The RoboSax Project (1991-2001): forms of performer/machine interaction in works by Jonathan Mustard and Lindsay Vickery

Lindsay R. Vickery

Music Department, Western Australian Academy of Performing Arts,
Edith Cowan University
email: l.vickery@cowan.edu.au

Abstract

Since 1990 Jonathan Mustard and I have been creating a body of work for solo wind instrument (acoustic or electronic) and interactive electronics. This paper gives an overview of these works and their exploration of interaction and control issues between the performer and machine. It discusses the evolution of the continuing project through the use of software and hardware from the Yamaha TX81z, through MAX to MAX/MSP.

1 Introduction

The 'works for solo instrument and interactive electronics' genre probably begins with Gordon Mumma's work for horn and electronics, *Hornpipe* (1967) which used 'microphones for "listening" to sounds made by the horn as well as for analyzing the acoustical resonance of the space' (Winkler 1998 p.12). It was an ambitious start for a genre that has had to overcome many challenges in order to develop a diverse, expressive and meaningful repertoire.

The attractions of the genre for composers are extremely varied, but some principal motives include: the desire to extend or transform traditional instruments; the ability to create textural and compositional complexity from a single live source; the non-linear possibilities of interactive work; and the possibility of creating a 'custom' improvising partner

A number of different systems have sprung up to deal generically with the types of interactivity composers may wish to employ. The best known of these began to be commercially available in the early nineties: Interactor (Subotnik/Buchla/Coniglio), Hyperinstruments (Machover), Cypher (Rowe), Kyma (Scaletti) and Max (Puckett/Zicarelli).

The RoboSax Project began in 1991, and has continued over the last ten years as a forum for composer Jonathan Mustard and myself to explore the possibilities available for solo wind player and electronics. The work has in general been sustained through personal resources and has necessarily developed a pragmatic approach to materials and technology. This has resulted in a growing concentration on software rather than hardware solutions to musical problems.

The RoboSax Series of works encompass a variety of combinations. It is interesting to note the increasing tendency toward extremely straightforward set-ups:

The RoboSax Project:

1. Mustard: *RoboSax I* (1991) Yamaha WX Series Windcontroller, 2 Yamaha TX81z Tone Generators and Sequencer

2. Vickery: *DiceGame* (1991) Clarinet, Microphone and Ensoniq DP/4 Effects Unit

3. Vickery: *27Matrix* (1995) Yamaha WX Series Windcontroller, Yamaha TG500 Tone Generator and Macintosh [MAX]

4. Mustard: *Robosax III* † (1996) Soprano Saxophone, MIDI Footcontroller, General MIDI Synthesizer and Macintosh [MAX]

5. Mustard: *RoboSax IV 'The Arsonist'* (2000) Alto Saxophone, Microphone and Power Macintosh [MAX/MSP]

6. Vickery: *<as viewed from above>* (2000/01) Soprano Saxophone, Microphone and Power Macintosh [MAX/MSP]

† RoboSax II (1992) Yamaha WX Series Windcontroller, 2 Yamaha TX81z Tone Generators and Sequencer is currently withdrawn

In common to the six pieces is a desire to create sonic environments of textural and compositional complexity that can be generated in real-time through the actions of a single performer. The instrumentation of the successive works shows a continuous progression through available technologies from programmable tone generators controlled via MIDI to digital audio controlled by analogue audio.

It is possible to detect differing principal preoccupations underlying the works. In a broad sense these preoccupations embody three contrasting paradigms of musical organization Dramatic; Structural; and Symbolic. Mustard's contributions, under the generic designation *RoboSax*, explore elements of interaction and control between the performer and machine. Their effectiveness is due principally to intrinsic dramatic tensions set up by the composer between the live performer and the interactive machine components.

My works *DiceGame* and *27Matrix* both focus on 'real-time' composition possibilities, in particular the generation of algorithmic structures via improvisation. Their success resides in their ability to mediate between fixed 'pre-compositional' processes and free improvised performance. My most recent work *<as viewed from above>* in contrast seeks to establish a new structural model based on formalizing certain characteristics of psychological obsession in a non-linear interactive format. Its relies on the symbolic quality of music to mirror certain patterns of thought in a way that is, if not consciously recognizable, at least resonant with its audience.

2 Mustard's RoboSax works

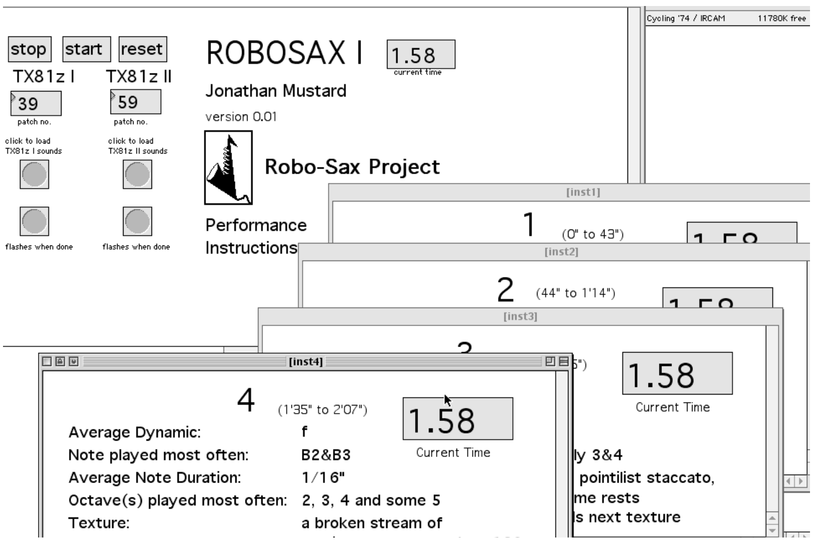

Figure 1:Jonathan Mustard's *RoboSax I* (Max Version) (Mustard and Vickery 1996)

In his programme notes for *RoboSax IV* Mustard states his credo for the series as a whole:

They are the protagonists in the drama or perhaps the dilemma of 20th / 21st Century existence where the interaction of humans and machines (computers in particular) is ubiquitous and the question of which element in the equation has the control at any one time becomes ambiguous. (Mustard 2000)

His choice of title is typical of his oeuvre as a whole, reflecting in equal measure a strong engagement with social issues and self-deprecating humour. "RoboSax" alludes to Paul Verhoven's 1987 film RoboCop and the spin-off series that followed it. Mustard both references and satirizes his own interest in the film's central theme - Human's increasing integration with machinery. At the same time he draws attention to the parallels between composers' and film makers' tendencies to create series of related works under the same title, with its undercurrent of cynical exploitation of success and implication of mechanical mass-production.

2.1 RoboSax I

Mustard's first work in the series places the computer firmly in control. The piece took up the opportunity of exploring the possibilities of the Yamaha WX series of Windcontrollers (WX7 [1988] and WX11 [1989]) that had recently emerged on the market. The performer plays a WX11 MIDI windcontroller while a computer changes patches on a synthesizer that can between them generate up to 32 notes for each note triggered by the performer. In the original version the WX performer with a score and a stopwatch attempted to carry out 'meta-compositional' instructions within particular timeframes.

The catch in carrying out these instructions is that the stream of MIDI information generated by the WX instrument is realized through the filter of a sequencer's continuous alteration of synthesizer patches - over which the performer has no control. Mustard's ingeniously programmed patches on the TX81z add to the computer's ascendancy, confounding the performer's expectations by reordering and/or harmonizing the pitch of each key or changing sounds with each new re-articulation.

The restraint of dividing control of various musical parameters between performers and machine was being independently explored in the same year by David Jaffe and Andrew Schloss in their work *Wildlife* (1991) (Rowe 1993 p.85). Although more complex in its capabitities, *Wildlife* required a NeXtstation, a Macintosh IIci, a RadioDrum(baton), Zeta Violin, a Yamaha TG77 and two sound cards.

The original version of *RoboSax I* called for an Atari computer to send patch changes to the tone generators. In 1996 I created a MAX version of the piece combining the score, stopwatch and Atari's functions together. In this version the performer is presented with the instructions at the appropriate time via MAX patcher windows that open synchronously with the sequence.

But *RoboSax I* 's subtitle (The Strathfield Massacre) reflects a more profound level in which the work attempts to form some kind of response to the arbitrariness and horror of this infamous incident in which a gunman went on the rampage in one of Sydney's outer Western suburbs killing 13 people including himself and injuring many others.

At the core of the work's formal structure then is the knot of our incomprehension of such events: the performer placed in the position of a powerless observer who is required to play from a list of verbal instructions without really knowing what is going to be the end result.

2.2 RoboSax III

In his *Robosax III* Mustard turned this model on its head, placing the performer (this time on an acoustic instrument) directly in control of most of the musical parameters via MIDI Foot-pedal Controller. In *RoboSax I* the performer is continuously required to make performance decisions based on (often incorrect) expectations of what the aural result might be. In contrast *RoboSax III* puts the performer in the taxing position of controlling not just their own instrument, but up to seven other virtual instruments via the foot-pedal simultaneously.

The Yamaha MFC-10 foot-pedal used for this piece is capable of sending 128 program changes (via a bank of ten buttons at a time), as well as continuous controller information from up to 5 "volume" style (CV) pedals. Program changes sent from the MIDI foot-pedal control harmonic, timbral, tempo and several other MIDI controller parameters.

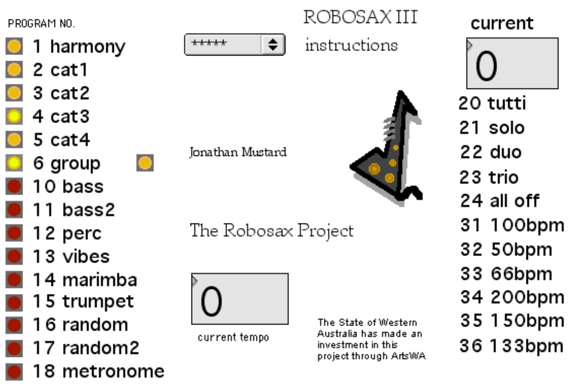

Figure 2. Mustard's RoboSax III

The first six program changes send changes to the harmonic and textural composition of the piece; program changes 10 through 17 make timbral changes (i.e. choose different instrumentations); 18 turns on and off the `metro` object - effectively the engine that generates MIDI events in RoboSax III; 20-26 create various instrumental (timbral) combinations; and 31-36 control the tempo of the metronome and therefore of the whole piece. Additional parameters not visible on this front panel allow continuous tempo, pitch-bend and panning control via additional foot-pedals.

A problem central to all electronic music is the issue of obsolete/superseded equipment rendering works impossible or difficult to reproduce in the future. *RoboSax III* attempts to partially redress this situation by standardizing the equipment required to perform the work: a MIDI foot-pedal, a Macintosh computer capable of running MAXplay and a general MIDI synthesizer. Other versions of the work are of course possible using any MIDI compatible tone generator or sampler, however Mustard's General MIDI format establishes a minimum technical requirement that is readily realizable.

Compositionally *RoboSax III* attempts to create something akin to a virtual 'jazz combo' to accompany the soloist. This is reflected in the choice of typical jazz instruments as the principal timbres (ie Muted Trumpet, Vibraphone, Double Bass, Kit...) and also in more general characteristics such as the texture and syncopated dynamics.

RoboSax III belongs to the 'improvising partner' paradigm - creating a virtual combo under control of the soloist. Mustard's aims mirror those of improviser Bruno Spoerri. *'The important thing for me was to have partner who threw balls at me, who gave me a reason to react in a certain way, but would react in a logical way. ' (Bruno Spoerri in Chadabe 1997 p.322)*

Below the 'pseudo-combo' surface however *RoboSax III* employs many techniques typical of Mustard's notated linear works like *Nine Realizations of a Brass Guerilla* (1988) or the *Automaton* Series (1989-90). In the MAX software a central `metro` object sends 'clock' information in the form of a continuous stream of pulses. Mustard utilizes a form of restricted random choice to disregard certain pulses in order creating gaps in the stream and therefore rhythmic variety. Additionally, in some cases rather than deleting the pulses, he reroutes them to different instruments creating an interplay of interlocking rhythms (hoketus) typical of his music. Pitch aggregates used to create melodic and harmonic material are chosen from tables along similar principles. One of the most interesting features of *RoboSax III* is two 'Random' buttons (Program change 16 and 17), both capable of sending all of the patch's variable parameters into overdrive – rhythms, voices and instruments proliferate exponentially in a disturbing crescendo of activity.

2.3 RoboSax IV

In *RoboSax IV 'The Arsonist'* Mustard establishes a middle ground in the struggle between performer and computer control. Again the soloist performs on an acoustic instrument, the Alto Saxophone, but this time reading a relatively conventional notated score. The computer (running MAX/MSP) also performs its samples according to a linear script. What differentiates this work from the traditional 'soloist plus tape' format is that Mustard sets up a feedback loop whereby the live performance and the computer performance modulate one another. The audio via microphone from the live performance modulates both the characteristics of the computer's samples and its own input via filtering to create a third stream. The resultant modulations of the computer's audio in turn influence the live performance.

The same motif of modulation occurs in the score for *RoboSax IV* where Mustard has the soloist modulating their own performance via a complex system of cross-fingerings that create pitch and timbral variation. Elements of chance resulting from the use of some graphic notation, relatively free timing in the score and use of random number generators in the script enliven the work, elevating it into a region somewhere between freedom and control.

3 Vickery's RoboSax works

3.1 DiceGame

My initial interest in this field began with the concept of live processing of an analogue instrument. *DiceGame*, like Mustard's *RoboSax IV,* is a (mostly) traditionally notated work for clarinet with an added electronic component, in this case the Ensoniq DP/4 Effects Processor. *DiceGame* belongs to a series of my works dating from the early 90s, each using a short series of numbers (cypher) as a means for generating all musical parameters from the micro- to macro-scale. The work's fractal nature is augmented, and to some extent its definition is enhanced, by the addition of electronic processing.

The interactivity in *DiceGame* is limited by the available interfaces between the machine and the performer. By far the most straightforward control is achieved through the use of a foot-pedal to step through processor settings, and CV pedal to make continuous changes in certain parameters. Despite the fact that in 1995 interactive signal processing was still in relatively early stages, some greater level of interactivity is achieved through simple use of signal amplitude measurement. The most obvious example is the use of Gate processors to limit, depending on their volume, what signals are passed on to other effect patches. This approach is somewhat analogous to that used by Todd Winkler in his work *Snake Charmer* (1991) where the modulation speed of a chorus effect is controlled by the changing dynamic level of the clarinet soloist.(Winkler 1998 p.249)

The DP/4 part consists of 14 separate patches that are ordered into a 24 patch sequence. The overall form of the work is of nine main sections, consisting of five contrasting types of material. The sections are in the order 532214451: and are relatively easy to hear as they have gradually diminishing pitch resources: section 5 uses a 15-pitch group, 4 uses 12, 3 - 9, 2 - 3 and Section 1 uses 1 pitch. During the repeated sections (there are adjacent 2 and 4 sections) the electronic processes help to define the structure through processing changes. For example the second '2' section is marked by a change in harmonization.

The processes involved are used to reinforce the work's structure - emphasizing the changes between sections and, to some degree, the sub-sections within them. The effects can essentially be broken down into combinations of pitch and time distortions of the live signal.

Some examples will suffice to illustrate:

• *Velocity Octaver* is the principal effect during the opening section '5'. It processes signals above a certain volume (db) adding four delayed semi-quavers each harmonized at the octave below the original note in tempo MM.= 90. Signals below that volume pass through with no effect. The aural result is that accented notes are doubled with a lower octave, which is repeated, as four delayed semi-quavers.

• *Rainbody* delays and pitch-shifts notes downwards by small amounts (in inharmonic intervals) with the aural result being a proliferation of asynchronous descending micro-tonal lines similar to Ligeti Micro-Polyphony.

• The *Tempo Digital Delay* delays the audio signal to four different degrees. Each is then panned hard right, hard left, middle right and middle left in the stereo spectrum. The length of the delay is determined by the position of the control voltage (volume) pedal notated in the part. The aural result is a five-part canon (including the live signal) panned across the stereo spectrum.

• '*Ascending Whole-tones'* delays notes by a semi-quaver (within the tempo) with four repeats. Each delayed note is a whole-tone higher than the last, creating a four note ascending whole-tone scale.

3.2 27matrix

My second work for solo wind and electronics was *27matrix*. It is an interactive improvisation environment written in MAX. An improvising soloist (again a WX series Windcontroller) provides the raw MIDI data that is transformed by MAX in real time into various structures.

27matrix combined live improvisation with the sort of formal processes that I had been using in my music between 1990 and 1995. These structures are rigorously governed by a principle of self-similarity - all material created by the computer draws on a nine-digit cypher (the same one used in *DiceGame*) as its generating kernel.

This type of formal structure originating in the works of Webern has found particular resonance in the field of electronic music where simultaneous control is applicable to an ever-increasing range of parameters – even to the level of the shapes of soundwaves themselves (Yadegari 1991). In interactive computer music one of the best early examples is Gary Lee Nelson's *Fractal Mountains* (1988-89) which maps the performer's notes via MIDI into a series of graphs that control a wide variety of musical parameters in real-time (Nelson 1994). In *27matrix* the fractal algorithm is predetermined, but give rise to a diverse range of musical textures.

Like *DiceGame*, *27matrix* comprises five basic environments that employ a number of different strategies to transform the live MIDI input from the Windcontroller. Each reiterates the cypher structure in the pitch and/or rhythm domain. They are discussed in the order in which they are heard.

Environment Five: This environment is an inversion of typical jazz improvisatory practice: pitches from the performer's improvisation are fed into the accompaniment and successive notes are gradually spread out over five octaves. The rhythmic structure of the accompaniment itself is constructed of canonic versions of the cypher rhythm.

Environment Three: The performer's incoming notes are repeated between two and five times depending on their velocity: the higher the velocity the more repetitions. The frequency of the repetitions is dependant on the octave in which they are played: the higher the register the faster the repetitions.

Environment Two: The first note played in this patch is captured and played in the original and retrograde version of the cypher rhythm. The performer’s improvisation is harmonized by between one and nine pitches (dependant on the note’s velocity). Every 27th note the soloist plays triggers a rapidly repeated note to travel across the stereo field.

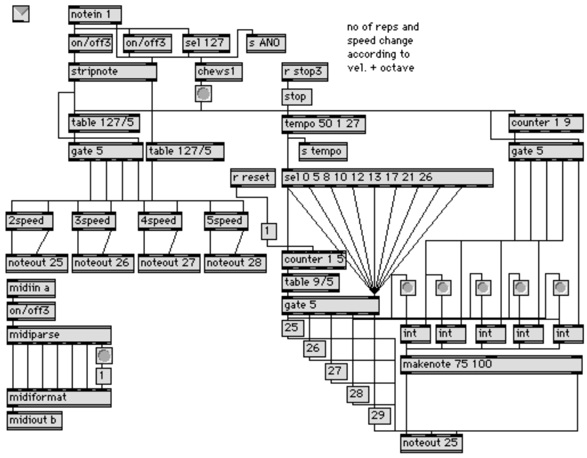

Figure 4. *27matrix* environment 3

Environment One: The first note the soloist plays is captured harmonized and repeated in the cypher rhythm. On the cypher's completion another rapidly repeated chord travels across the stereo spectrum.

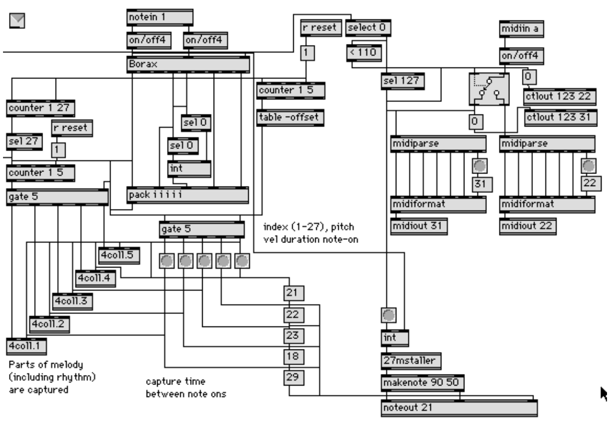

Figure 5. *27matrix* environment 4

Environment Four: The soloist's improvisations are captured, and played back on different instruments, in different registers and at different speeds surrounding the performer with different versions of their material. The solo instrument is a flute-like sound if above a certain velocity and a gamelan type sound if below. Changes of instrument create 'stuck notes' that can only be turned off by playing a full velocity note. A full velocity note also triggers the cypher rhythm to be played.

3.3 <as viewed from above>

<as viewed from above> is written in MAX/MSP and like Mustard's *RoboSax IV* 'listens' to the live performance of an acoustic instrument as a trigger for proceeding through the samples of a text. An increasing involvement with texts in my work since 1997 has led to a desire to expand the techniques developed in *27matrix* and to develop models of interaction that are based more psychological considerations.

Whereas Mustard's *RoboSax* works focus primarily on the drama inherent in performing interactive music, and *DiceGame* and *27matrix* are both concerned with the real-time generation of fractal structures, *<as viewed>* springs from a desire to re-evaluate the implications and possibilities of computer interaction itself. It was evident that there was a degree of tension between the highly deterministic and linear nature of my earlier works' fractal structure and the inherently non-linear potentialities of real-time interactivity. Traditional musical formal models had been developed, in the main, for a linear environment where complex structures could be coordinated only through the synchronization of multiple performers to a single linear temporal framework. The advent of interactive computer technology capable of creating complex non-linear structures has brought about a need both to re-assess these kinds of formal structures and consider the potential for new ones.

<as viewed> is an initial attempt at developing such a new formal structure. It draws partly on developments in literature such as the ‘hypertext rhizome’. Defined by Slovenian philosopher Slavoj Zizek as being a structure that ‘*does not privilege any*

order of reading or interpretation; there is no ultimate overview of "cognitive mapping", no possibility to unify the dispersed fragments in a coherent encompassing narrative framework. (Zizek, 2000 pp.37)

The hypertext rhizome has been developed into a powerful dramatic paradigm by MIT professor Janet Murray. She coined the term ‘Violence Hub’ to designate hypertextural works in which a central event is examined from different perspectives.

The proliferation of interconnected files is an attempt to answer the perennial and ultimately unanswerable question of why this incident happened. These violence hub stories do not have a single solution like the adventure maze or a refusal of solution like post-modern stories; instead they combine a clear sense of story structure with a multiplicity of meaningful plots. The navigation of the labyrinth is like pacing the floor; a physical manifestation of trying to come to terms with the trauma; it represents the mind’s repeated efforts to keep returning to a shocking event in an effort to absorb it and finally, get past it. (Murray 1997 p. 135-6)

Zizek identifies the potency of this novel formal structure in Lacanian terms as referring to the ‘*trauma of some impossible Real which forever resists its symbolization (all these narratives are ultimately just so many failures to cope with this trauma*) (Zizek 2000 p.38). <*as viewed*> attempts to sonically reproduce a formal structure of this type.

At the heart of <*as viewed*> is a short text. The text is structured in a similar way to Murray’s ‘Violence Hub’: it circles a central irreducible problem. I would, however, like to suggest a more generalized term for this kind of formal structure - ‘Event Hub’ – in recognition of the wide variety of circumstances, not merely violent ones, that one is cable of obsessing over.

Each line of text is recorded as a separate sound file. The computer can choose to replay and manipulate any previously chosen sound file of text, but is constantly narrowing its own number of text choices. In effect the patch left to its own devices will choose to 'obsess' over - in this case repeating and deforming - an ever diminishing group of samples. The live performance 'distracts' this process and forces it to act upon new material until all of the samples have been exhausted.

The MAX/MSP pitch mapping object `fiddle~` (Puckett et al 1998) forms the bridge between the live performer and the computer. This object is used to approximately map the current frequency and amplitude of sounds from the performer and also makes a guess as to the beginnings of phrases based on amplitude changes. Computer pitch-tracking/score following was one of the tasks MAX was created in 1985 to solve (Chadabe p.183). The effectiveness of this technology still remains an issue for interactive computer music. The accuracy of the CPU intensive object `fiddle~` for example is a trade-off with the number of other objects/processes that can be included in the patch. For this reason <*as viewed*> uses `fiddle~` only to obtain generalized contours of the live performance.

This information is processed by <*as viewed*>in three distinct layers. Layer one cues text samples based on the beginnings of the live performer's phrases. It also manages the samples so that the texts do not play simultaneously and have appropriate pauses between groups of lines of text. Layer two manipulates the samples that have been played up until that point. It uses frequency and amplitude information as well as information pertaining to the amount of activity in the live part to change playback speed, assign loops and loop lengths and pan the samples. The final layer creates an overall mix between the live performance, the expanding text and the manipulated text and processes the result using comb filters and reverbs.

<*as viewed*> responds to a need for new structural models to better take advantage of developments in interactive technology and in particular their non-linear possibilities. Its response is lyrical/abstracted rather than based on any formal linking of 'real' information about the construction of human memory (such as it is understood).

3 Conclusion

Despite their surface similarities the six works in the *RoboSax Project* make use of a great variety of different paradigms of interaction between soloist and electronic sound. In these works the correlation between the form, content and technical means appear to be closely linked. The *RoboSax Project*, developed in a relatively isolated non-institutional environment, are good examples of the potential for the creation of engaging work with minimal resources.

References

Chadabe, J., 1997. *Electric Sound: The Past and Promise of Electronic Music*. Upper Saddle River, New Jersey. Prentice Hall

Murray, J.H., 1997. *Hamlet on the Holodeck: The Future of Narative in Cyberspace*. Cambridge Massachusetts, MIT Press.

Mustard, J.A., 2000. *Program notes for RoboSax IV*

Personal correspondence with the author

Nelson, G. L., 1994. *Real Time Transformation of Musical Material with Fractal Algorithms.* http://timara.con.oberlin.edu/~glnelson/gnelson.htm

Rowe, R.,1993. Interactive Music Systems: Machine Listening and Composing. Cambridge Massachusetts, MIT Press.

Puckett, M., T. Apel, and D. Zicarelli, 1998. Realitme audio analysis tools for Pd and MSP. *Proceedings of the International Computer Music Conference.* International Computer Music Association, *1998* pp. 109-112

Winkler, T., 1998. *Composing Interactive Music: techniques and ideas using Max.* Cambridge Massachusetts, MIT Press.

Yadegari, S.D., 1991. "Using Self-Similarity for Sound/Music Synthesis." *Proceedings of the International Computer Music Conference.* International Computer Music Association, pp. 423-424.

Zizek, S., 2000. *The Art of the Ridiculous Sublime*. Seattle Washington, Walter Chapin Centre for the Humanities: Occaisional Papers 1, University of Washington Press.