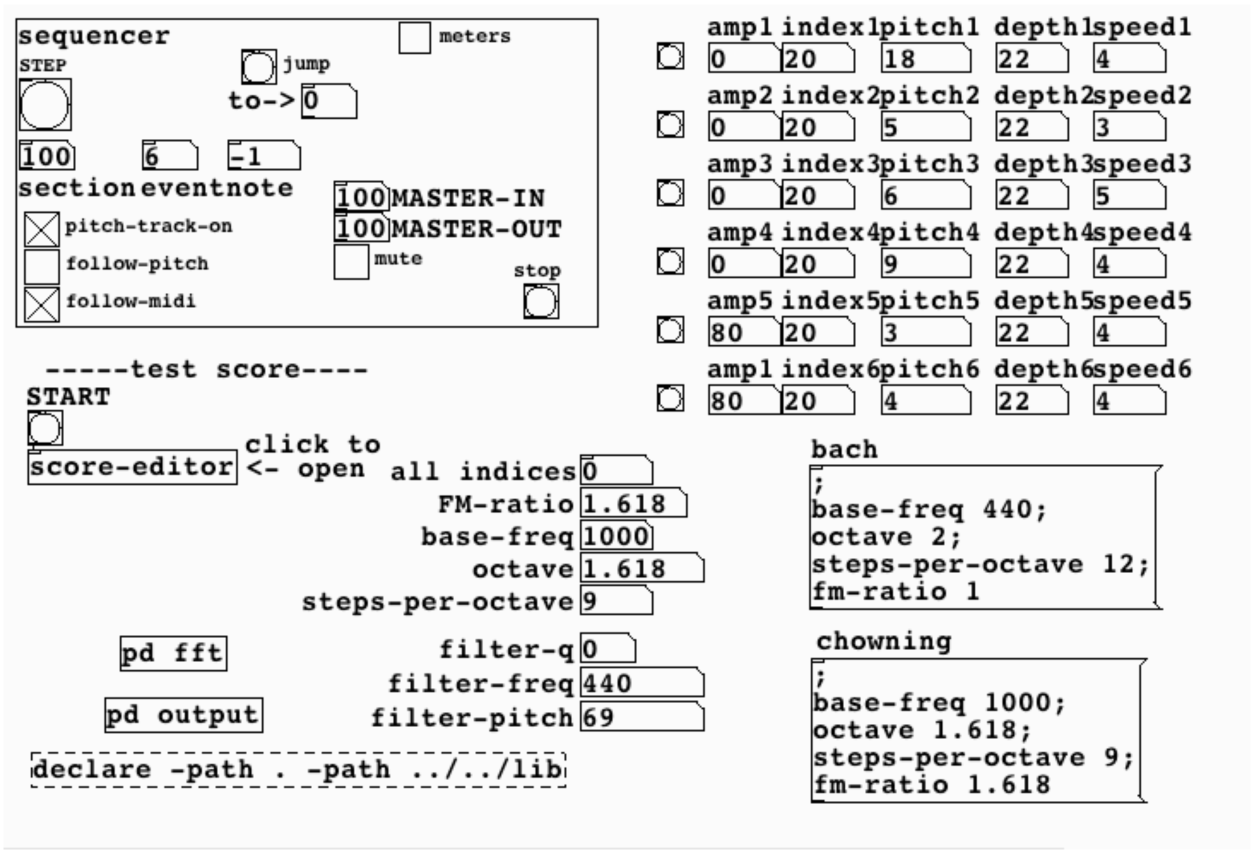


Figure 4: Relative Amplitude and Frequency Analysis produced on the fly by Pd.

The patch has also the advantage that the user, instead of the computer, can actually fire events. Because of this the piece can be slowed down in orders to analyze its aural perception and transitions from one event to the next. Now an analyst can hypothesize about the transitional aural effect of the events of one note transforming into the next and make distinctions by comparing them to the 1:2 octave systems. For example: Since the first event reduces the amplitude of pitch 18 from 80 to 0 at the same time as pitch 5 increases its amplitude from 0 to 80 a downward shift through the overtone series as defined by ϕ can be achieved. This downward motion foreshadows the general downward motion toward the center of the composition. This downward motion also can signify a release of tension that was initially created to begin the composition by only highlighting the upper partials of the overtone series and then shifting these to lower partials of the overtone series. It also becomes clear that a

[25] Dahan: 66.

lowering of the partials creates a pitch that is more stable and possesses a higher density. Figure 5 shows *Stria* in Progress.



sequencer
STEP
jump
to-> 0
meters
100 section
6 event
-1 note
pitch-track-on ☒
follow-pitch ☐
follow-midi ☒
100 MASTER-IN
100 MASTER-OUT
mute
stop

amp1	index1	pitch1	depth1	speed1
0	20	18	22	4

amp2	index2	pitch2	depth2	speed2
0	20	5	22	3

amp3	index3	pitch3	depth3	speed3
0	20	6	22	5

amp4	index4	pitch4	depth4	speed4
0	20	9	22	4

amp5	index5	pitch5	depth5	speed5
80	20	3	22	4

amp1	index6	pitch6	depth6	speed6
80	20	4	22	4

-----test score----
START
score-editor <- click to open

all indices	0
FM-ratio	1.618
base-freq	1000
octave	1.618
steps-per-octave	9
filter-q	0
filter-freq	440
filter-pitch	69

pd fft
pd output

```
declare -path . -path ../../lib
```

bach
```
;
base-freq 440;
octave 2;
steps-per-octave 12;
fm-ratio 1
```

chowning
```
;
base-freq 1000;
octave 1.618;
steps-per-octave 9;
fm-ratio 1.618
```

Figure 5: Stria in Progress within Pd.

This type of analysis will give the music community new insights into the inner workings of *Stria* without having to be technically proficient in high level programming languages. The example patch clearly shows the multi-dimensional capabilities of a patch and the different layers that make up a patch. The advantage to be able to aurally perceive a tiny change and being able to focus on a small change, will give theorists new levels of detailed oriented analysis of EA-Music. The Stria example shows how a patch can be used as a re-composed score and how it's functionality can be expanded to yield additional information.

Timeline

10.04.2007 – Thesis Proposal

12.13.2007 – Thesis Draft

01.17.2007 – Return of Thesis Draft

02.14.2007 – Submit Thesis to Committee

03.06.2007 – Thesis Defense

03.13.2007 – Submit Thesis to Dean

03.28.2007 – Submit Thesis to Toulouse Graduate School

05.02.2007 – Hand in Final Corrections to Thesis to Toulouse Graduate School

Bibliography

Baudoin, Olivier. "A Reconstruction of Stria." *Computer Music Journal* 31, no. 3 (2007): 75-81.

Chowning, John. "Stria: Lines to Its Reconstruction." *Computer Music Journal* 31, no. 3 (2007): 23-25.

Dahan, Kevin. "Surface Tensions: Dynamics of Stria." *Computer Music Journal* 31, no. 3 (2007): 65-74.

Meneghini, Matteo. "An Analysis of the Compositional Techniques in John Chowning's Stria." *Computer Music Journal* 31, no. 3 (2007): 26-37.

Puckette, Miller. "Max at Seventeen." *Computer Music Journal* 26, no. 4 (2002): 31-43.

________. "Using Pd as a Score Language." In *ICMC*, 184-187. Götheborg, Sweden, 2002.

Stravinsky, Igor. *The Poetics of Music*. Translated by Arthur Knodel and Ingolf Dahl. Cambridge: Harvard University Press, 1993.

Zatta, Laura. "The Assembling of Stria by John Chowning: A Philological Investigation." *Computer Music Journal* 31, no. 3 (2007): 38-64.