Larry Polansky
4 New HMSL works
Roulette, 11/22/87

with guests Jody Diamond, Philip Corner, and Daniel Goode

(B'rey'sheet) (In the beginning) (Cantillation Study #1)
for voice and interactive computer; Jody Diamond, voice

Cocks crow, dogs bark, this all men know. But even the wisest cannot tell, why cocks crow, dogs bark, when they do
for computer, computer controlled signal processor, three performers, and stuff; Philip Corner, Daniel Goode, and LP, performers

17 Simple Melodies of the Same Length
for clarinet and computer; Daniel Goode, clarinet

Simple Actions (2) (Distance Music VI)
for computer

About the pieces

The four pieces on this concert are written in the computer music language *HMSL (Hierarchical Music Specification Language)*, written and developed by myself, Phil Burk, and David Rosenboom at the Center for Contemporary Music, Mills College. HMSL is a language and programming environment for experimentation in musical artificial intelligence, composition, and real-time performance. It offers the user (programmer) a large set of high-level musical and experimental tools. It can be used in a real-time performance context, or in "non"-real-time compositional contexts. The language itself is the product of about 6 years work on the part of the authors, and is still evolving, improving, and expanding. It has benefitted tremendously from the work of many other composers and experimenters who have contributed to its development in various ways. HMSL is currently being distributed on the Apple Macintosh (Plus, SE, and II) and the Commodore Amiga computers, and is equivalent on the two machines.

The pieces on tonight's concert are all written since June 1987 (although *B'rey'sheet* existed earlier to that on the HMSL prototype machine at the CCM). Each explores, and attempts to extend the possibilities of the HMSL environment (in particular, the real-time environment) in different ways. What the four pieces have in common, though, is that they attempt to emphasize my current interest in the evolution of composition, perception and performance thru intelligent software, written by composers. Each piece constitutes a kind of experiment in a different aspect of real-time intelligent software design.

(B'rey'sheet) (In the beginning...)(Cantillation Study #1)
(for voice and interactive live computer) (for Jody Diamond)

B'rey'sheet is the first of a set of pieces called the *Cantillation Studies*, which are based on computer aided morphological transformations of the 11th and 12th century Masoretic cantillation melodies for the singing of Torah (Shabat morning tropes). The other two works in this set so far are: *E'leh Tol'd'ot (These are the generations) (Cantillation Study #3)* (for William Winant) for four marimbas and interactive computer; and *V'leem'shol (And to rule...) (Cantillation Study #2)* (for Ann LaBerge and David Rosenboom) for five flutes. Each of these pieces is based on successive 17-verse sections of the Torah. The tropes are used as a basis for computer melodic transformation.

In *B'rey'sheet,* the tropes are sung, unadorned. The computer "listens" to the melodies and generates it's own events based on what it hears, and on where it is in the piece. There is a predefined trajectory of "computer attention", which specifies that at the beginning of the work the computer will be pretty much ignoring the voice, but will gradually and continuously increase its attention until the end, when it tries to follow the voice closely. The computer deals with several musical parameters in its transformations: pitch, duration, intonation and many types of timbral and spectral deformation of four sine waves. Many of the spectral deformations used here are HMSL routines which are usually used for melodic transformation (retrograde, inversion, localized and ranged randomization, etc.), but here they are applied directly to the waveform itself in real-time, at audio rates. Once again, the degree and frequency of deformation of the sine waves that the computer will produce is dependent on which of the piece's 17 verses (or sections) it is currently in.

One interesting aspect of the computer algorithms is that an intonational trajectory is also travelled by the machine, beginning in a 17- limit tuning space, and ending in a kind of simple 3-limit one. The tuning algorithms used here fall into the category of what I have called "paratactical tuning" (see my article of a few issues back in the *Computer Music Journal*)— all of the tuning is done in real-time, in response to the input, and no concept of scale, or gamut, is ever invoked. The pitches used are not chosen, but generated by the machine in real-time, similar to the way a chorous or string quartet would dynamically tune to itself over the course of a piece.

B'rey'sheet primarily uses Amiga local sound, with the exception of a single sine wave produced by the Yamaha FBO1 which follows the voice throughout. The sine wave and the voice are tuned to a simple septimal just scale (8/7's for major seconds, 12/7's for major sixths, etc.). This version of the piece was premiered at the International Computer Music Conference, Urbana, Illinois, August, 1987.

Cocks crow, dogs bark, this all men know. But even the wisest cannot tell, why cocks crow, dogs bark, when they do (for computer, computer controlled signal processor, performers, and stuff)

Cocks crow... is a live improvisation for three performers and computer. The computer generates, in real-time, simple instructions for the performers to follow. It is seventeen sections long, each of whose length is determined to some extent by the machine, and to some extent by one of the performer's input to the machine while the piece is being played. At the beginning of each section, the performers are asked to do different things, and the machine itself will perform in a different way.

Cocks Crow ... makes significant use of the ability to define custom *instruments* in HMSL. That is, raw compositional data in the language is never interpreted until the user specifies its "destination", and the way that data is interpreted is entirely up to the user. In this piece, four such instruments are defined. The first, called a *Chorder,* is an instrument which reads "melodies" (called *shapes* in HMSL) of five dimensions: duration, fundamental pitch of the chord, average loudness of the chord, harmonic complexity of the chord, and on/off time ratio for the duration of the chord. Harmonic complexity is here defined as the height of a pitch in the harmonic series of a fundamental.

The second instrument, and most important for this piece, is defined for the MIDI controlled signal processor used, the Roland DEP-5. A system exclusive library has been written by for this device in HMSL which allows the user to control all of the DEP-5's parameters in real-time, without using such MIDI notions as "preset" (or "program"), "note-on", etc. One can modulate, for example, the reverb time by the computer at very rapid rates, without changing the other device parameters. The DEP-5 instrument in this piece responds to "melodic" information for reverb time, pre-delay time, high frequency damping, filter Q and center frequency, and chorousing rate, depth, and feedback. As is the case with the Chorder instrument, one of the performers can graphically edit the shape of change for the complex "melodies" of this instrument in real-time during the piece.

The third and fourth instruments used here, though used very sparsely, are similar system exclusive instruments for the FBO1. These will be described in more detail below, in *Simple Actions.* As in that piece, no MIDI "note-on" is ever sent to the device, and it is used rather like a computer controlled analog synthesizer.

The entire piece is defined as an HMSL *structure*, a class of musical objects in the language which "knows" how to play its component parts, or more accurately, has an internal intelligence for deciding when and how to play its component parts. This intelligence is user defined, and can be of great complexity, as can the components of the structure itself (for example, it could contain other structures with equally complex intelligences). In this piece, the structure decides what instructions to give to the performers, how long to make the sections, what to do itself in each section, and other things pertaining to the instruments and sounds themselves.

This piece was premiered, in much simpler form, at the Center for Contemporary Music, Mills College, in June, 1987. The performers were myself, Amy Neuburg, and Jarrad Powell.

17 Simple Melodies of the Same Length (for Dan Goode) (for clarinet and computer)

17 Simple Melodies ... is more or less "classic" artificial intelligence. Data is gathered by a kind of perceptron(seventeen melodies of seventeen notes each), the data is sorted (the intelligence), and finally, the data is re-played as a kind of simulation. The form of the piece very clearly follows these three sections. First, the clarinetist plays a 2 1/2 octave chromatic scale, to tune the computer. Then the musician improvises seventeen melodies, and the computer "captures" them. In the second section, the computer "sorts" these melodies into three different lists, each containing all the melodies. In the third section, the machine plays back those three lists.

My main interest in this work is in the second section, or the sorting procedure. For several years I have been working on a theory of what I call *morphological metrics* — ways in which distances between arbitrary morphologies (shapes, melodies) can be computed in ways that are perceptually meaningful (and cognitively interesting and evolutionary). This theory has been described elsewhere in some detail (see the ICMC Proceedings, 1987), but this piece is a very direct application, or experiment, in using these formulas and techniques in real-time. The metric function used here is something I call the Ordered Combinatorial Direction metric. It is fairly complex, but it might be simply described as a distance function between two melodies which is solely interested in the "up/down/sameness" of them, paying careful attention to the order in which this directionality occurs. I believe that this metric gives a good indication of what is sometimes called the "profile" of a morphology. It ignores things like "magnitude" of intervals in the melody, register (or tempo in the duration dimension), etc.

In the second section of the work this metric (OCD) is applied to the seventeen melodies to sort them in three ways. All three of the resultant lists represent "distances from some source melody", in this case the first melody that the performer plays. In other words, the first entry in each list is the melody "furthest" from the source (by some definable measure of "far" and "near"), and the last entry in each list is the source, or first melody itself (whose distance from itself is zero). For the first list, only the pitches of the melodies are used for the metric, in the second, only the durations, and in the third, equal weight is given to the pitches and the durations.

The first section captures the melodies by use of a Gentle Electric Pitch to Voltage Converter (an analog device which detects pitch information) and a home-brew analog to digital converter built for the Amiga. This information is interpreted by a class of intelligent objects in HMSL called *recorders*, which are more usually used in the system for conventional MIDI multi-track recording. The second section, which calculates all of the distances between shapes and sorts them according to these distances in the three lists, takes about 3 seconds. It makes use of a high-speed Batcher Sort written by Phil Burk, and part of HMSL on the Amiga.

The third section, where the computer actually makes sound, is extremely simple. The three sorted lists are played back polyphonically by the machine, using Amiga samples made by myself and Phil Burk. These samples are not in any way "exotic", and are actually distributed freely with HMSL for any user who wants them. They include various mandocello, mandolin and dobro sounds, various voice sounds, some metal bowls, and some simple analog electronic sounds.

This is the first time this piece has been performed.

Simple Actions (2) (Distance Music VI) (for Daniel Kelley) for computer

Simple Actions (2) is a revision of *Simple Actions,* an earlier work (March, 1987) done entirely for Amiga local sound. This new version integrates a new system exclusive library for the FBO1, and the ability to build one's own custom graphics screens in HMSL, and interact with them live. The piece itself is one of a set of works for performer/programmers called *Distance Musics*, recently published in *Perspectives of New Music.* This is a set of pieces for live interactive computer systems, in which each of the performers must write their own code to perform the specific pieces.

Simple Actions is based to some extent on Minsky's "society of mind" concept — that complex intelligences are often created by the difficult to predict interaction of many simple intelligences, which share a certain informational ecology. Other ways to describe this notion might be as groups of cellular automata, or in a biological sense, as a large Petri dish of highly interdependent small organisms. The score for the piece defines a "simple action" as some musical process which can be described in less than one sentence or so. These actions have the further description that they must, as much as possible, share data with each other, so that when they alter that data through their own behavior, that data is altered for all other actions which might use it. In other words, all of the actions tend to alter the ecology of the system in some way, and the behavior of each is affected by the behavior of all.

The performance of this piece is rather unusual, and difficult. The performer can bring to life, or put to sleep, any of the actions, and in some cases, alter the range of behavior of the actions. More than that is difficult to control, since the system behavior often takes on a kind of life of its own. The actions themselves are extremely simple: beeps, glissandi, timbral changes, simple melodies, chords, simple modulations, etc. However, the number of processes possible at any time is very large — in this version seven FBO1 voices and four Amiga voices can do any or all of these things at any given time. In HMSL, these processes are simply set up as *jobs* — processes which execute a user-defined function at a certain time interval (the time interval can of course be changed in real-time by the job itself as part of its function!).

As in *Cocks crow...*, the notion of a MIDI "note-on" event is avoided entirely. Instead, the FBO1 is controlled entirely thru system exclusive code. This allows me to not only specify actual frequencies for the instrument (it has approximately a 1.6 cent resolution, and unlike the TX81Z and DX-7 II allows for individual pitch by pitch tuning as opposed to a "keyboard tuning definition" for the whole instrument), but to also change all of the voice parameters rapidly in real time. Once again, the notion of "preset" becomes meaningless, all parameters of any preset can be (and are) modulated by the computer as fast as one wishes, and as such, the timbral possibilities of this inexpensive device become much more expansive. In *Simple Actions,* many of the little "critters" are designated to control real-time timbral parameters of the FBO1 (like LFO rate, shape and depth, algorithm, etc.). The DEP-5 and the Amiga local sound can also be contolled from the screen in similar ways.

Simple Actions (2) is, for me, a deliberate attempt at musical and cognitive co-evolution of myself and the machine. Like *17 Simple Melodies...*, it is, admittedly, a small beginning at a new form of musical and performance intelligence. But it is, I think, a beginning, and not an end. The nature of the piece obviates all attempts on my part at traditional notions of creativity, cleverness, drama, and "musicality", and forces me to rethink, in its performance, the nature of the composer and of compositional, creative and perceptual intelligence.

Thanks

To Mission Graphics Support, (433 E. 6th St.), a digital art gallery and Amiga dealership for the loan of computer monitors; Brooks Williams and Henry Lowengard; Phil Burk; Dan Goode; Jody Diamond; and Phil Corner.