Chroma

Newsletter of the Australian Computer Music Association, Inc.
PO Box 4136 Melbourne University VIC 3052

Number 3 & 4
DECEMBER 89

In This Issue

ACMA CD in 1990

During 1990 ACMA will be producing a CD of recent Computer Music by ACMA members. Expressions of interest are invited from those would like to have a piece included.

At this stage, one likely scenario would be that the production of the CD would be funded by the represented composers themselves, with ACMA providing coordination and distribution. It depends, of course, on how many people are interested, but the overall cost of production (roughly $5,000 for a run of 500 copies, with printing) would be met by those composers represented. So for example, with 75 minutes available on a CD, 10 pieces averaging 7.5 minutes each, means each composer would contribute $500. Half the CD's produced could be returned to the composers to dispose of as they wish and half retained by ACMA for sale to members (at discount), the general public, librairies, etc. A percentage of these sales would go to ACMA to cover costs.

This is just one possible arrangement, but if you're interested drop a note to The Secretary, including details of the piece, title, duration etc. If there's a big response, we may have to narrow it down somehow, but we'll cross that bridge when we come to it.

If we can get this to work for 1990, there's no reason why we can't produce a CD each year!

This double issue of Chroma is the last for 1989. Chroma 5 is due out in March 1990 and we're looking for contributions.

I know it's difficult with a quarterly newsletter to give advance notice of upcoming events, but there are a lot of performances going on out there that other people would like to know about. So if you have something going on let us know so we can include it in Chroma. Likewise, reviews are a useful way of communicating with those who didn't get to your performance, as well as a means of documenting them. Chris Knowles' review of the MIMA performances in Melbourne is something we hope to see more of.

This issue also contains some practical, down to earth, articles from Jim Sosnin on accessing the sound hardware on the Amiga, and from Roger Glanville-Hicks on getting MIDI files into *Finale*, the Macintosh music notation program, and Warren Burt summarises features of *Sound Globs* and *M*. David Worrall gives an overview on *floating exceptions*' portable, geodesic dome for multi-media performance, and Larry Polansky, one of the authors of *HMSL* is interviewed by Alistair Riddell. There are also outlines of the Canberra Institute of the Arts and La Trobe University Postgraduate courses at in Music Technology.

Subscriptions for 1990 are now due. Please send your renewals now (still only $10 for individuals) to ensure continuity. A Membership application is on the last page of this issue. Naturally, new members are welcome.

Anthology of Australian Music on Disc

On 6 December the Anthology of Australian Music was lanched in the Mural Hall in Parliament House, Canberra by none other than Hazel Hawke. The AAM is a compilation of recordings of recent compositions and performances by Australian musicians. In all there are 15 CDs accompanied by a comprehensive book containing biographies, annotations, essays, photographs and analyses. The recordings include performances by Flederman, Synergy, Petra String Quartet, Canberra Wind Soloists, Zocchi and numerous others. Represented composers include Banks, Wesley-Smith, Vine, Conyngham, Meale, Sitsky, Hollier, Schiemer, Hair, Edwards, Dreyfus, Sculthorpe, Brumby, Tibbetts, Boyd and many others.

A limited edition boxed set of all 15 CDs and the book is available for $300 plus postage and handling. Individual CDs are available at $22 each plus p&h. The book is available separately for $30 plus p&h.

There are 3 CDs of electronic music from a variety of composers. The 3 CDs contain works by the following composers:

CSM4 - Cary, Gerrard, Parish, Worrall, Pompili and Riddell.

CSM5 - Worrall, Tahourdin, Exton, Fredericks and Cary.

CSM6 - Burt, Milsom, Altoff, Mann, Mummé and Chesworth.

AAM was produced by the Canberra School of Music. To order, contact:

Anthology of Australian Music on Disc
Canberra School of Music
GPO Box 804
Canberra ACT 2601
phone (062) 49 5734

Dear Editor,

Seeing that I've written a couple of things for this issue, I thought I might just as well write in the "letters to .." as well!

Firstly let me say that I'm impressed by the presentation and layout. Those of us who spend a considerable portion of their day using wordprocessors, know only too well the time that goes into setting this kind of quality document. Well Done! One small point: My copy of Chroma 2 arrived with a blank page 11 - the first page of the Members List, and I wonder whether others missed out too.

What do you and other members of the Association feel about sharing the mailing list with the members? There's so much going on in this field and so little intracontinental communication that it would be great to build up a "who's who" mailing list. This list could be added to by contributors/members. We in Canberra would be happy to maintain the list and even set up a database of contacts and interests if people are interested.

Perhaps if anyone objects to being on a distributed list they could write (electronic or snail mail) or use some other form of analog or digital means of communicating this to you and they could be left off the list. People signing on or re-signing could be asked if they object to being on this list.

Yours sincerely,
David Worrall October 30, 1989.

Dear David,

Whoops! Sorry about your page 11. I've included a copy of p11 and 12 with your copy of this issue. Checking through the back issues of Chroma 2, I can't find any others with pages missing, so I hope this was just a one off error. However, if anybody else has received a degenerate copy, please let me know.

I have informally discussed making the database of members available, for other than ACMA business, with the Secretary and we agree it should go on the agenda for the next Committee meeting. Perhaps this is something that all the members would like to put their opinion on. I think it's a difficult question; we all want to see our privacy protected to the extent that we don't get a mail box full of irrelevant junk. But, on the other hand we would all be interested in getting information on music technology and related stuff. One way of avoiding the problem is that Chroma should be seen and used as the means of passing on relevant information to members. Afterall, this is a member's newsletter and if there's anything at all that you want to let members know about, Chroma is theplace to put it; that's what its for!

The alternative, if you think about it, is 80 or so copies of the database, with possibly 80 mail outs on matters that could be covered here. It's a matter of efficiency and economy too, is what I'm suggesting.

If any members have a feeling on this issue, please write to let us know, as soon as possible. [Ed.]

National Composers' Conference 1990

In 1990 the National Composers' Conference, organised by the Australian Music Centre, is to be held in Brsbane, probably from 17-20 August.

The AMC has invited ACMA to participate and is looking for a proposal from us as to what we want to do. There are various ways in which we might contribute to the conference, e.g. concert(s) of works, presentation of papers, panel discussions etc. Topics could cover anything from technical reports, to analyses of works, speculative aesthetics to (as Larry Polansky might put it) a general "nerd down" on music technology.

This is a real opportunity to participate in open, informed discussion on music technology in Australia. As an added incentive to attend, we will be holding our 1990 AGM in Brisbane during the conference.

We need your ideas and input. If you have any suggestions for events, want to present a paper, or have a proposal for work to be presented at a concert, slip it in the post to:

The Secretary
ACMA, Inc.
PO Box 4136
Melbourne University 3052

We need to put a co-ordinated proposal of what we want to do, to the AMC early in 1990, so get those ideas in.

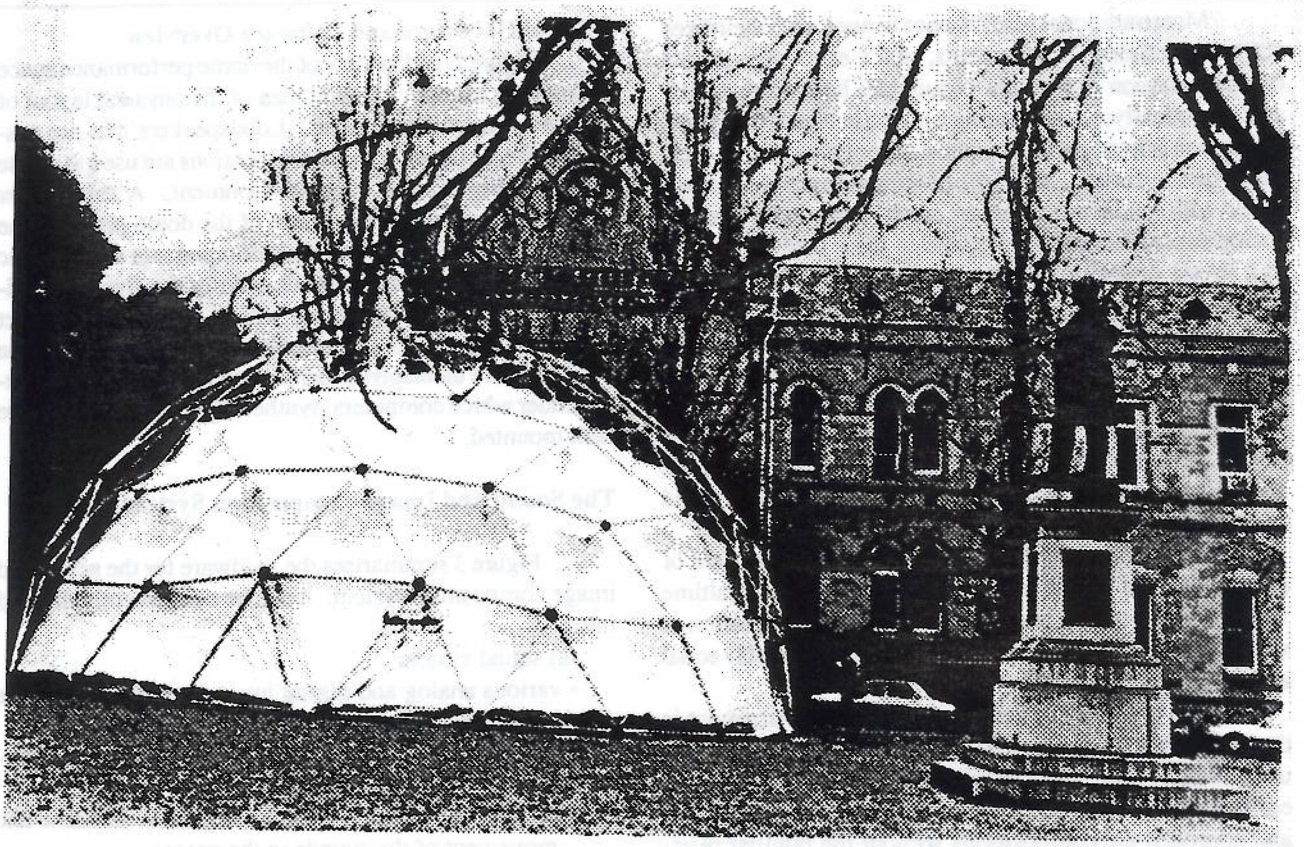


Figure 1. **The *floating exceptions* portable geodesic dome performance space.**

A Music and Image Composition System for a Portable Multichannel Performance Space: A Technical Overview

- David Worrall

This article gives an overview of the sound and light synthesis system for the *floating exceptions* portable performance dome.

The *floating exceptions* Dome

floating exceptions is a group of composers and computer artists who formed in 1985 to produce electrospatial art; that is, omnidirectional coordinated computer-generated electroacoustic music and visual imagery. They have designed and built their own portable performance venue - a seven meter high white geodesic dome (a four frequency icosahedral breakdown geodesic hemisphere, class II method II) with a seating capacity of approximately 200. (Figure 1). The dome is currently located in Canberra.

Members of the group at present and their main interests are Stuart Ramsden (graphics, hardware and software), Virginia Read (music, software) Kimmo Vennonen (music, hardware) and myself (director, music, software). The then current state of this research was summarised in the recent work *Life Dreaming*, which was commissioned by the Australian Bicentennial Authority and given multiple performances in the New Directions Festival in Sydney September 1988.

Aesthetic Considerations

In making music, composers work with our capacity as humans to perceive different characteristics of sound: pitch, duration, dynamics and timbre (tone colour). Our ability to perceive variations in these characteristics is limited. We seem to perceive some characteristics more acutely than others. An intuitive understanding of this phenomenon by past composers has led to more compositions with, for example, wide variations in pitch at a constant dynamic level than compositions in which the pitch is constant and dynamics vary. The widespread and detailed construction of tuning systems, scales, modes and tonalities in the pitch domain compared to only the most general dynamic contouring seems to support this view.

The recording and broadcasting of instrumental music performances and the standard techniques of electroacoustic music involve simultaneous control of only a limited number of channels of sound. As humans we have quite acute sonic location abilities, yet relatively few past composers utilized this in anything but the most general ways– possibly because of the technical difficulties of doing so!

Once I became aware of the two dimensionality of what I was doing I realized that it would be much more exciting if the music could 'leave the ground' and become truly three-dimensional. The artistic feasibility of this approach was enhanced or supported by the knowledge that physiologically, human beings have an extremely well developed ability to locate certain types of sounds in three-dimensional space.

Most traditional performance venues, with their fixed stage and seating arrangements, don't encourage spatial experimentation and so are unsuitable, both aesthetically and acoustically, for presenting a music in which the spatial dimension is important. This music is not proscenium arch, stage-focused music. Thus, the presentation problems associated with spatial multichannel music, to say nothing of the sociopolitical model encapsulated in these theatres, led me to a strong dissatisfaction with conventional performance venues.

This dome project then, was born out of a dream to find a more flexible and truly three dimensional performance venue. The use of multiple channels allows one to concentrate on the volume of the enclosed space and the way sound is distributed in it, rather than a 'stage' performance, allowing the making of music in which the location of sound can be treated as an independent variable.

With a combined expertise and the current state of electronics it is possible to incorporate dynamic realtime visual synthesis into a composition, allowing the creation of works in which both the aural and visual domains play equal and complementary roles.

One gratuity in performing in this environment is that audiences of all ages, far from being distracted by the unfamiliarity and "high tech" nature of the environment, easily accept the new set of listening and viewing conditions and approach the compositions without the familiar resistance that many people have to electroacoustic music.

General Hardware and Software Overview

Figure 2, a plan view of the dome performance space without it's canopy, gives an idea of the physical layout of this environment/instrument. Loudspeakers that are suspended from sixteen equidistant locations are used to create the three dimensional aural environment. A sub-woofer system on the floor in the centre of the dome provides the low frequency (less directional) components of the sonic spectrum. Surrounding this sub-woofer are five appropriately angled video projectors which are used to project graphic images onto the screens opposite them. These projectors are themselves surrounded by a decagonal desk-top under which computers, synthesisers, and amplifiers are rack mounted.

The Sound and Image Composition System

Figure 3 summarises the hardware for the sound and image composition system. It can be conveniently divided into:

(a) sound system

- various analog and digital input devices such as tape recorders, synthesisers, samplers;
- a sixteen channel equalisation and amplification playback network;
- a spatial distributor for controlling the localisation and movement of the sounds in the space;
- computers for music composition running our own realtime MIDI event generator software, written in FORTH.

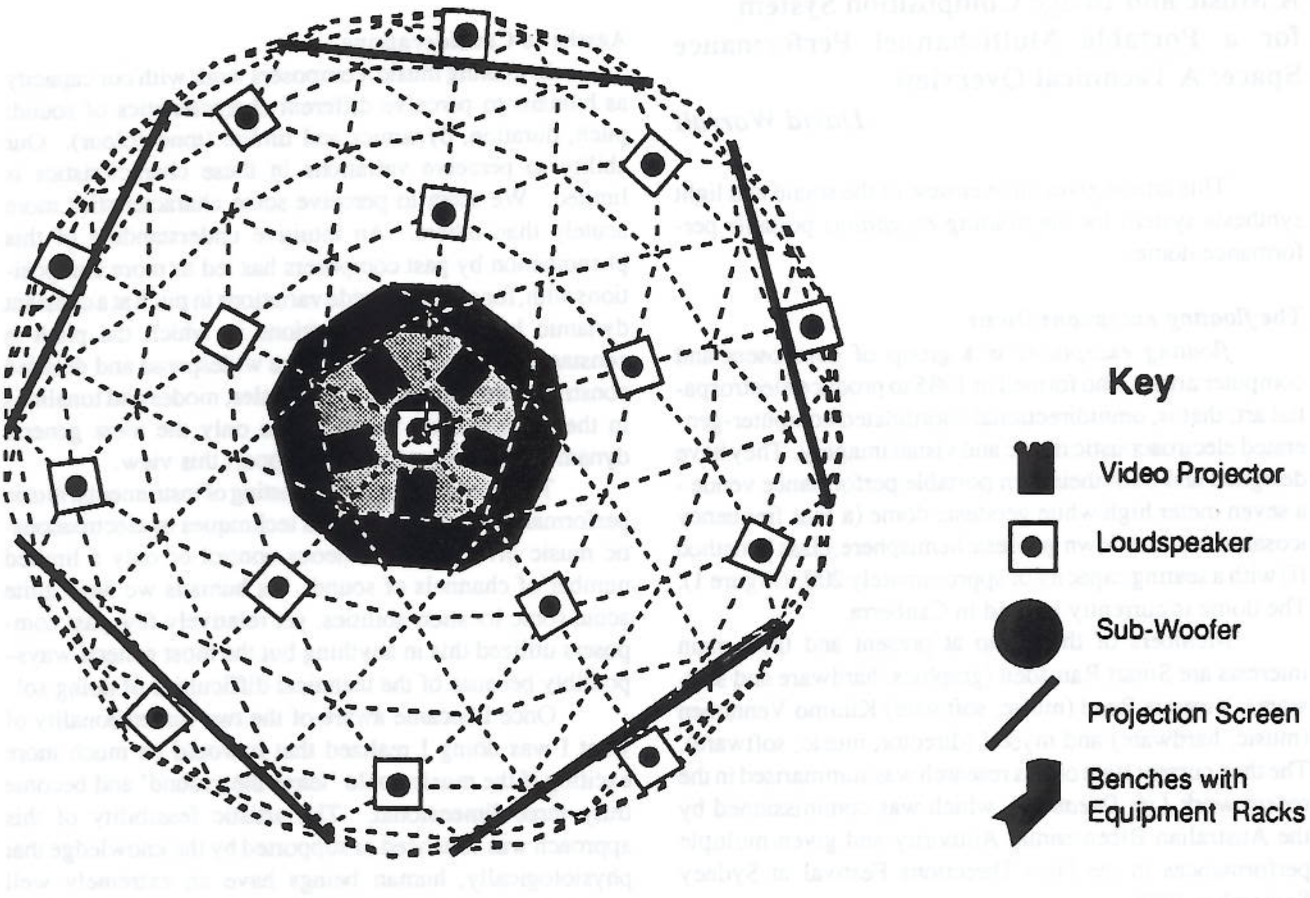


Figure 2. A plan view of the geodesic dome performance space

(b) graphics system

- five computers for generating graphic images in real time, one for each of the projectors;
- a computer for generating image composition and image location instructions using our own realtime software, written in FORTH;
- a Spatial Image Distributor for channelling image synthesis instructions to the various graphic engines;

(c) Various devices for data input and realtime control of sound and graphics:

- mouse
- ASCII keyboard
- track ball
- joy-stick
- MIDI keyboard
- Pitch to MIDI convertors

To my knowledge this work is unique. From a technical viewpoint, the most interesting and innovative aspects are:

- the Spatial Sound Distributor which uses custom built hardware controlled by a 'FORTH on a chip' (a Maestro supercomputer) controlled by an Applix 1616 microcomputer and the MIDI and other interfaces to it;
- the software for realtime music and image composition and the image/music communication protocols;
- the Spatial Image Distributor, a Maestro supercomputer, which translates high level image descriptions into synthesis instructions for particular image generating computers;
- the omnidirectionality of the environment and its aesthetic implications.

At present this work is almost entirely funded by the individuals involved.

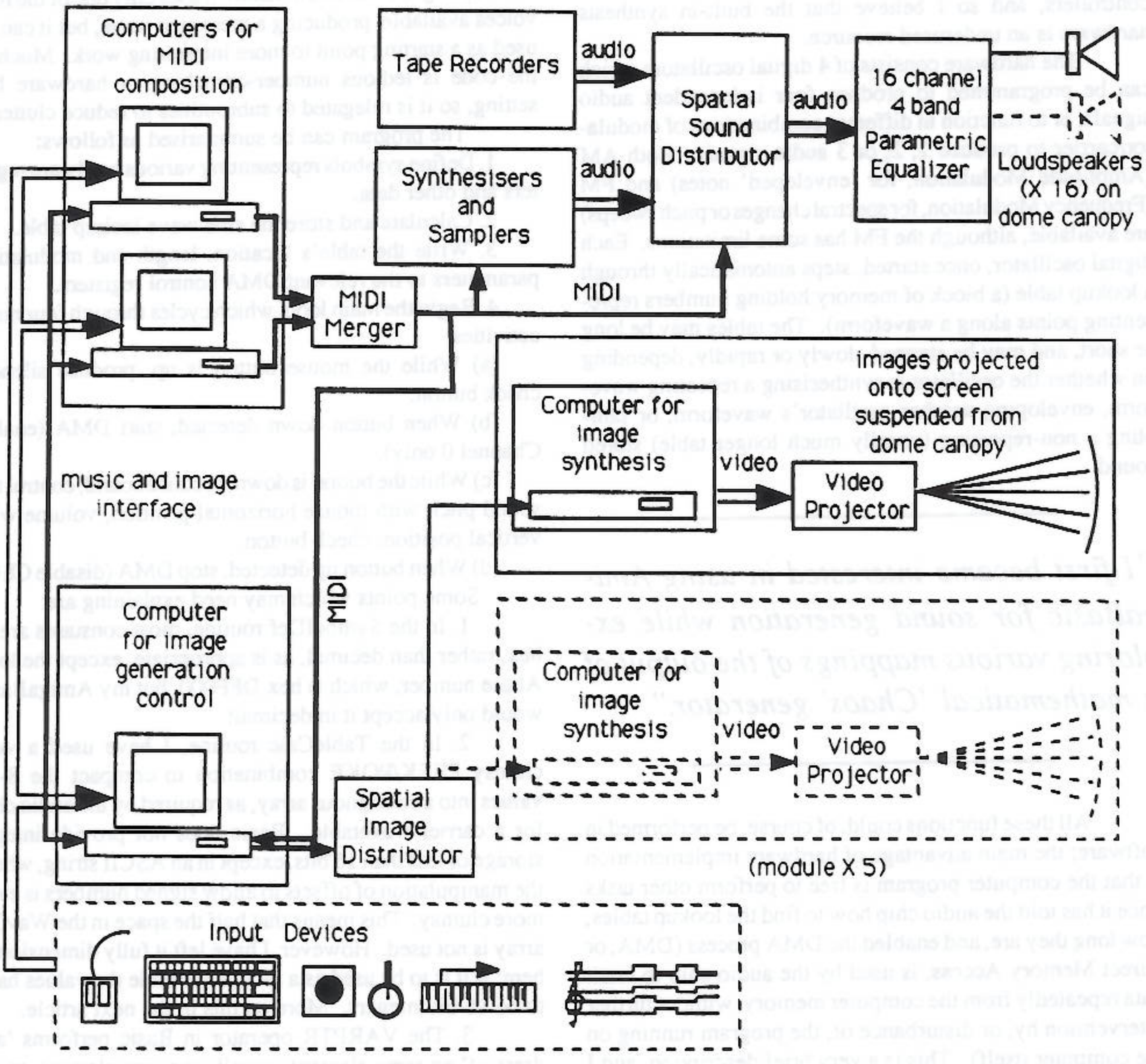


Figure 3. Diagram of the music and graphics composition system for the geodesic dome performance space.

AmigaBasic Real-Time Sound Synthesis

- Jim Sosnin

This introductory article allows musicians with access to an Amiga computer to explore real-time sound synthesis, utilising only the hardware facilities and Basic language that are supplied as standard. Limitations of Basic's inbuilt SOUND calls are overcome by accessing the audio hardware directly with peeks and pokes.

All Amiga owners must be proud of the spectacular colour graphics capabilities of the Amiga; most would also be aware of the less dazzling, but rather powerful 4-channel digital audio synthesis hardware built into the machine, although in general, commercial software utilises this only for replaying sampled audio to accompany games etc. Musicians are more likely to use their Amigas as MIDI controllers, and so I believe that the built-in synthesis hardware is an underused resource.

The hardware consists of 4 digital oscillators which can be programmed to produce four independent audio signals, or to function in different combinations of modulator/carrier to produce 1, 2, or 3 audio signals. Both AM (Amplitude Modulation, for 'enveloped' notes) and FM (Frequency Modulation, for spectral changes or pitch sweeps) are available, although the FM has some limitations. Each digital oscillator, once started, steps automatically through a lookup table (a block of memory holding numbers representing points along a waveform). The tables may be long or short, and may be stepped slowly or rapidly, depending on whether the oscillator is synthesising a repeating waveform, enveloping another oscillator's waveform, or sampling a non-repeating (usually much longer table) stored sound.

> *"I first became interested in using AmigaBasic for sound generation while exploring various mappings of the output of a mathematical 'Chaos' generator."*

All these functions could, of course, be performed in software; the main advantage of hardware implementation is that the computer program is free to perform other tasks once it has told the audio chip how to find the lookup tables, how long they are, and enabled the DMA process (DMA, or Direct Memory Access, is used by the audio chip to fetch data repeatedly from the computer memory, without further intervention by, or disturbance of, the program running on the computer itself). This is a very brief description, and I have omitted many details, but it all means that, even if the program is written in a slow language such as AmigaBasic, polyphonic real-time sound synthesis, with continuous or 'dynamic' control of many parameters, is possible.

I first became interested in using AmigaBasic for sound generation while exploring various mappings of the output of a mathematical 'Chaos' generator. AmigaBasic was ideal for trying out lots of different program and data combinations quickly, without having to wait for recompiling, linking, etc, but the sound was limited to fixed waveforms. Since I had already developed various 68000 Assembler routines for accessing the audio chip, producing much more interesting sound textures, I decided that translation to Basic would be worthwhile. I now have Basic routines that produce enveloped sounds with independent pitch and duration, compound sounds with mouse control of crossfade between spectra, and even some FM, although very little dynamic control is possible in this case.

The accompanying example program, the AmigaBasic Mouse Theremin, is neither exciting nor useful, but it does demonstrate how two parameters of a synthesised sound can be controlled continuously, in real time, without interrupting the waveform itself. It uses only one of the four voices available, producing a mere sine wave, but it can be used as a starting point to more interesting work. Much of the code is tedious number-crunching or hardware bit-setting, so it is relegated to subroutines to reduce clutter.

The program can be summarised as follows:

1. Define symbols representing various hardware registers and other data.
2. Calculate and store the sine wave lookup table.
3. Write the table's location, length and modulation parameters to the relevant DMA control registers.
4. Begin the main loop, which cycles through four main activities:

(a) While the mouse button is up, produce silence, check button.

(b) When button down detected, start DMA (enable Channel 0 only).

(c) While the button is down, produce sound; control the sound pitch with mouse horizontal position, volume with vertical position; check button.

(d) When button up detected, stop DMA (disable Ch 0).

Some points which may need explaining are:

1. In the SymbolDef routine, most constants are in hex, rather than decimal, as is appropriate, except the long Abase number, which is hex DFF000, but my AmigaBasic would only accept it in decimal!

2. In the TableCalc routine, I have used a very clumsy PEEK/POKE combination to compact the 8-bit values into a contiguous array, as required by the audio chip for a carrier wavetable. Basic does not provide integer storage of less than 16 bits, except in an ASCII string, where the manipulation of offsets to allow signed numbers is even more clumsy. This means that half the space in the Wav0% array is not used. However, I have left it fully dimensioned here; if it is to be used as a modulator table the values have to be 16-bit integers. More on this in the next article.

3. The VARPTR operator in Basic performs 'address of' an array element (usually the first element, to set a pointer) or of a simple variable. If the variable occupies more than one byte, then VARPTR points to the MSbyte, due to 68000 architecture. Hence PEEK (VARPTR (SinVal%) + 1) to access the desired LSbyte of SinVal%,

which occupies 2 bytes. The use of VARPTR is full of traps: if a new variable is introduced after VARPTR is used, the address is shifted by Basic, and the pointer will point to the wrong place. I have used dummy assignments early in the code (e.g. MX = 0) to fix the space allocation right from the start. If there is a Basic guru out there, please tell us if there is a better way!

4. In the DmaParams routine, extended versions of POKE facilitate the data transfer to 16-bit registers (POKEW), and 32-bit registers (POKEL).

5. In the main loop, the DmaCon control register works as follows: the state of bit 31, the MSbit, determines whether any other selected bits are set or reset. When the MSbit is set, (value &H8000), any other '1' bits will be set to the active state. When the MSbit is reset (value 0), any other '1' bits will be reset to the inactive state. In either case, any other '0' bits will not be changed from their set or reset condition. This means that if you wish to enable more DMA activity, e.g. to have two waveforms sounding, then at some point you will need to add extra code for the corresponding disable.

Example:

POKEW DmaCon, (Set + Aud0En + Aud1En): rem enable
POKEW DmaCon, (Clr + Aud0En + Aud1En): rem disable

Remember that, for this to work, you will have had to prepare all the other Ch 1 parameters also.

6. On the AmigaBasic screen, the mouse 1,2 coordinates range from (0,0) to (636,244). The minimum allowed Aud0Per value is 124, and Aud0Vol must range from 0 to 64. The MX, MY offset and scaling allow the ranges to match fairly closely.

7. The program runs reliably on an Amiga 1000 with the standard half Megabyte of RAM. If you have an Amiga 500, 1000 or 2000 with extra memory, hide it with 'NOFastMem' beforeyou run AmigaBasic, otherwise the lookup table may be written into higher memory where the DMA hardware cannot access it.

If there is some interest, in future issues I will describe AmigaBasic programs for AM, FM, and otherwise generating dynamic spectra, and extended mouse techniques such as using velocity for 'bowing' effects and phase shifts.

References:

Amiga Hardware Reference Manual, Addison-Wesley Inc.

Amiga Basic (Microsoft Basic for the Amiga), Commodore-Amiga Inc.

Program Example - AmigaBasic Mouse Theremin (cont'd on page 8)

```
REM AmigaBasic Mouse Theremin (mouse controlled sound generator).
REM orig 29-Nov-89 Jim Sosnin.  Mouse controls pitch & vol, 1 voice only.
REM edit 30-Nov-89 JVS fix wavetable so only LSbyte values remain.

REM The DIM, as usual, reserves space for array variables, and in this
REM program, all simple variables also are initialised (unusual) so that
REM their space is reserved too, and VARPTR works correctly later on.

TabLen = 50:          REM length of waveform lookup table.
DIM Wav0%(TabLen)
MX = 0: MY = 0:       REM storage for mouse horiz and vert values.
AngInc = 3.14159 * 2 / TabLen:   REM i.e. TwoPi / number of steps.

GOSUB SymbolDef:    REM define symbols for hardware addresses.
GOSUB TableCalc:    REM calculate sine wave table.
GOSUB DmaParams:    REM prepare audio hardware for DMA.

PRINT: PRINT " Press left mouse button, move to control pitch and volume."
PRINT " Select STOP from the RUN menu to quit."

WHILE 1:              REM a 'forever' loop.

  WHILE MOUSE(0) = 0
                      REM wait till mouse left button down.
  WEND
                      REM mouse down now.
  POKEW DmaCon, ( Set + Aud0En ):  REM start DMA sound, Ch 0 osc.
  WHILE MOUSE(0) < 0:     REM continue sound while mouse down.
    MX = 760 - MOUSE(1)
    MY = MOUSE(2)/4:   REM new scaled values each time.
```

```
      POKEW Aud0Per, MX
      POKEW Aud0Vol, MY:    REM simple example, 2 param only.
    WEND
    POKEW DmaCon, ( Clr + Aud0En ):   REM turn sound off, await mouse.

WEND:                       REM end of main program.

SymbolDef:

  Abase = 14675968&:        REM Amiga audio hardware base address.
  DmaCon = Abase + &H96:      REM Direct Memory Access control reg.
  ModCon = Abase + &H9E:      REM Modulation Control reg (AM & FM).

  Aud0Loc = Abase + &HA0: REM stores Ch 0 lookup table location.
  Aud0Len = Abase + &HA4: REM stores Ch 0 lookup table length/2.
  Aud0Per = Abase + &HA6: REM stores Ch 0 period, DMA 'ticks'.
  Aud0Vol = Abase + &HA8: REM stores Ch 0 volume.

  REM for Ch 1, Ch 2, Ch 3 sets of corresponding control registers, add
  REM hex &H10, &H20, &H30 respectively, to above Ch 0 addresses.
  REM e.g., Aud1Loc = Abase + &HB0.

  Set = &H8000:          REM used with DmaCon and ModCon.
  Clr = &H0
  Aud0En = &H1:          REM LSbit to enable/disable Ch 0 DMA.
  Aud1En = &H2:          REM next bit for Ch 1, etc.
  Aud2En = &H4
  Aud3En = &H8

RETURN

TableCalc:

  PRINT "Generating Wavetable ..";

  FOR i = 0 TO TabLen-1
    SinVal% = SIN( AngInc*i ) * 127
        REM SinVal% is 16-bit variable, but only 8 LSbits used.
    POKE VARPTR( Wav0%(0) )+i, PEEK ( VARPTR(SinVal%) + 1 )
        REM above line packs array with 8-bit elements only.
        REM note that the 68000 stores LSbyte at higher address.
    PRINT ".";:     REM progress indicator.
  NEXT i

  PRINT: PRINT: PRINT "                         Wavetable ready."

RETURN

DmaParams:

  POKEL Aud0Loc, VARPTR ( Wav0%(0) ): REM pointer to start of table.
  POKEW Aud0Len, TabLen / 2:     REM 8-bit table entries, WordArray/2.
  POKEW ModCon, ( Clr + &HFF ):  REM ensure no modulation, this example.

RETURN
```

Jim Sosnin's AmigaBasic Mouse Theremin, continued

Larry Polansky on HMSL and Computer Music.

- Interviewed by Alistair Riddell

(Melbourne 2/8/89)

AR Larry, you are primarily known as a computer music composer because of your involvement with HMSL, what is your background in both computer and electronic music? When did you first encounter the computer?

LP When I was 18 and in my first year at college. Until then I was primarily a jazz player, also playing traditional American music. I was a gigging professional musician but eventually decided to focus my musical energies on composition, although I still perform a great deal. I first encountered the computer as a mathematics student. Mathematics is important to me in my understanding of music, and I actually got a degree in math, concentrating mostly in pure math. I see that now I was interested in form, and that early interest in topology and set theory seems to me to be very akin to my current interests in musical form. My first programs were simple little "AI" experiments: counterpoint writing programs and the like on a PDP 11/45. This was at a small college in Florida.

AR Did you work in that area when you went to Illinois?

LP No, I went to Illinois several years later. After I left Florida, I met James Tenney at the University of California and started working with him.

AR Tenney, as I understand, had been quite involved with computers.

LP Tenney is a very important figure, although not as well known as I think he should be. He was, for example, instrumental in adding algorithmic power to the early digital synthesis programs at Bell Labs. I believe that Jim (and others of course) encouraged Max Matthews to add provisions for writing one's own compositional subroutines to generate score files. Tenney, although also interested in timbral experimentation, said, more or less: "well, what we really want to do is compose." I think his interest in using the computer as a compositional mind was very much influenced by his association with Cage. He saw the power of the computer as a compositional and perceptual modelling tool. In the early 70's Tenney and I worked on a computer program to model hierarchical temporal gestalt perception. The results of this project were later published in the Journal of Music Theory, and a few other places. I still think it was and is a quite extraordinary program, even though it's quite small (and of all things, written in Fortran). But we worked for about two years in refining it, to make the program a very integral model of the theory, which was primarily Jim's. It was a great experience for me, to work with someone like that so closely, for so long.

Around that time I also worked in many different types of technology. I worked at Stanford, and I worked a lot with computer controlled analogue equipment. Basically, whatever I could get my hands on I used. At the University of California at Santa Cruz I worked with Bob Hoover, who later became founder of 'Mimetics' and responsible for the Amiga Soundscape software. He and I interfaced an old U.S. navy reject Interdata Model 3 computer, and had it controlling Moogs, and even did some live performances with it (I remember Gordon Mumma somehow magically coming up with a truck for us to move this thing in!). This kind of work was all machine language programming, with no mass storage, and very slow. In those days, the early 70's, there wasn't much around in terms of computing power (at least for me), and one had to be fairly resourceful. You could work very hard and produce very little, but you learned a lot and it was certainly fun. Of course, I've worked on a lot of different systems since then!

AR You have also been paralleling your work in electronics with writing for traditional instruments.

LP I really like writing for instruments and I like working with performers. I have never been exclusively interested in electronic sounds, and this is perhaps some sort of artifact of my interest in compositional artificial intelligence. I do work with various synthesis techniques, and have done a bit of experimental work in that area, but that has never been my area of speciality. I'm much more interested in generative, formal, and compositional processes (although sometimes, happily, these two areas of computer music intersect). Much of my computer music work results in instrumental works. Strangely enough, this focus may be shifting. I've recently been working on a synthesis algorithm which stems from what was essentially a formal process. But mainly I've worked with whatever sound producing means have been immediately available, and often they have happened to be performers. Many of my close friends are great performers, so I like to work with them. For example, I wrote a computer generated flute piece

> *"I do work with various synthesis techniques, and have done a bit of experimental work in that area, but that has never been my area of speciality. I'm much more interested in generative, formal, and compositional processes..."*

(*V'leem'shol*) for Ann Laberge, and there's the tap dancer pieces, things like that. With someone like Ann, you hear her play, and you have to ask yourself "How can I NOT write for someone like that"? It really goes both ways. Most of my recent music has been more or less for solo interactive computer and one performer (like *B'rey'sheet*, which I do with my wife Jody Diamond, and which we did at the Astra Choir concert in Melbourne in August. On that concert I also did a duet with Chris Mann, which will be released soon on an Artifact Records CD). I like this situation a lot. It seems to integrate many of my interests. I get to work with my performer friends and also do live computer music. The synthesis aspect has been a hard one for me to deal with. On the one hand I'm not really that attracted to tape pieces and that seems to me the main area of really interesting synthesis up to now, especially if you are not, as I am not, a hardware genius. If one were really committed to the most powerful synthesis techniques, it seems to me it would be difficult to reconcile that with a commitment to live electronic performance at this time (although with all the new DSP developments, this is certainly changing!). Unhappily for me, and I think many other composers find themselves in this situation, I've had to accept a lot of sound producing stuff which I didn't build or design myself. I've always concentrated on software design and theory, but it's not a comfortable situation for me to use a lot of other musical stuff designed by other people, especially those with more commercial intentions. I've been pretty successful at doing some odd things with pre-existing gear, but I don't really think that ultimately this is satisfactory. So great performers for me have been a kind of a way out. They are a way of using a sound producing means that is accepted, and I think pretty honest.

"I really feel that I am part of a community and I think we're all in danger of being kind of precious about our work. I think that's an old way of thinking about music. One way to help usher in the millenium perhaps is to acknowledge that community, and it's beauty."

I want perhaps to take this notion of sound one step further, because I think we're at a point where we're all going to change our thinking. For me, I feel like I can personally begin working seriously in the combination of live computer work, software development, and synthesis again, because of all the fantastic developments in small cheap signal processors. The 56000, TMS320-x0 series, and other chips are making it possible to do very interesting things in in more or less real-time. In fact, several composers, like Tim Perkis for instance, have been working in this area for some years. And of course they're getting faster and faster. The NeXT machine is also changing a lot of minds. Admittedly it's still expensive, but not like a VAX. Even the Amiga was a real revolution to me. Four channels of DMA sound. Ok, it's 8 bit, and sounds pretty funky but you didn't have to solder and you didn't have to design circuits, and you got a very hands-on access to the waveforms themselves. I've been able to do a lot of interesting experimentation with that aspect of the machine, and people like Robert Marsanyi have done some extraordinary work in this area.

AR In looking over some of your compositions I noticed that they're all dedicated to people and that you acknowledge them and their ideas a great deal in your work.

LP Yes, although I'm very positive about my own work, I'm not shy about references. In fact I believe very strongly in our interconnections. I really feel that I am part of a community and I think we're all in danger of being kind of precious about our work. I think that's an old way of thinking about music. One way to help usher in the millenium perhaps is to acknowledge that community, and it's beauty. When I do pieces like the *Distance musics* (published in 'Perspectives') where every piece is a tribute to some other composer (but I suppose, ultimately, very much my own) I am trying to acknowledge that very heterarchical intellectual and musical community. I think a lot of composers are shy about acknowledging influences, they say "well, that's MY own idea" ... and so on, like musical ideas are some kind of possession. But I feel part of a community of mind – I WANT to be part of it and I WANT to help engender it. Again this probably comes back to something like HMSL, it's very much a group action.

I think that if one really looks at say, the early work at the San Francisco tape centre, composers like Don Buchla, David Rosenboom, Tony Gnazzo, Ramone Sender, and especially the League of Automatic Music Composers, were all working towards a musical community – and that's not a dead idea, it just somehow got overlooked in certain areas of our musical environment. The technology wasn't quite there in the sixties for certain experiments to this end (although I think one could say that David Rosenboom's biofeedback work is pioneering in this respect) but now it is! Now we're on BITnet, we're sending each other discs, we're doing lots of interactive and communicative pieces, we're working on code collaboratively and, of course, these methods and ideas will evolve and change radically and wonderfully in the next five years. We can't even imagine what those ways will be. Perhaps we'll be sticking electrodes on our heads and thinking pieces. I like that a lot. I'm of course not originating these ideas, but I am part of a lot of this work, and I'm grateful for this.

AR Could you tell us something of the history of HMSL?

LP David Rosenboom, James Tenney and I had been very close friends for many years. We had worked together in various ways and shared similar ideas about form, transformations of forms, recognition of forms and computer modelling. We shared an office together in 1976 while we were writing that "perceptron" program I mentioned earlier. David was writing some similar programs of his own as part of his 'On Being Invisible' series, to analyse responses from the brain.

So it really has a theoretical underpinning from three people who were thinking about some common issues. I think Jim is the theoretical godfather of the whole thing. His work in this area goes way back to 'Meta-Hodos' in the early sixties, and his insights still exist in various forms in HMSL. I think if one looks at David's earliest electronic music, one also sees a deep concern with the idea of languages. This evolved quite naturally, I think, into a concern for language environments for composition and performance. He participated in the development of the Buchla/Crowe language, Patch-IV, which was a terrific hybrid control environment, and also wrote the language FOIL (Far Out Instrument Language) for the Touche, which he built with Don Buchla. We began thinking about implementing a very general and very powerful language for small computers that a lot of people could use in radically diverse experimental situations. Our interest in this, combined with our common theoretical bent and the advent of 16-bit microprocessors in the late 70's, led to the inception of HMSL. I think David saw HMSL as the next generation after things like 'PatchIV' and 'FOIL', and it really does come somewhat from that tradition of flexible experimental environments. Of course, it's become a LOT more than that, but in my very earliest prototypes I was interested in abstracting, for example, things like the notion of definable stimulus/response events that was so wonderful in PatchIV. David brought me to Mills College (Center for Contemporary Music) in 1980/81 and we set to work on writing HMSL. We built a 68000 S-100 system for the prototype. We also started the Seminars On Formal Methods, which were focused on formal systems of thought about music and language, and we tried to get lots of interesting people to Mills so that we could all talk about these ideas and work with together. Dan Kelley, for example, was just starting to think about 'Masc' at the time, and in fact 'Masc' and HMSL share some basic routines. We were trading Forth routines with lots of people at the time. One spring we brought Ron Kuivila to the CCM, while he was working on the first version of Formula, and I was playing with some simple HMSL scheduling ideas. Ron contributed some very powerful and interesting code and ideas, and so did of course, many others. We were all sharing ideas. There seemed to be something fascinating in the air at that time, people thinking about experimental music languages because the technology allowed it. Something we couldn't do in quite the same way before.

One of the reasons we picked Forth was that it was kind of a *lingua franca* for small computer music users, especially in the Bay Area. Many composers who did live stuff, like David Behrman, George Lewis, Joel Ryan, John Bischoff, and others, knew it, used it and liked it. It looked like it was going to continue to be very important in that area. David envisioned the CCM as a kind of 'language clearing house' for this kind of work, and it actually was to a great extent. It was someplace people could come to and trade ideas, and try out new things. It was VERY active and VERY busy in those days, and also a lot of fun. Of course, it's STILL busy and active! We wrote the prototype of HMSL in those first couple of years. Much of the time went into the design, simply thinking about what things like the data structures should be called. I wrote the prototype on the S-100 system in Forth with a lot of help from friends like Dan Kelley and Phil Burk, who were just sort of hanging out at the Center, hacking away, and helping me through some gnarly operating system problems.

Phil Burk started hanging around the Center because we were the only people in town with a 68000 running that he could experiment on. This was slightly before the days of the Macintosh and Amiga. He just wanted to play with it. Phil is a computer genius who used to build Z-80 systems, as I used to say, from the body parts of small furry animals in his basement. He was really thrilled about playing around with the 68000 and he was an invaluable aid. We became good friends, and he's such a fantastic programmer and brilliant thinker that he was a natural to add to the HMSL 'team'.

About two years later when the system was more or less up and running at the Center but a bit clunky and hard to use (for example, since it was a 16-bit Forth, you could only have 64k of code!), we were fortunate to be able to bring Phil into the project as a full third design partner. His first contribution was to say "Look, there is this thing called Object-Oriented programming out there that we can probably use" We had actually seen 'SmallTalk' but Phil had been working professionally on a very early compiler for the Mac called 'Neon' and he was very enthused about Object-Oriented programming. He also recognized the natural affinity between the ways we had designed the language and the concepts of OOP languages.

> ***"Phil [Burk] actually wrote the first version of HMSL in Object-Oriented code on a Commodore 64 (!) because that's what he had at home."***

Phil actually wrote the first version of HMSL in Object-Oriented code on a Commodore 64 (!) because that's what he had at home. MIDI didn't exist at the time, but when it came we MIDIfied the system very quickly; that wasn't very hard. Phil wrote an Object-Oriented version for the system at the Center, and that system was used for Pauline Oliveros' *Dear John* (a work for John Cage's 75th birthday, commissioned by the West German Radio) many of my own pieces, some graduate student works, and others. But since this was an S-100 system, it wasn't really portable and was running some really old fashioned technology, like a stand alone S-100 graphics card. Even though there was MIDI, it was tied into a Buchla 400 digital oscillator system - for which we had to kind of hack the interface. Phil used to joke that it was pretty portable – anyone who had an ERG S-100 based 68k system with a Buchla 400 oscillator card could run it! Then the Amiga came out. The Mac had been out for a short time but it was still new. The original Mac's were difficult to program on, and it was clear that there would be some rapid developments that would make them more accessible. We got very excited by the Amiga though – it was cheap, had a nice operating system, lots of power, was fast, and all kinds of interesting features.

Phil became immediately involved writing a Forth compiler called JForth for it; he was a big Forth fan at that time. So he did that, and we got developer status on the Amiga, and within a month or two he had a version of HMSL running on the Amiga, and all of a sudden there wasn't 64K of memory to play with but a couple of Megabytes. Amazingly, I was able to more or less transfer the piece *B'rey' sheet* from the ERG to the Amiga with very little revision.

For the first time HMSL became a good environment that a lot of people could use. That was Version 2.0, which we distributed to various people (like Nick Didkovsky in NY) for Beta testing. The response was very enthusiastic, I think, because it was so 'hardcore', and a lot of composers (like Nick for example) were looking for flexible, powerful environments, and weren't really too put off by the difficulty. In fact, they enjoyed it! We wanted to get a lot of feedback – it was buggy but sort of worked. I also started doing pieces in it, and Phil Burk, Phil Stone and I actually did what may be the first concert ever done entirely with Amiga local sound, in San Francisco. We used HMSL, and had the three computers communicating in various ways. It was strange and very interesting to me! At the same time Mills College became committed to Macintosh technology, and we decided that our strategy for HMSL would be to support a Mac/Amiga parallel. We ported HMSL to the Mac, which took some time because there was a different Forth compiler for the Mac and back then, the Mac operating system wasn't easy to figure out. The port was tricky and consumed Phil for quite a long time. But it worked and HMSL has been more or less free and clear since then.

From that time on we've been concentrating on developing it further, distributing it, teaching it, using it, writing about it, and documenting it. It's been quite a project! It's still changing radically. Version 4.0, which will be out soon, is very different and far superior, I think, to the previous versions. It will have a whole new graphics system, lots of sophisticated MIDI support (like some interesting sequencing stuff, MIDIfiles, user-definable patch editing, a score entry system), and some nice refinements to the data structures themselves which will make them more powerful and I think easier to understand. We're excited about it of course.

"...we got developer status on the Amiga, and within a month or two he had a version of HMSL running on the Amiga, and all of a sudden there wasn't 64K of memory to play with but a couple of Megabytes."

AR On reading the various articles about HMSL that have appeared in recent years, I was struck by both the terminology and conceptual model that has emerged from the work at Mills college. It seems to me that new users will have to confront this before they can mould HMSL to their own way of composing. In other words before they can reject any part of this work they would have to know it fairly well. Is this more or less the case?

LP Well, they certainly would under some circumstances, but they do not necessarily have to get involved too deeply with that part of HMSL. It would be possible to just use HMSL as a programmable MIDI or video generator, although I think there are other systems that can do that as well. But we were interested in the fertility of the system. We wanted to create a deep environment where people could take our technical and philosophical ideas further, or reject aspects of them. HMSL is distributed as source code and it is very well documented. You can carve up the system as deeply as you like, and we'll help! We'll tell anybody anything. If you wanted to rewrite the scheduler itself we'll tell you how to do it. We're very open about it. The idea is that people will do that – make it their own. Many have done exactly this sort of thing, and that gives us a lot of gratification.

AR What is the future of HMSL at the moment?

LP Well 4.0 has to come out. We'll keep distributing it and supporting the community aspect of it. I don't think there is a danger of it becoming obsolete within a few years. I also want to become the most active user of it! After all, I designed it to make the kind of music that interested me, and now I figure I've got a right to take some time and actually make some of those pieces.

Further Reading:

Polansky, Larry, and David Rosenboom. 1985. "HMSL (Hierarchical Music Specification Language) A Real-Time Environment for Formal, Perceptual and Compositional Experimentation." In *Proceedings of the 1985 Computer Music Conference*, ed. Barry Truax, pp. 243-50. San Francisco, CA: Computer Music Association.

Polansky, Larry, David Rosenboom, and Phil Burk. 1987. "HMSL: Overview (version 3.1) and Notes on Intelligent Instrument Design." In *Proceedings of the 1987 International Computer Music Conference*, ed. James Beauchamp, pp.220-27. San Francisco, CA: Computer Music Association.

Polansky, Larry, and John Levin. 1987. "The Mills College Center for Contemporary Music's Seminar in Formal Methods series: A Documentary Survey." *LEONARDO*. (Special issue on Visual Art, Sound, Music and Technology), ed. Larry Polansky. Vol 20, No 2. pp.155-64. Pergamon Press.

Polansky, Larry. 1987. "Paratactical Tuning: An Agenda for the Use of Computers in Experimental Tuning." *Computer Music Journal*, 11:1(61-68).

Tenny, James. 1961. *Meta-Hodos and META Meta-Hodos*, 2nd Edition 1988. ed. Larry Polansky. Oakland, CA: Frog Peak Music.

Sound in Sync – Recent M.I.M.A. Performances

- *Chris Knowles*

Other Pleasures, a season of experimental film and video presented by The Modern Image Makers Association, was held in November at the State Film Theatre in Melbourne. Among the programs was an evening of experimental music performances which were accompanied by the composers' films or videos. M.I.M.A. regularly curates programs which incorporate images and performance, about twice a year, because of the significant amount of work produced in this area. They have been among the most challenging and successful of M.I.M.A.'s programs and this most recent program, curated by Sue McCauley and titled Sound In Sync - In Performance, included works by Warren Burt, David Chesworth and Sonia Leber, and Arf Arf.

WARREN BURT, *A RANDOM WALK AROUND MELBOURNE*.

In each of the ten or twelve films and fifteen or so videos that Warren Burt has produced he's worked with a different idea of juxtaposition. In the "Meditations" videos made in '86 he experimented with the traditional Hollywood approach of producing music that would match the mood suggested by the visuals, however his favourite working relation is one of structural or conceptual analogy between sound and image. The piece that was performed at M.I.M.A. was like a complete separation of church and state, in which Warren poses the question "Can we learn to see and hear contrapuntally?"

In a strict sense the images and music performed at M.I.M.A. had nothing to do with each other, and in fact Warren played the same piece of music without images at Linden the previous Sunday. He has now decided that that piece of music is going to be used as the sound track for another film, and has developed yet another sound track for the film he showed at M.I.M.A.. Warren likes to experiment with combinations of images and sounds, although in this particular piece the film was made using a process that was very similar to the ways in which he often composes music.

The film itself and the music that Warren played are both what Herbert Brun would call tracings left by a process. Warren first selected eleven random locations on a Met transport map and traveled to each spot in random order. At each location he would try to put himself in as Noh minded state as possible and just film five macro shots for twelve seconds each, producing five close-up looks at found objects. To show the transition between the locations he shot random single frames which produced flickering, animated linking sequences.

In this performance Warren was using *Mpc*, Voyetras' version of *M* for the IBM computer, to control a Yamaha TX81Z and an Akai S900. Using *M*, he composed twenty four musical fragments, grouped into six major styles, each containing four sub-groups or versions of their particular orchestration, each of which could potentially go on forever. Many of the fragments were made with random number inputs. He pulled out the good old book of random numbers to choose the pitches, the rhythms and the timbres, and then listened to how they were combined within *M*.

The samples in the Akai were mostly things he had sampled himself; a bamboo flute, several different kinds of percussion samples, mostly from little home made drums he had made out of post-pak tubes and P.V.C. pipe, and the tuning forks that he made back in '85. In the piece he was using microtonality as a colouristic effect. The tuning forks were always played detuned in quarter tones and doubled, to produce beating and broaden the sounds. Some commercial piano samples and some oboe samples were each doubled and programmed in quarter tones, so they could be played at either normal A440 pitch or a quarter tone higher, or as two pianos or oboes doubled in quarter tones, playing incredibly out of tune. One of the textures, for example, consisted of two pianos and two oboes, each playing their own independent lines, with one of the pianos and one of the oboes playing a quarter tone higher.

In combining sounds and images, Warren says his notion is simply to see how it works, not in the traditional sense, but to see what happens for himself. "When you play a piece of music next to an image, do they combine even if they're not the normal sort of combinational things, or if they don't combine what do we mean by 'don't combine'?"

Occasionally he would try to match the music performance with what was happening in the film. For instance, in the very fast single frame animation sequences between the scenes, he always tried to return to the same musical fragment, which was sort of fast, with lots of drum things and a wistful little accordion melody in it. He actually succeeded only a few times because those sections are very short, about ten or twelve seconds, and he chose to use the mouse rather than the ASCII keyboard.

In fact, in a purely musical situation when playing that particular *M* patch, Warren says he tends to move between fragments very very rapidly, trying to switch to something else as soon as he recognizes what it is that is happening. "I'm always having to fight myself, when I'm playing, into saying 'let it go a little longer'. I'm always trying to send myself into places that I don't know, trying to change my own mind as well as anybody else's. So when things get a bit too comfortable and familiar I tend to change things, but I also have to keep reminding myself that I know it far better than anybody else out there."

SONIA LEBER AND DAVID CHESWORTH, *LET ME CONVINCE YOU*.

Sonia Leber and David Chesworth have helped each other on various personal projects in the past, but they have very rarely worked together on collaborative projects. The M.I.M.A. performance provided them with the opportunity to start from scratch, and because of their close relationship they have managed to avoid many of the formalities and obstacles often encountered in the collaborative process.

Together they discussed the broad concepts which tempered the piece, consulting with each other throughout. Sonia worked out the initial idea, constructed the stories and did the visuals; and David composed and produced all of the sound and music. The sounds and images for the piece were based on recordings of real environments, a train station, the Myer Food Hall, a race meeting and a phone booth, with real people in those situations.

Sonia wanted to shoot small scenes from life and to later superimpose them with short fictional narratives, in an attempt to tell the stories of the various characters lives in three sentences, exploring relationships between documentary and fiction, truths and untruths and the role of the film maker's voice. Together they tried to set up a number of different levels. So there are the real events that you're watching; the voice over coming from the film telling these stories, and that same voice represented by a person, Sonia, on the stage, telling another kind of story--a longer and more personal story which is somehow related to the events on the screen, and which is not resolved with the other shorter stories until the final scene.

"David decided to make his own sound track by using grabs sampled from the actual environment, while Sonia was similarly engaged with her images and plot."

They were looking for a way of making use of the performance aspect, so they set out to produce a piece using pictures which had a certain lack of information, which the music could supply. The idea was that if you watched it without sound you would have no idea of what was going on, or a very different idea, but the sound gave little clues all the time, on one level giving an idea of the setting and the space. In an imprecise way the sound was trying to provide a perspective on how to approach the visuals and the other information.

David was working to try to situate his sound to suit the particular levels as the piece progressed. He was initially interested in the compositional questions raised by what it means to play live music with a film. For example, why not just put the music on the sound track? He wanted to avoid a referential approach where the music is commenting on the visuals in a direct way, even though he has worked that way in the past. He was interested in the contradiction which occurs in cinema, where the audience is placed outside of the visual space, because it's happening through a window formed by the screen, and at the same time the audience is surrounded by, or in the centre of another space formed by sound, because it's in stereo and often uses additional speakers placed well into the auditorium. He wanted to try to move in and out of these different spaces.

This idea was paralleled in the use of voice, where certain stories were coming from the sound track and certain stories were being read live.

Each environment that they visited had it's own living sound track, so David decided to make his own sound track by using grabs sampled from the actual environment, while Sonia was similarly engaged with her images and plot. This approach served also to distance the sound in live performance, helping to reconcile the immediacy of live sound with the depth of the images. David wasn't keen to simply appropriate sound from the environments they chose, some of which actually contained their own background music. In dealing with the challenge posed by the semi-documentary atmosphere of the project to his personal compositional inclinations toward the use of pitch, harmony and melody, David ended up experimenting with techniques that he would not normally have contemplated. By playing around with blocks and wedges of captured sound and incidental music, arranging and composing with them to form a musical and environmental collage, he became interested in the apparent transformation of the 'narrative' music and sounds in the real environment into film music, causing diegetic sound (which can be seen to exist within the screen area, like dialogue) to become extra-diegetic (sounds which occur off screen, like voice over and film music).

David finds the use of samplers fascinating in that he can construct musical environments that always have a more suggestive edge to them, and finds that he has the ability to create sounds with varying degrees of reality or reference to their origins. For example one of the samples used in the piece was of a space which had real things happening in it, scrapes and bangs or whatever, which was lowered in pitch and so changed into something totally different, suggestive of other things, while also maintaining its original character. He doubled all of the samples with a very low organ sound which gave a pitch to every sample because, he says, he's so into pitch that he has to actually put them in.

He prefers to compose music for a particular listening context, where the listening situation itself can be integrated as other layers into the music, and it's in the interactions between these different layers that he finds the greatest interest. In this way he is able to layer visuals, performance, speech as well as singing and instrumentation within an overall context that reaches beyond pure music, which he often finds uncomfortably artificial.

ARF ARF, *WORM WORDS.*

Of the performers included here, the name Arf Arf is probably the least known to the readers of Chroma. Arf Arf is a small group of sound poets, film makers and performers (and occasionally children), varying in number from four to six. The Members include Frank Lovice who, for this article, acted as spokesperson for the group, Marisa Stripe, Rudy Kral, Michael Buckley, Marcus Burgner, Richard Frenken and Ivor Cantrill. They include among their influences the Russian futurist poet V. Khlebnikov, Hausmann, Emmet Williams, and Ditter Rot.

They began by simply making sounds with their voices, looking for things that were appealing, but eventually they felt the necessity to score their pieces in order to try to structure the work they had been practicing. At the same time there was pressure from within the group to also perform improvised pieces, which were included in the sets performed in pubs and other venues. Their scores were extremely simple, based around three generalized pitch areas, low middle and high, and to a large extent the scores have remained in pretty much this form. Later, they began inventing different kinds of scores in order to come up with new material for performance, which have included scores that exist as films, which are projected during the performance of the piece and perform an additional function as visual accompaniment. Another approach has involved the listing of various phonetics after the names of the performers over which each performer quickly draws a squiggly coloured line serving as a graphic orientation.

"They are dealing with the peripheral, forgotten, or supposedly repressed elements in speech and sound. They feel that to produce this effect, using technology would perhaps be impossible, because it would involve removing the sound from its immediate genesis."

Chaos, sometimes quiet, occasionally dominant, is pretty much a part of any Arf Arf performance, with occasional exceptions. Even their most finely rehearsed performances seem to embody the possibility of imminent chaos. Opinions often conflict within the group which are resolved, they say, through democratic processes, although perhaps not always in time for the performance. They try not to separate their functions within the group, so all available members are involved in all processes including the film component. The theme for particular film used throughout the M.I.M.A. performance, of people falling over and shouting at each other (silent), was conceived by Frank, edited by Michael, shot by various members of the group, and performed by the group and others.

On the night of the M.I.M.A. performance they felt stronger on thematic ideas than actual material. They performed several short pieces of around a few minutes each, Often, rather than having fixed titles, the pieces are identified by casual references like "the domestic violence piece" An exception to this general rule was *Worm Word*, written by Marisa. It had been their ambition for some time to perform with Richard's rendition of *Wooden Heart* on piano accordion because, as Frank says, it was the one song he could half play.

Arf Arf generally try to avoid the use of musical instruments because of their tendency to shrink the possibilities of phonetic exploration. They are interested in the musicality of words and story telling, the beauties and terrors that are inherent in language that come out in speech patterns rather than composing sound into definite structures. English is the group's preferred language, however they are occasionally tempted toward Italian, employ their knowledge of Latin, and use the syllables of unknown languages. *Pekodo* was a 'found' piece about sin, scribbled in Portuguese on a piece of paper lying in the street. It has been previously performed in London in a walkway with spectacular acoustics beneath the Thames.

The group sees an advantage in having so many human voices in that it allows them to perform with minimum technology. *Domestic Arguments* produces the effect of breaking language into molecular components through the opposite process of layering voices. They are dealing with the peripheral, forgotten, or supposedly repressed elements in speech and sound. They feel that to produce this effect, using technology would perhaps be impossible, because it would involve removing the sound from its immediate genesis.

Arf Arf are interested in the fact that humans are performing and that they're having emotions, and the thinking process is going on at the very moment of performance. They believe in the value of spontaneous expression, or lack of careful conditioning, which they describe as a throwing out or hurling from the body, and which embodies an attitude toward day to day existence. Their performances are infused with nerves and emotion, the resulting impression of chaos is instantly engaging.

Sound Globs and *M*

- Warren Burt

Sound Globs, Interactive MIDI Composition/Improvisation Program for the IBM-PC and Yamaha C-1 from 12 Tone Systems, PO Box 226, Watertown, MA. 02272 USA - phone: (617) 924-7937–price US$175.

M, Interactive Composition/Improvisation Program for Macintosh, Atari, Amiga and IBM-PC and compatibles, including the Yamaha C-1. Mac, Amiga, Atari versions from Intelligent Music, 116 North Lake Ave. Albany, NY 12206 USA - phone: (518) 434 4110 - price: Amiga US$199; Atari US$200; Mac US$250. IBM-PC version *Mpc* from Voyetra Technologies, 333 Fifth Ave. Pelham, NY 10803 - phone: (914) 738 4500-price US$289

"Like almost every real time composing/performing program, both these programs deal with notions of 'randomness', but do it in significantly different manners."

For the past six months or so I've been working with two 'off-the-shelf' composing/improvising programs and I must report I am totally delighted with both of them. *M* and *Sound Globs* will not be all things to all people, but for those who want to explore possibilities in real-time composing well beyond the range of traditional sequencing, they are well worth investigating. Admittedly, when you work with someone else's composing software, you are also working with someone else's idea of what ways of composing are interesting to them. However, both these programs offer a wide enough range of possibilities to be useful in many ways. And for those who are just beginning to explore this area, or want to write their own composing software, both these programs provide interesting approaches to study.

This review is based on the IBM-PC versions of both programs. *M* is also available for the Mac, Amiga and Atari. The Mac and IBM versions are functionally identical. The Amiga and Atari versions have some differences from the Mac/IBM version, but these are mostly of the 'bells and whistles' type. For more detail, contact Intelligent Music directly. Intelligent Music and 12 Tone Systems do not, at present, have Australian distributors, so you should order directly from them. Voyetra is represented in Australia by AudioLogic of Braeside, Vic., but at the time of this writing they were unsure as to whether they carried *Mpc* or not. In any case, a quick ISD call to the US suppliers would be a good idea.

Like almost every real time composing/performing program, both these programs deal with notions of 'randomness', but do it in significantly different manners. *Sound Globs* is almost totally based around the concept of random selection, and, in fact, allows you to sculpt the kind of randomness you want in some very interesting ways. *M*'s approach is no less based on randomness, but allows you to start off with your own musical patterns, and then permute or combine those in various ways, choosing how much randomness or cyclic order you want. In both programs, there are very well thought out user interfaces which allow you to change almost any aspect of the program while you are hearing the musical output. You can control either program from your own computer, or with any MIDI controller, or with another computer. Both progams also allow you to compose or improvise large scale forms and have the programs follow these. In addition, both programs can store their output as standard MIDI files, so you can use their results with a variety of other software. It would be perfectly possible, for example, to perform with either program, and use it's MIDI file output with a notation program to notate what you've done for performance by live musicians. Further, *M* allows you to import MIDI files and use them either as raw pitch material to be acted on by its various possibilites or to be used as a sequence which plays quite independently of your manipulations of *M*'s own material. This allows a performance using *M* to have characteristics not built into it. For example, I could imagine performing using the very elegant pitch bending and modulation capabilities of *Sound Globs*, making a MIDI file of that, importing that into *M* (which does not allow either pitch bend or modulation wheel control), and performing routines *M* is ideally suited to, while also hearing the kinds of bending and modulation it can't do.

Turning to each program in more detail, *Sound Globs* has a two page user interface. On the first page, called the Edit Page, one specifies base values, called *quanta*, parameter ranges, probability distributions, MIDI channel assignments, and pitch bend and modulation function selections. These are stored as individual textures. On the Performance Page, the other user page, one can combine these textures in real time, send program changes to your synthesizers, type in performance and recording instructions and change any parameter of any texture with the mouse. All in all, a rather flexible system.

The Edit Page allows you to specify base values for pitch quantum (i.e. how many cents each scale step will have, from 1 to 600), time quantum (what is the minimum pulse you are working with - from 5 milliseconds to 5 seconds), pitch anchor (what MIDI pitch number are you reckoning your ranges from), loud anchor (what MIDI velocity value are you starting with), and bend anchor and mod anchor (settings for modulation and pitch bend).

The pitch quantum setting is especially interesting, as it gives *Sound Globs* a microtonal capability not available with any other program I know of. *Sound Globs* gets its microtonal abilities by sending out a pitch number and a bend value with each note for any pitch quantum value other than 100 cents. If you're only interested in 12 tone equal temperament, you can use *Sound Globs* with any synthesizer, but if you want true microtonality, you'll also need to use a synthesizer (like the cheap Yamaha TX81Z) that allows you to have monophonic voices on more than one MIDI channel. You can assign a texture to more than one MIDI channel at a time, having the MIDI channel assignment change sequentially with each new pitch. Then, with the TX81Z set to have 8 monophonic channels - voila! – real time microtonal polyphonic music! This feature alone makes it worth the price.

Next one sets parameter ranges. Having defined a base pitch, for example, one then specifies how many pitches above and below this base value will define the pitch range. Horizontal density corresponds to times between events – what is the longest and shortest time that will occur between events in millisecond units given by the time quantum. Duration sets the note length, the time between MIDI note-on and note-off events for each note. This can be longer or shorter than the horizontal density, resulting in a nice mix of staccato and overlapping events. However, there is no way of slaving duration to horizontal density, so if you want a melody that is always legato with changing durations, you'll have to figure out a way for your synthesizer to do it with long envelopes or some such. This, to my mind, is the only flaw in an otherwise wonderful system. Vertical density specifies how many notes will happen on each attack, from 1 to n, and loudness sets the ranges for velocities.

The values for all these ranges are then chosen randomly. How are they chosen randomly? By referring to a probability distribution graph you set for each parameter. You draw a graph specifying, for example, 9 times as many middle Cs as high E flats and no Ds at all, and another that says there will be 8 times as many staccato notes as legatos, and another saying that 80% of the events will be single notes, with 10% being dyads and 10% being chords of 7 or 8 notes. Further, you can apply the graph of any parameter to any other parameter. Total serialists, take note. You can have up to 24 different textures at once, and each of these can have different probability distributions, different timbre assignments, different kinds of pitch selections, different tunings, different ranges, etc. And you can combine these in real time, sculpting the sound to your taste. For those with an interest in randomness, stochastics and microtonality, this is one wonderful program.

M's approach is similar, but different. Where *Sound Globs* is an elegant realization of one (very diverse) interest, *M* provides a more general purpose programming environment with many ways of doing things. *M*'s user interface consists of a single well organized page. All other features appear on pop-up menus. With *M*, one first records a pattern of pitches into memory. The program offers a number of ways to do this from single note or chordal playing on a keyboard, to drawing pitch patterns with a mouse, to importing MIDI files to act as these patterns. This pitch memory is not a traditional sequencer, as it does not record durations, but is a repository of pitch patterns and harmonic resources for you to work with.

To this raw pattern of pitches you then apply cyclic patterns of durations, accents, and legato-staccato values, which can also have random selection of elements in them. You then select voice selection (is a given voice playing or not?), velocity ranges (what is the loudest and softest a sound will get), note density, (100% of notes playing, or less?), note order (will you hear the notes in the order you specified? or in a different order made up by the computer? or totally randomized? or in a mix of these three? and what percentage of mix would you like, sir or madam?), transposition, and time distortion values (a unique and wonderful feature that allows you to give rhythmic life to your music, setting up a range of patterns from simplistic jazz swing feels to more complex Nancarrow-like patterns of acceleration and deceleration), MIDI channel assignments, and program change values. For each of *M*'s four polyphonic voices, you can have up to six completely different sets of these cyclic patterns and variable selections, all selectable in real time. Then you can save any selection of these values into a 'snapshot' which you can instantly select (you can have up to 26 snapshots). You can save any order of 'snapshots' into one of 9 'slideshows', and save performances into a 'movie', which you can convert to a MIDI file. Further, the tempo, time signature, rhythmic phase of the voices, how you are controlling *M*, (with your computer, or with a MIDI controller, or with an external MIDI sync) are all controllable in real time. An external MIDI controller can also be used in a variety of interesting ways, from merely playing accompaniments with a MIDI thru assignment, to being used to transpose pitches, to being used to fulfill all the functions of the mouse on the screen, to being used in what is called 'step-advance', where you can control the rhythm and articulation of your performance with the controller, but let *M* select the pitches. You can also use a 'conducting grid' to select user-specified sets of the cyclic patterns and random variables in real time, or even specify how you would like *M* itself to make decisions of tempo, timbre, and composing logic. It's a delightfully rich composing and performing environment, one that offers lots of potential for many different kinds of music.

Sound Globs works on any IBM compatible with 640K of RAM; uses a Roland MPU-401 or compatible interface; Hercules, EGA, CGA, or VGA graphics; and a mouse. It provided satisfactory speed and response even on my old steam-powered XT clone, though a faster machine has made its operation, especially at the extremely fast end of things, just that little bit cleaner. Voyetra claims *Mpc* will work on any IBM compatible, but because it operates using the MS-Windows graphic system (a version of which is supplied with *Mpc*), it really does require a machine of AT speed or higher. On my old XT, it could take as long as 15 seconds for the screen to update after every change of every parameter. Once I sped it up to AT speed, however, it worked as fast as any Mac version I've seen. It also requires DOS 3.0 or higher; 640K of RAM; the Roland MPU-401 or compatible MIDI interface; and you really do need a hard disk to use it properly. So if you're poor (like me), *Sound Globs* will be the program to start off with, but as soon as you can save up enough Weet Bix cards to upgrade your system, get *M* as well. For those of you who worship at the churches of Amiga, Atari, or Mac, there's only *M*, so what are you waiting for?

Finale Tips

- Roger Glanville-Hicks

If professional quality music printing suitable for publication is your aim, then Coda Music Software's program *Finale* is going to be on your list of contenders for the job. Despite its unwieldy size and frustrating documentation, (version 2.0, just released, apparently addresses this problem. The version used in this article is 1.2.6), it does have many good points not the least being the finished product looks great. This for the most part, is due to the program's capability to do nonlinear spacing, and a set of elegant music fonts.

Finale can be a daunting experience to the uninitiated but with a little perseverance you can learn to navigate the software quite simply. The purpose of this article is to show an easy method for painless instant gratification. This involves a two step process – transferring your file from a sequencer to *Finale Power Plus*, and thence to *Finale*.

Let's say you have some MIDI sequencer files that you want to put into notation, or alternatively, you have a piece in manuscript and a MIDI sequencer you are familiar with that supports the MIDI file format. If the latter is the case you will first have to play your piece into the sequencer, then, in either case, quantize it making sure that the durations, as well as the attacks, are pulled into line. Of course, if you step record no quantization will be necessary. If you have a sequencer that doesn't support the MIDI file format then find someone who has one and just dump the file across (record it in real time using MIDI sync), then save as a MIDI file. *Professional Composer* files can be opened up into *Performer* then saved as a MIDI file, *Deluxe Music* files can be converted via Opcode Systems' *Sequencer* and or *Vision* programs.

It's a very good idea to take extra care during the quantizing operation as this will save a lot of editing in *Finale*. Triplets, quadruplets, quintuplets and other such groupings of notes will all come across intact as long as they are quantized properly. Similarly, if your music contains multiple meter and key changes make sure you enter these (if they're not already) into the conductor track of your sequence.

Take a note of the number of tracks, remembering the order in which they occur, the rhythmic complexities in the sequence and the MIDI channels used. You can simplify matters further by setting all tracks to MIDI channel 1, then save as a MIDI file and choose 'Type 1' - separate tracks.

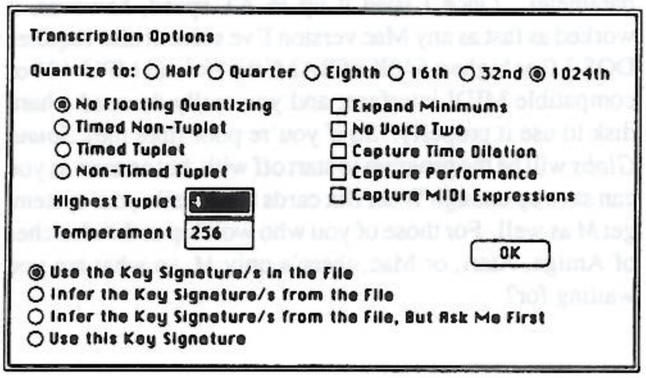


Figure 1

Open up *Finale Power Plus*, the smaller program in the Coda package. Go to the File menu and select **Transcribe MIDI Sequencer File** then locate and open your MIDI file. The dialog box shown in Fig. 1 will appear.

The **Quantize to:** radio buttons at the top of the screen default to a 1024th which means there'll be no quantizing. Leave it this way.

Floating Quantizing is an option that was added to Power Plus in the last update of the program and is not covered in the manual except in conjunction with the Transcription Tool in the main program. It is this feature that enables accurate transcription of more complex groupings of notes. If your file is rhythmically straightforward leave the default setting of No Floating Quantizing. Otherwise try one of the other radio buttons.

The **Non-Timed Tuplet** option generally produces the best results as long as you set the Highest Tuplet and Temperament numbers correctly. Just remember to keep the **Highest Tuplet** figure higher or as high as the largest grouping. i.e. If your file contains groups of triplets, quintuplets and septuplets, set the number to seven. The Temperament figure as a rule is best kept at 128 or 64. You could also try 32 and 16 etc.

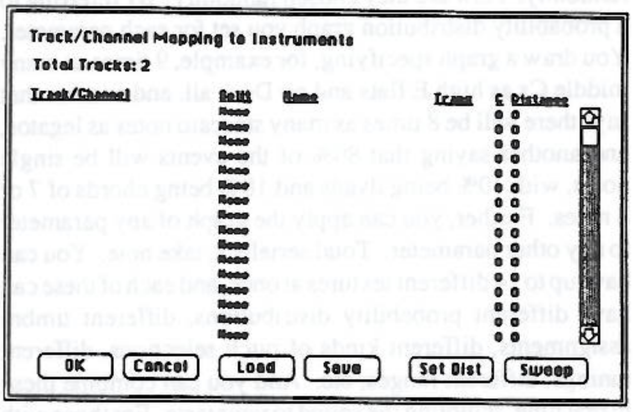


Figure 2

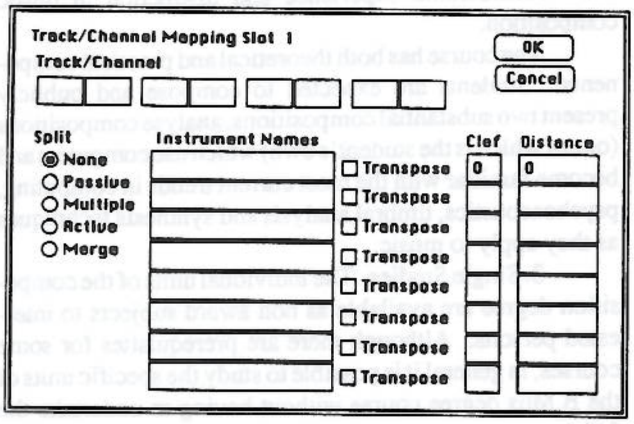
The **Key Signature** radio buttons are quite clear. If you have set keys in your sequencer file leave the default setting and the same keys will be present in the score. Otherwise you may not want a key signature in which case select **Use this Key Signature** and OK the default C major.

Of the remaining check boxes it's a good idea to check **Expand Minimums** if you have skimped on your quantizing of durations in your sequencer and consequently have decided to employ one of the quantizing options at the top of the screen. In this case you will avoid any notes shorter than the quantization level being transcribed as grace notes, a situation that can be very annoying.

No Voice Two affects the organization of individual parts transcribed onto a single stave. If, for example, your file was a Bach chorale and you had played it into your sequencer one voice at a time with each voice on a separate track, then leaving this box unchecked, as is the default, will result (if this is what you're after - see 'mapping' later in this article) with two separate voices per stave. On the other hand if you check it you will get a chordal style. i.e. two notes on a single stem.

The remaining three check boxes relate to MIDI information that we are not concerned with here. Leave them unchecked unless you particularly want Finale to play back your MIDI controller information and tempo changes. Clicking OK brings up the dialog box shown in Fig.2.

No information is actually typed directly into this dialog box, it only serves to give an overview of your final set-up. By clicking anywhere on any line another dialog box appears. It is important to understand that what you are doing here and in the next dialog box is 'mapping' the tracks of your sequence to staves in the score. In *Finale* terminology an Instrument correlates to a stave. Click on the first line and the following dialog box appears :



To keep things simple, let's say your file has four tracks, each with a single monophonic voice. Type two numbers into the first pair of Track/Channel fields. The first is the track number, the second the MIDI channel number of the track. Now set the clef. Numbers 0 - 7 determine which clef. (0=treble, 3=bass). Click OK to return to the previous dialog box where you'll see your information entered in the first column.

Repeat this procedure for the other three tracks, each time clicking on a new line to enter a new mapping slot and typing the track number into the first field and the MIDI channel number in the second.

When you're done click on the Set Dist button at the bottom of the screen and OK the default settings. Finally click OK and your file will be transcribed.

When it's finished choose **Enigma** as the file format. The other format- clip file, could be useful if you had one or two particularly troublesome tracks that you work on independently, then import into your score later saving transcription time.

This procedure has produced an open score format. Had you wanted to merge two tracks onto one stave and keep independent voices take the following steps :

• Repeat as above leaving No Voice Two unchecked.

Track/Channel Mapping Slot 1

Track/Channel

1	1	2	1				

• In the Track/Channel Mapping Slot 1 enter the first two tracks of your sequence into the first two consecutive pairs of fields.

Track/Channel Mapping Slot 2

Track/Channel

3	1	4	1				

• In the Track/Channel Mapping Slot 2, enter the second two tracks of your sequence into the first two consecutive pairs of fields.

• The rest is the same as above.

Finale is technically only capable of having two independent voices per stave, but if you want more you just take two or more staves and layer them on top of one another, making sure first that you freeze stems in the right directions. Do this with the Mass Mover Tool.

Finally go to the main *Finale* program and open up your file which is now notated. The first thing is to get the spacing looking better so you can see what needs fixing.

• Load in one of the sample spacing allotments libraries using the **Load Library** command in the File Menu.

• Select the Mass Mover Tool - 

• Choose Select All from the Edit Menu

• Hold down either number 3 or 4 and double-click the first measure.

Things should be looking much improved now. Save your file. Unfortunately most actions that you would like to undo in *Finale* are not undoable so you just have to save before doing anything you're not sure of, and re-open the file if it doesn't work out.

Editing is easiest using the Speedy Note Entry Tool,

but before you proceed it's a good idea to turn off a couple of the automated default settings that can get in the way. Go to the Special Menu and select Speedy Entry Options. Uncheck Enable MIDI Keyboard and Clip To Measure. Checkout the manual for details on the Speedy Note Entry Tool.

Working this way in *Finale* can save a lot of time and in most instances it works very well. You can also get a lot more complicated in terms of merging and mapping tracks to various staves. There are a few other alternatives, like the Transcription Tool, HyperScribe Tool or dumping your sequence directly into *Power Plus*. The Transcription Tool is the most powerful, but takes some time to learn. If you've already spent a lot of time learning your sequencer and developed fast editing with it then you might as well use it instead.

Finale is such a large program that you tend to develop your own methods for working and then never use half of the available tools at all.

Happy Transcription.

Computer Music Studies at Canberra Institute of the Arts

Background

The Canberra Institute of the Arts (CITA) has one of the most extensive electroacoustic music facilities in the country and is a vibrant centre for the research, development and performance of computer music.

The Institute's commitment to this genre is evidenced by its recent announcement of the formation of a new Australian Centre for the Arts and Technology (ACAT); a centre for the teaching, research, performance, recording and publishing of music and graphic art made with new technology. This centre, the first of its type in Australia and linked to similar centres around the world, will provide a focus for the education of tomorrow's multidiciplinary artists and composers.

Over the last 30 years the fields of electronic and computer music, and computer graphics and animation, have become disciplines which have more in common with each other than they do with their individual origins. This is due in no small part to the fact that they both have become computerised. With the advent of digital computers, musicians and visual artists are, for the first time, using the same tools - tools for which mental dexterity has to a large part superseded manual dexterity. Currently, the wide range of applications for sound and image technologies is challenging traditional modes of communication and education, and has created many new and rapidly growing industries.

As a natural partner in research and development of software and hardware, in the training of individuals for the industry and in the beta testing of new products, ACAT is actively seeking cooperation with industry. The Centre's public profile in its research and touring performance role also makes it ideal for companies seeking to enhance their public image.

CITA is home to the electrospatial art group *floating exceptions* who have designed and built an experimental space - a portable geodesic dome - for research, and the performance of compositions in multichannel sound and light under computer control.

Courses

Courses involving music technology are being constantly revised and new courses initiated. Presently someone wishing to study computer music can do so at the Canberra Institute of The Arts' School of Music by undertaking:

1. a Bachelor of Music degree in composition,
2. a Graduate Diploma in Computer Music, or
3. Single studies subjects taken from the degree courses and eventually leading to a Diploma of Music Technology.

1. The Bachelor of Music (B. Mus) with a major in composition is a four year, full time degree course with an emphasis on the composition of original music. There is equal weighting given to the development of instrumental and electroacoustic techniques. The degree is not a purely theoretical one; instrumental lessons and a core of practical studies ensure a balanced musical training.

2. The Graduate Diploma in Computer Music is an advanced course in computer music composition for the composer who has already achieved a high level of technical and musical skill. This course, the first of its kind in Australia, requires one year of full-time study. Applications will be received at any time and the year of study commences at enrollment.

In determining an applicant's suitability for entry to the course, account is taken of the nature and level of previous studies and/or professional experience. A considerable compositional technique is assumed. As a guide, applicants are considered from those who either hold a degree, diploma or equivalent qualification in music, or can show considerable experience and distinction in music composition.

The course has both theoretical and practical components. Students are expected to compose and publicly present two substantial compositions, analyse compositions (one of which is the student's own) which use computers and become familiar with the most current trends in computing, psychoacoustics, timbral analysis and synthesis techniques as they apply to music.

3. Single Studies. The individual units of the composition degree are available as non award subjects to interested persons. Although there are prerequisites for some courses, in general it is possible to study the specific units of the B.Mus degree course without having to undertake the full degree programme. Prospective students must be literate in music.

Topics covered in these courses include: acoustics, algorithmic composition, electricity and electronics, computing, MIDI, psychoacoustics, sound sampling and synthesis.

New Courses presently being written are a Master of Music (M.Mus) degree in composition, a Bachelor of Music Technology, a Diploma of Music Technology, and in collaboration with the Canberra School of Art, a Master of Arts (in Graphics and Technology), Bachelor of Arts Technology and a Diploma of Arts Technology. These new courses will become part of the academic programme of the Centre for the Arts and Technology.

David Worrall is the director of the Electroacoustic Music Studios and Acting Head of ACAT.

Resources and Facilities

The Institute's School of Music has two electroacoustic music studios, a music library of over 30,000 books, journals, scores and records, and one of Australia's finest concert halls.

Audio facilities at ACAT are undergoing a major upgrade, and computer graphics/animation facilities are being established. However, at present, the audio facilities consist of: two quadraphonic electro-acoustic music studios, and a transportable performance equipment for live performance, including: 1x24, 1x16, 1x12, 2x6 channel mixing consoles; numerous tape recorders, cassette recorders & PCM; numerous microphones; Fairlight 2X & Akai S900 samplers; various FM & NLD digital synthesizers;

Serge, EMS, Moog analog modules; Macintosh computer network, various microcomputers, Maestro supercomputers; portable performance space for multichannel playback and computer graphics animation.

Further information can be obtained by writing to:

Student Administration
Canberra Institute of The Arts
GPO Box 804
Canberra City, 2601 Ph. (062) 5708
or by phoning David Worrall on (062) 49 5754

Graduate Diploma in Contemporary Music Technology at LaTrobe University Music Department

The Graduate Diploma in Contemporary Music Technology at La Trobe University focuses on the rapid technicological developments in contemporary music design and production. It is one of only a small number of such programs in Australia. Founded in 1989 with the help of a major grant from the Victorian Education Foundation, it combines access to first rate professional facilities with a comprehensive curriculum and teaching by internationally recognised staff.

The program is one year full-time or 2 years part-time, and consists of coursework and major project submissions. Coursework components divide into the following three areas:

• **MIDI Systems:**
synthesis, sampling, sequencing, composition, score production, performance skills, etc.

• **Audio Recording:**
multitrack studio and location recording techniques

• **Computers and Music:**
microcomputer programming, and direct digital signal processing and synthesis

Coursework consists of two 'major' components and one 'minor' component, chosen from the three coursework areas. Major components continue for a full academic year, while the minor component runs for only the first semester. The program assumes a solid grounding in the basics of each of these areas has been achieved, and aims to produce graduates who will exercise a leadership role in contemporary musical developments.

The student formally begins project work in the second semester on his or her chosen topic of specialisation, under the guidance of a nominated supervisor. Students who can demonstrate reliable competence in particular coursework components may be allowed to commence their projects earlier than normally scheduled. Major project work culminates in the submission of a project folio which includes audio, score and possibly video work and computer software in various media formats, and a supporting dissertation. The final deadline for folio submission is February 1; students are able to work over the summer and break periods, when the labs are less heavily booked by other users.

Work in the projects may include, but is not limited to, the following topics: composition (all styles), MIDI synthesis and systems, direct digital synthesis, synthesizer performance techniques, synthesizer ensemble, musical acoustics, FM theory, filtering, nonlinear and modelling methods, sound treatment, sampling, intelligent instrument design, microcomputer score production, music software development, algorithmic music production, multitrack recording, digital sound recording, video and SMPTE, film music.

Expected Vocational Outcomes

The diploma awarded is Grad. Dip. (Mus. Tech.). Diplomates will be qualified to pursue a number of different professional areas of specialisation. These correspond to such positions as sound engineer, film composer, band performer/composer, arranger, radio producer, audio consultant, music software designer, music educator (public or private), synthesiser programmer, MIDI systems specialist, synthesist, video producer, independent music researcher. Rapid change in music technology means that some such positions did not even exist five years ago, and that there are others, now undreamt-of, that are likely to be significant in five years time.

Lecturing Staff: David Hirst, Jeff Pressing, Jim Sosnin.
Technical Staff: Tony Falla, Chris Lai.
Facilities: Two MIDI Composition Laboratories.
Synthesis Research Laboratory.
Extensive Macintosh and Mini-computers.
16 track recording facility with 2 studios.
Linotrnic Score and DTP Production Facility.

Equipment and software available includes top-end technology from Apple, Yamaha, Roland, Tascam, Akai, Alessis, Oberheim, Fairlight, Passport, Opcode, Digidesign, Kurzweil, Mark of the Unicorn and other companies.

Requirements for Admission

Applicants should have a previous Bachelor's degree in music; a solid background in composition and basic keyboard performance skills are strongly recommended. Applicants should also be able to demonstrate solid basic skills or experience in each of the three coursework component areas. A limited number of applicants with distinguished musical credentials, but without a previous degree, may also be given places.

Application

Applications closed on Friday 8 December, 1989, but late applications may be considered, subject to the availability of places. Application forms are available from:

Raelene Dalzell
Department of Music
La Trobe University
Bundoora, Victoria
Australia 3083.
(03) 479 2879.

Australian Computer Music Association, Inc.

1990 Membership Application

ACMA was formed in June 1989, with the intention of providing a means for sharing information on a range of areas of music and technology in Australia, including:

music / sound synthesis and processing
MIDI
music notation
commercially available software and hardware products
algorithmic composition and strategies
Amiga, Atari, IBM, Macintosh, main frames etc.

The annual membership fee is $10 (individuals, $100 institutions/organisations). Please complete the details below and forward to:

The Secretary
Australian Computer Music Association, Inc.
P.O. Box 4136
Melbourne University 3052

Cheques should be made out to the Australian Computer Music Association, Inc. Membership entitles you to receive and contribute to Chroma, the Association's quarterly newsletter and participation in all ACMA sponsored events.

Name:

Address:

Telephone: (   )

Organization:

Particular areas of interest / equipment / software used:

What computer do you use?

Do you agree to allow your name, telephone number and interests to be circulated to other members of the Association?

☐ Yes ☐ No
☐ Tick here if this is a Renewal

Signature: Date:

Member No:
Receipt Date: