

---

---

## EDITORIAL

Welcome to issue 24 of **Chroma**, the occasional newsletter of the Australasian Computer Music Association. You are receiving this newsletter because you have either been a member of ACMA in the past, are currently a member, or have expressed an interest in computer music to someone on the board. This is ACMA's 10<sup>th</sup> anniversary. In that ten years, it has produced a number of concerts, 24 issues of this newsletter, 6 conferences, and 2 CDs. This year's conference, in Wellington (see the article by John Young, the conference organiser, in this issue), promises to be an exciting event. In this issue are articles by Gordon Monro, Garth Paine, and a rather large survey of newly available music software for Windows by yours truly.

We are looking for members! An organisation such as ACMA only exists as long as its members are active! We need people to write Chroma, edit it; to organise concerts, do publicity, etc. ACMA, and computer music in general, is in a stage of transition. Once an activity that took place mainly in the academic environment, computer music has become democratised and now happens in many styles, venues, and genres. If there ever was any stylistic consistency among those who used the computer to make music, that day is long gone. In this new de-institutionalised environment, the role of computer music (in fact, the role of all the arts!) is very much up for grabs. ACMA can be whatever its members want it to be, as long as they are active within it. If you like what you see in this issue (or if you don't and want to change it!), consider joining or renewing your membership in ACMA. Membership forms, and state representatives are listed at the end of this newsletter.

-Warren Burt, guest editor.

---

## ACMA MELBOURNE CHAPTER MEETINGS

David Hirst and Warren Burt met in late January over pizza at Papa Gino's in Carlton to discuss the future of ACMA in Melbourne. They decided that what was really needed was a time when Melbourne members could come together and discuss matters of interest. Warren mentioned that he was also a part of the AXLE group of visual, sound, and action poets, and that he quite enjoyed their monthly

get-togethers. So they decided that a monthly gathering time for ACMA Melbourne members could be enjoyable. Warren offered his place as a venue for starters. They decided that the THIRD THURSDAY of every month would be a good time. So:

**ACMA MELBOURNE CHAPTER  
MEETING:  
THURSDAY, 18 MARCH  
7:30 PM  
at WARREN BURT'S place:  
6/17 BROADWAY, ELWOOD  
9531-2059**

Bring ideas, things to share, etc. We'll probably send out for the pizzas.

---

## ACMA 1999 CONFERENCE - THE IMAGINARY SPACE - WELLINGTON, JULY 7-10

**John Young**

John.Young@vuw.ac.nz

The 1999 Australasian Computer Music Conference will be held 7 - 10 July at the School of Music, Victoria University of Wellington. This year's conference will feature a festival of 9 concerts supported by 5 papers sessions. With music chosen from 140 submissions, the 1999 ACMC promises to be a memorable event. Music in all categories of acousmatic, live/interactive, instrument and tape/computer, installation and video were received. The concerts will be held in the School of Music's Adam Concert Room - a 180-seat concert chamber perfectly suited to electroacoustic music - with a 16-channel sound system. Additionally, Concert FM of Radio New Zealand will be broadcasting two featured programmes of music from the conference PRIOR to the conference itself!

English composer Jonty Harrison will be keynote speaker for the conference. The 'playful volatility' of Harrison's music is widely appreciated, and the conference will also benefit from his many years experience as director of the world's pre-eminent electroacoustic music diffusion systems - the Birmingham Electroacoustic Sound Theatre (BEAST). Harrison's visit is sponsored by the British Council and the University of Birmingham.

All conference events will be open to the public. Further information available at: [www.vuw.ac.nz/music/studios/acmc.html](http://www.vuw.ac.nz/music/studios/acmc.html)

---

---

## Report on the 1999 ANAT Summer School

**Gordon Monro**

G.Monro@maths.usyd.edu.au

I was a participant in the 1999 ANAT (Australian Network for Art and Technology) National Summer School, held this year in Sydney from 11<sup>th</sup> to 29<sup>th</sup> January. The Summer School is intended to give practising artists an "immersive experience" in computer media, with tutors who are themselves artists in digital media. The Summer School is heavily subsidised by participating organisations and through State and Federal grants. Entry is competitive: applicants submit a portfolio.

This year's Summer School had an additional theme of "Science and Art": we had "artist talks" from artists who are engaging in some way with science, and several of the Summer School participants were also working in this area.

There were 14 participants: a very diverse lot. About six of us had a primary interest in music/sound, either from a "classical" music background or through the sound art/sound installation scene. We also had a film maker, a sculptor, a poet, several Web artists, an installation maker... The most extraordinary art practice among the participants was that of Yonat Zurr, who uses tissue culture to create artworks. It turned out that I already knew three other participants through the electronic music scene: Rodney Berry, Lea Collins (from ACAT) and Solange Kershaw. A lot of the people involved seemed to be pretty well-known in their respective fields.

The Summer School was hosted at Metro Screen in Paddington. We each had a Macintosh computer loaded with software, and we were given 24-hour access; Brad Miller from Metro put in a lot of time making it all happen. The instruction focused on multimedia. We were exposed (at very high speed) to image editing (PhotoShop), video editing (Premiere), Web page construction (DreamWeaver), sound editing (ProTools), Java, VRML (Virtual Reality Modeling Language), and more. There were also a couple of sessions at Sydney Vislab (The Sydney Regional Scientific Visualisation Laboratory, run by a consortium of universities in NSW), where we got a look at the big visualisation package Houdini. "Immersive" was certainly the word!

Vision was somewhat privileged over sound. For example, there was a video projector but no decent sound system. It was also noticeable that our tutors were generally more knowledgeable about images, and would be fussy about things like (visual) anti-aliasing in the same way that sound people are fussy about sound quality.

The "artist talks" were fascinating. Among those who talked to us were Justine Cooper, who managed to get a complete 3D NMR scan of her body (not just the surface, but all the interior bits too) and made a video which is a fly-thought of herself; Horst Kiechle, who makes extraordinary faceted geometric constructions out of Visyboard, some of which are created algorithmically; Paul Brown, who uses cellular automata to make 2D artworks, and talked to us about artificial life and genetic algorithms; Dennis Wilcox, who demonstrated a machine involving mirrors, a rotating shutter, a video projector and a computer generating carefully adjusted images, the idea of which was to see "all sides of an object at once", something like a four-dimensional view of a three-dimensional object. The "lead tutor", John Tonkin, also showed us some of his algorithmically constructed animations. The one that really struck me was simply sheets of falling paper (modelled by springs and masses). They fell beautifully.

We also had a couple of "studio visits". We went to the large shed which is the home of Triclops, a group which does strange technically-related things. The star turn was a one-person hovercraft, steered by moving one's body weight, which we all had a go on. They have also acquired a computer-controlled earthquake simulator (which they have assembled and are learning how to program), and have created pyrotechnic displays involving clouds of droplets of molten iron. I also finally got to see Rod Berry's "Feeeping Creatures", an interactive computer artwork where animated cubes scurry about. The cubes emit music which is actually their "genetic material".

During all this there was quite a lot of talk about science and about the relationship between art and science. In my view some of the actual science was dodgy, but there was much food for thought.

What did I get out of it? I was happy with the focus on multimedia, as this is an area I wanted to learn about. I picked up useful information

---

---

about things like sound and video formats for the Web. I decided to concentrate on learning Java, and wrote an applet which genetically "breeds" images. I also did a little work with VRML.

More generally, I want to do further work in algorithmic processes and in the relationship between sound and images. I got some useful criticism, both artistic and technical, of my recent attempt at an abstract video (which was in the 1998 WATT concert). There was an enormous amount to digest, and I am sure the experience will influence my work in ways I can't yet see.

On top of that I met lots of interesting people, and it was all great fun (as well as totally exhausting). Towards the end of the Summer School there was an event where ANAT launched its 1999 programme, and we got to display our "work in progress", as well as hear talks by Stelarc, among others. After that we had an invitation to go to East Side Radio (89.7FM) for the program Hydrogen Jukebox, run by Zina Kaye (who also runs a Web radio site with Mr Snow, one of the School's tutors). Zina played some music composed by members of the Summer School, and meanwhile more and more Summer School people kept on turning up; an on-air party developed, which continued until the program finished at midnight.

I encourage other electronic musicians to apply for future ANAT Summer Schools. The ANAT people (in particular Amanda Crowley) put a huge amount of work into organising a complex program of great interest, and if future Summer Schools are like this one, they will be well worth participating in!

---

## **CEC DISCUSS@CONCORDIA.CA**

**Kevin Austin**

kaustin@vax2.concordia.ca

CECDISCUSS is an international email list for the electroacoustic / computer music community. It has over 350 members in 17+ countries. Topics range from the aesthetic to the technical, the historical to the philosophical and beyond. To subscribe to CECDISCUSS, send the following message:

subscribe cecdiscuss  
to:  
majordomo@concordia.ca

## **AUDIOBOX 1616HD**

**Garth Paine**

garth@creativeaccess.com.au

In the last quarter of 1998 my company Activated Space Pty Ltd installed a Richmond Sound Design AudioBox into the Immigration Museum (IMHAM), Flinders Street, Melbourne. This is the first AudioBox to be installed in Australia. As it is a piece of equipment that I am sure will be of interest and use to the electroacoustic community, I am providing here a brief review and a number of URL references.

The AudioBox DM1616HD is a 16 input, 16-output matrix mixer. It has programmable equalisation and delays on all inputs and outputs and has 8 channels of hard disk playback, which are fed into the input channels 9 to 16. Input channels 1 to 8 consist of eight synchronous channels of 20 bit analogue to digital conversion fed by female XLR at 15K ohms, accepting a signal as hot as +20 dbu. The outputs consist of 16 synchronous channels of 20 bit digital to analogue conversion fed to 16 male XLRs at 600 ohms, again running as hot as +20 dbu.

All inputs and outputs are provided with high audio quality electronically balanced line receivers and drivers. The AD, DA converters are 20 bit linear, fourth order delta-sigma, with a sampling frequency of 48 kHz. The stated frequency response is 20 Hz to 20 kHz plus or minus 0.1 db with a stated dynamic range of 102 db, unweighted across the above noted frequency response spectrum.

The programmable equalisation allows up to 12 bands of true parametric equalisation to be assigned to any input and any output with a centre frequency anywhere between 20 Hz and 20 kHz, a Q from 0.5 to 50, and a 20db cut or boost. There is processing power to accommodate a total of 96 bands of EQ across the entire box. The delays are programmable from 0 to 5.2 seconds in one one-hundredth of a frame increments, however the total delay assignable across the entire box at any one time is 5.2 seconds.

The AudioBox is equipped with control protocols for midi-in, midi-out, midi-through and SCSI.

You will see from the above specifications that the AudioBox is capable of delivering very sophisticated control of multi-channel sound

---

---

dispersion. It is possible, using the 256 matrix cross-points to programme the movement of audio on a single input, across the 16 outputs over time with varying gain structures.

In my use of the AudioBox at the Immigration Museum (IMHAM), I tried to establish unity gain setting for all inputs and outputs, approaching the setting of levels using the cross-points. The cross-points are individually programmable and may be faded over time, which, depending on speaker placements, causes an illusion of the physical movement of sound source in the time domain. Use of programmable equalisation in synchronisation with the changing of gain structures will augment the sense of perspective.

The AudioBox is programmable using Midi Show Control. Midi Show Control developed, by Charlie Richmond, is a protocol extension, to the standard midi set.

Midi Show Control addresses sounds on the hard disk as **clips**. The **clips** are programmed into a **path** which is itself a part of a **list**. It is possible to have up to two lists open at any one time with any number of paths open and active within those two lists. It is therefore possible to have a great number of audio elements playing simultaneously with control only limited by the number of inputs and outputs available.

The show control cues are stored on the internal hard-disk providing virtually unlimited non-volatile show programming capacity. A show is the overall structure containing the programmed lists, paths and clips. A number of shows can exist on the AudioBox simultaneously, however, only one show can play at a time. A selected show is assigned as the default to be loaded when the AudioBox is turned on. Any show can be called up at any time using Midi Show Control commands.

The AudioBox can be programmed from a PC or Macintosh. The PC software ABedit (supplied with the box for free) presents a less user-friendly interface requiring the programmer to open parameter windows for each control input. The Macintosh software ABControl provides all parameters on a single interface (much like a mixing desk) with an added Cue list window with columns for each kind of control data. Clicking on the appropriate column opens the data edit window associated with that parameter. It is possible to draw speaker positions and dispersion angles

within the Macintosh software, a function not available within the PC software.

Each of the gain structures established, as a cue within the AudioBox is stored as a preset. The programming therefore requires simply the recall of a gain preset number. A preset may be allocated a submaster control so that all gain structures with that preset can be increased or attenuated through the movement of the allocated submaster.

The programming and the dumping of audio files into the Audio Box, takes place over SCSI. Audio is prepared as mono AIFF files and is transferred using a software utility from either a PC or a Macintosh. The AIFF files must be at 48 kHz.

## GAIN CONTROL

The AudioBox provides two forms of automated gain ramping, sub-mastering capability, muting and soloing, and downloadable gain curve tables. The two forms of gain ramps are **exponential** and **table driven** and may be selected as part of the programmable gain command. The duration of the gain ramp is adjustable from a few milliseconds up to one hour. The gain changes are always ramped at the DSP level on a per sample basis ensuring that the gain changes are artefact free.

Up to 32 submasters are available for live input and output level control. Each sub master can be independently assigned up to 32 input and output points and each control point can have any number of submasters assigned to it. All of the programming functions of the AudioBox are driven by 4 synchronised DSP chips configured in a shared memory architecture, coordinated by a separate micro-controller running a real time multi-tasking operating system.

The programming command set for the AudioBox is contained in an on board EPROM. This means that firmware upgrades are possible from a host computer. The MIDI Show Control system is a little bit cryptic. For those of you unfamiliar with it, I recommend John Huntington's **Control Systems for Live Entertainment**, (Focal Press 1994).

I am sure when you have examined the URLs provided you will agree that the AudioBox is a very sophisticated and useful piece of equipment for electroacoustic music.

---

---

Some of you may have noticed that Darren Copeland has organised Sound Travels, a touring concert of Canadian electroacoustic music in which he uses the AudioBox as the primary audio playback and sound dispersion system. He is using the Macintosh ABCControl software, which allows real time changes of gain structures, EQ delay and dispersion assignments.

It is my hope that such a system would be available in a public (possibly outdoor) performance space in Melbourne within the next couple of years. It is certainly a project into which I have already put a good deal of energy and for which I continue to seek opportunities.

As stated at the beginning of this brief article, this is really only an overview. The intention is to alert those of you interested in sound dispersion to a useful tool, and provide some URL's for further research.

It is worth noting that the AudioBox installed at IMHAM has been running from 8:00AM to 9:00PM daily since October 1998 without failure of any kind. The AudioBox is part of a much larger installation under the control of an Alcorn McBride V4+ show controller. You can find a system schematic on my site at [www.creativeaccess.com.au/~garth](http://www.creativeaccess.com.au/~garth).

The AudioBox is available from Peter Jago at System Sound in Melbourne.

A schematic of the matrix structure of The AudioBox is shown on the last page of this newsletter.

Other references:

Richmond Sound Design  
<http://www.show-control.com/abspecs.html>

Third Monk Software  
[www3.bc.sympatico.ca/hfi/dm16.html](http://www3.bc.sympatico.ca/hfi/dm16.html)

[www3.bc.sympatico.ca/hfi/ABCControl.html](http://www3.bc.sympatico.ca/hfi/ABCControl.html)

---

## **SOUND TRAVELS TO NEW FRONTIERS AT THE MUSIC GALLERY**

Toronto, Ontario, Canada – Adventurous listeners can soon explore new frontiers in sound presentation as Sound Travels touches down at Toronto's Music Gallery on February

12, 13, and 14, 1999. For three days, carefully orchestrated concerts of electroacoustic and radiophonic works by Canadian composers will set of to challenge and expand auditory expectations. The concerts will be complemented by interactive installations located at the Music Gallery, using sound, theatre and puppetry to engage audiences in pressing social issues.

Sound Travels will present concert works by Canadian electroacoustic composers Martin Bartlett (with Matt Rogalsky), Ned Bouhalassa, Darren Copeland, Yves Daoust, Ken Newby (with poet Robert Anthony), Chris Rolfe, Randall Smith, Barry Truax, and Hildegard Westerkamp. There will also be a sampling of the newest composers to emerge on the Canadian scene.

Featured installations will include: Lynda Hill's Dark Forest (2-6 p.m.), a peretrating elegy to victims of violence and their loved ones drawing on sound spatialization and butoh performance; and the interactive megakiosk Bells & Whistles (7-8 p.m.) by Mark Brownell and Leslie Ashton, which blends recorded sound, puppetry, object-oriented programming, and agit-prop theatre into a peculiar ATM meets Punch and Judy fusion.

Sound Travels is a presentation of New Adventures in Sound, an organization founded by composer Darren Copeland to cultivate a whole new experience of sound that he calls 'immersive electroacoustics'. By February 14, 1999, in less than one year of activity, New Adventures in Sound will have facilitated residencies for over ten composers, produced two performance installations and co-presented six concerts in three cities (Vancouver, Montreal, and Toronto). The foundation for these activities is a new method of presenting studio-crafted music that allows a composer to automate spatialization so that sounds occurring simultaneously in a composition can move independently in three-dimensional space. Composers prepare the spatialization over a two week residency to ensure the listening experience is more life-like, enveloping, and musically vigorous.

Sound Travels is presented by New Adventures in Sound and the Music Gallery and is sponsored by Harmonic Functions, Third Monk Software, Show Pro, eyeWeekly, Le Conservatoire de musique de Montreal, EuCuE, Music Section of the Canada Council for the Arts, and the City of Toronto.  
Darren Copeland <darcope@interlog.com>

---

---

## A HAPPY GLUT OF WINDOWS MUSIC SHAREWARE AND SOFTWARE SOME PERSONAL SELECTIONS

Warren Burt  
waburt@melbourne.dialix.com.au

There are those who like to write their own software from the ground up. And there are those who carefully learn one piece of software, and stick to it. Then there are those of us who are fond of found objects, seeing how cleverly we can twist them to our own purposes. The current glut of music software available for PCs running Windows 95 is certainly a bonanza for those of us in the latter category. There are dozens of programs out there on the web, some doing a variety of things, others doing only one thing, but that might be the one thing you need. For the past two years, I've been downloading and using many of these programs, trying them out, discarding some, and intensively using others. In fact, while writing this article I came across several more interesting programs, some of which are included here. There really is an explosion of new resources becoming available for the PC. This has led to my completely optimistic (and maybe totally unrealistic!) notion that this is the best time ever to be involved with music technology. The number of ways of working with sound that are available cheaply and easily is truly inspiring. The following 39 programs include my personal favourites, as well as some new ones that have piqued my interest. All of them are currently available at the websites listed. Between Feb 26, and March 3, 1999, I visited every website listed in this article. At that time, all the software listed here was available. However, given the volatility of the web, by the time you're reading this, all these sites may have vanished. Good luck.

First, some preliminaries. Good selections of currently available music shareware and software can be found at the following websites:

<http://www.harmonycentral.com/Software/Windows/>

<http://www.hitsquad.com/smm/>

(over 1900 programs!)

<http://www.sonicspot.com>  
<http://www-users.york.ac.uk/~plh102/http://mirror.aarnet.edu.au/simtel.net/Win95/music>

(a good collection of fractal music programs) If you have difficulty contacting the websites below for a particular piece of software, I suggest you try the sites listed above. Often, their links are more robust than the addresses which access the software directly. I don't know why this should be the case, but in my case, it proved to be true.

For ease of dealing with all this software, I've divided the software up into four categories: MIDI, Sound Synthesis, Sound Editing, and Live Performance-Sound. The MIDI category covers both non-real-time generating and real-time performance software. Shareware refers to software that you can download a partially or fully working version of, and then, on payment of a fee, you can enable into a fully functional version. Freeware is just that, free for the taking. Software is a commercial product, where you have to pay the price up front to get the product. Note though, that many of the software companies have demo versions you can download. Prices are listed in US dollars mostly, not out of any desire to participate in cultural imperialism, but because that's how most of the software distributors list their prices. Check the individual websites for more details. As you will see when you get to the websites above, there's lots more available than what I've listed here. These are just my particular favourites.

### A: MIDI – ALGORITHMIC COMPOSITION TOOLS

Part 1: Programs I know. Some of these are real-time programs, some just generate MIDI files. All have proven useful at one time or another.

#### 1. KINETIC MUSIC MACHINE (KMM) (version 2.07) (Software - \$49 US) <http://algoart.com>

Before there was MAX, there was Music Box, a shareware program by software guru John Dunn. Music Box appeared around 1986, and it has been being upgraded and improved ever since. It shows the MAC bias of so many academic computer music authors that neither Music Box nor Kinetic Music Machine, as the upgraded software version was renamed by Dunn several years ago, even crack a mention in standard lists of available software for algorithmic composition. This despite the fact that KMM is a fabulous program for real time algorithmic music making, and is extremely

---

---

powerful, though the power comes at the expense of its not having an elegant graphic interface. Each of the more than 100 modules, which feature many wonderful and exotic functions, is represented by a column of ascii characters. Further, numerical values are displayed in hexadecimal (base 16)(!), though decimal (normal base 10) displays are available within the program. Once you adjust to the interface, tho, you'll be amazed at the infinite ways that the modules can be connected together to make complex musical systems. It's been my main environment for algorithmic composition since around 1992, so I think that says something about its versatility.

## **2. KINETIC ART MACHINE (KAM)**

(version 2.00)(Software - \$59 US)<http://algoart.com>

Add graphic display capabilities to Kinetic Music Machine, upgrade the interface so all numbers are now in decimal, and you come up with Kinetic Art Machine, a program for simultaneous generation of real time music and graphics. The graphics generation, like the music generation, deals with primitives. For graphics - shapes, colours, and movements of them; for sound - notes, velocities, durations, continuous controllers, etc. Controlling both are basic logic modules - sequences, boolean logic, various kinds of random generation, etc. I've only begun to scratch the surface of KAM (mainly in a project where I generate graphics using stochastic methods to serve as spectrograms to be turned into sound by the various graphics-to-sound conversion programs listed below), but I can already see that it is an extremely powerful tool to generate interesting sounds and graphics together in real-time. And while you're at it, check out the other software at Dunn's site - his programs offer a number of interesting ways of thinking about musical structure.

## **3. A MUSICAL GENERATOR**

(version 1.2.0) (Shareware - \$25 US)<http://www.musoft-builders.com>

This is not a real-time MIDI program, but it's so powerful and well-thought out that it doesn't need to be. Where most shareware programs that feature fractals and the like allow you to use the fractal information to control specific parameters (usually only one) in a specific way, A Musical Generator allows you much more freedom. Any of the 10 fractals, 2 complex systems, 7 data types, or 4 miscellaneous sources can be independently applied to the pitch, durations,

dynamics, or the flow of time of any of 16 musical lines, each on a separate MIDI channel. The last parameter may need explanation. This is a parameter where the flow of time (not tempo) is controlled. In extreme cases, it can result in the linear order of the outputs of a function being totally scrambled. It gives you, for example, an interesting way of generating stochastic canons in several voices which get their information from a common mathematical function. All the parameters are editable, so that you can specify, for example, pitch ranges the functions will apply to, and also specify custom scales that the information will be mapped onto. The data types are also worth noting. You can use text files of plain numbers, or ascii text, or graphic images (in the \*.BMP format), or Lindenmayer systems. These will all be turned into data sets which can then be applied to any parameter. The conversion of graphic images to musical data uses two different algorithms. The first, for black and white only images, allows you to make drawings of graphic shapes, and to convert them to corresponding numerical data. So you could draw a black line on a white background getting higher and higher, and apply this to dynamics, and the musical line will make a crescendo. The second mode takes a colour image, and by separating it into red, green and blue components, gives you averages for the values of each of the three colours in your image. The results of this are much harder to hear at first, especially since at first everyone feeds an image of their pet cat or banana-slug into it just to hear what happens! This is quite a different way of processing graphic information into sound than the three \*.BMP to sound conversion programs listed below. I've only recently acquired this program, and I already love it. Unlike a lot of shareware, it's very deep, with lots and lots of room to explore.

## **4. TEXTURE 3.0**

(Freeware)

<http://fisica0.uns.edu.au/lrojas/> or  
[http://www.hitsquad.com/smm/win95/  
Making\\_Music](http://www.hitsquad.com/smm/win95/Making_Music)

From Argentina comes Texture, a versatile program with a single extremely powerful way of generating information. Like Russ Kozerski's classic Sound Globes, which, alas, never made the transition to the Windows operating environment, Texture allows you to draw probability distributions to control the kinds of randomness with which to

---

---

control pitch, durations, dynamics, density, etc. of your musical lines. You can also change these distributions in real time, immediately hearing the results of your changes. Further, you can define functions, which will change the settings of the parameters that the probability distributions are applied to, also in real time. For example, you may define a probability distribution where low notes are played less than high notes. You can then define a function that will change the range of the notes that this distribution is applied to, in real time. These sorts of hierarchical structures are at the very heart of Texture. As a tool for musical improvisation, it's superb. And as a tool for composition, where more time is spent on the crafting of each distribution, it's equally powerful. Texture also features a simple sequencer, so that you can save the results of different passes of the program, and then play earlier fixed versions simultaneously with a live changing one. User definable scales are also available, as is the ability to render the MIDI output of the program into \*.WAV files for sound editing use.

### **5. WINDCHIMES**

(version 1.01) (Shareware - \$20 US)<http://www.syntrillium.com>

This is a very clever "physical modeling" program for generating MIDI information. The physical model here is that of a windchime. You specify the makeup of the windchime and the kind of environment it's in. You specify the number of "chimes" (midi notes), what pitches they will play, the configuration of the chimes (circular with pendulum in the middle; in a straight line, striking each other; oval shaped; how far are they from the pendulum/each other) and the environment - the windspeed that is activating them. This speed can be either constant or varying in nature. It's a very simple program, but ingeniously conceived. And if you have a Midi Loopback Device (see below), you can route its MIDI output to other programs for further processing or saving as a MIDI file. I've actually found this program, because of the unique nature of its algorithm and musical output, to be very valuable compositionally. Several sections of my recent "Twenty Dekanies - Music for Microtonal Piano Sounds, Part 7" got their start with this program.

### **6. ALEATORIC COMPOSER**

(version 1.1) (Shareware - \$5 US)<ftp://ftp.tk.mesh.ad.jp/pub/win3/sounds>

/alcomp11.zip

This is an older program, from 1993, but it still has features that make me return to it again and again. The program originated as a thesis project of Carl Christensen, from Philadelphia, and it allows you to make any number of monophonic lines where both the pitch and rhythmic choice can be determined by either random or conditional probabilities. After selecting which kind of probabilities you want to work with, you adjust the parameters for the individual track. Each track can have its own MIDI channel, start and end time, program change, and volume (only one velocity level per track - you'll have to process the MIDI file in a sequencer to get dynamics shaping), plus its own pitch and rhythmic values selection and probabilities. If you select random probabilities, you select which rhythmic values you want to work with, and which pitches, and then click on one of the six kinds of probability distributions -Cauchy, linear, uniform, Gaussian, exponential, and Poisson. A set of values for your chosen parameters then selected according to your desired distribution. Don't like the values? Click on the distribution again, and new values will be selected. By clicking several times on the same probability distribution, you can quickly get an idea of what kind of values that distribution favors. If you select conditional probabilities, you're presented with a transition table of rhythmic values. You specify what probability of change you want for each transition. That is, a quaver might have a 15% probability of following a semiquaver, by only a 5% chance of following a crochet. Similarly for pitch - you're presented with a transition table for each pitch you want to use. You specify what the probability of it being followed by any other pitch will be. This can be quite time consuming, and its easy to set up transition tables that go nowhere - that is, that end up in silence. It's a good idea to graph these changes out on paper before you spend many minutes entering them in the program, only to find that you've made a dead-end structure. Using this conditional table, its easy to set up structures that mimic natural structures, such as protein assembly rules. Both the email address and the address for Carl Christensen, given in the help files, are now many years out of date. However, I've been in touch with him recently, so if you want to contact him after trying out Aleatory Composer, just contact

---

---

me, and I'll be glad to pass his details on to you.

## **7.IMPROVISE**

(version 1.30) (shareware \$25 US)  
<ftp://ftp.tk.mesh.ad.jp/pub/win3/sounds/impvz130.zip>

This is the second of two older algorithmic programs that I still find useful. *Improvise* was completed by David Pannett in March 1995, but again, it has useful features, such as a method of rhythm generation that's quite intriguing. *Improvise* is a real-time performance program (its output can be saved as a MIDI file, as well) that uses a uniform probability distribution applied to various parameter ranges, such as pitch, dynamics, rhythm, and texture. There are two modes of running, "jazz" and "experimental." Jazz uses tonal harmony, diatonic scales and conventional meters (it sometimes even sounds a bit jazzy!), while experimental uses the full chromatic pitch range and no meter. There are two varieties of the experimental mode, one of which allows multiple instances of the program to be run simultaneously, so if you want thick textures, this is the preferred mode. Choice of pitch and dynamics, for each of the 16 channels, is quite straightforward - you just specify a low and high limit for each. The rhythmic choice is a bit more tricky, and interesting. For each voice, you specify a certain number of rhythmic clock ticks. Then you specify a number of events that will be randomly placed within that time period. If you specify 17000 ticks, and only 1 event, for example, be prepared to wait a long time between events. Texture allows chords to happen, and you can set the ratio of chords to single notes. It also allows you to specify what percentage of the events will be played as rests. This feature thins out the texture nicely, but produces an anomaly in the MIDI file output. Any event performed as a rest will still be in the MIDI file, but with a dynamic and duration of 0. Be prepared to filter out lots of these ghost events when you process the MIDI file in your sequencer. All the controls of *Improvise* are instantly changeable in real-time. I've used it on a number of occasions when improvising with acoustic musicians such as David Tolley and Ren Walters. It was just about the only bit of MIDI software that enabled me to keep up with the rapid rate at which they change from one type of material to another. Like Carl Christensen, David Pannett has moved since this file was placed on the web. Again, those wishing to

contact him after trying out this software can contact me for his details.

## **8.BUILDING BLOCKS**

(version 1.1) (shareware \$30 US)  
<http://www.midiworld.com/AuReality>

*Building Blocks'* author calls it a modular sequencer. What it is is a full-featured MIDI processing and generating environment. It's able to take any MIDI input signal, or input from joystick buttons or motion, or mouse clicks or motion, and transform that into any kind of MIDI output signal in a flexible, user controlled manner. There are a wide range of processing modules to choose from - a full range of math functions, random generators, test generators, several kinds of arpeggiators and sequenceers, different types of clocks, delays, quantizers, waveform generators, sweep generators, scale generators, loop modules, etc. - and by patching these together, you can create extremely complex processings of incoming MIDI data, or you can generate it within the program itself. It's pretty easy to learn and use. Within half an hour of downloading it, I had a mouse controlled pitch, loudness and dynamics generating algorithm in the locrian mode happening. Also, even though the author recommends against it, you can set up recursive feedback patches within it (its surprising how many patchable programs don't allow this!), so fans of chaotic systems should have a treat with this program.

## **9.BIO2MIDI**

(version 1.1)(freeware)  
<http://algoart.com>

John Dunn's main interest is in the use of protein and DNA sequence information for making music. At his website, he has a number of products which do this elegantly. Additionally, he has this little freebie, which is in the category of those programs which do only one thing, but do it very well. In this case, the one thing is to convert text files, which have lists of proteins or dna information into midi pitch information. Since 20 of the 26 letters of the alphabet are used for coding proteins, it means that *BIO2MIDI* is actually a very specific letters-to-midi-notes conversion algorithm. I generated some interesting melodic lines by feeding in literary texts (an interview with a dominatrix, by Paul Theroux, which appeared in a June '98 issue of the *New Yorker*). These lines were then processed by other software, (in my case *Cakewalk*), so

---

---

that they formed parts of a larger composition.

### **10.FRACTMUS 2.5**

(Freeware)

<http://www.geocities.com/SiliconValley/Haven/4368/>

Composer Gustavo Diaz Jerez has given us this program, which also allows one to use different kinds of fractals to generate musical information. In this case, there are 8 varieties of fractals you can use (including old favourites like the Lorenz and Henon equations, as well as more exotic types like the Gingerbreadman and Morse-Thue equations), but the only parameter they can control is choice of pitch. Each of the 16 voices, however, can have their own fractal. And on each of the voices, you can specify which rhythmic value, timbre, dynamic level, pitch range, fundamental, and scale that you want the fractal applied to. The results of this program will usually be a series of lines, each of which plays with only one rhythmic value. Clever use of polyrhythmic relations between the voices can alleviate this. It also has a feature called the "composition maker", which allows you to set up sequences of parameters, and also allows for random choice of the parameters for each voice.

### **11.ARPX8**

(version 1.0.1.0) (shareware - \$24.95 US)

<http://www.technotoys.com>

ARPX8 is an arpeggiator program designed to process incoming MIDI information. It's very powerful. You can have up to 8 simultaneous arpeggios going, each with their own input and output range, kind of arpeggiation (up, down, updown, fragmented, random, or brownian motion), way of reading the incoming information (play held keys, or latched keys), duration, timbre, and a limited set of velocity variations. You can then have 12 sets of these 8 simultaneous arpeggios that you can switch between in real time. Further, any of the settings can also be changed in real time, making the program a much more sophisticated performance environment than it might at first glance seem. When combined with another program that generates MIDI information (such as Texture), thru the use of a MIDI Loopback Driver, the possibilities quickly multiply.

### **12.HUBI'S MIDI LOOPBACK DRIVER**

(version 2.5)(freeware)

<ftp://ftp.tu-darmstadt.de/pub/machines/ms-dos/Cica/win3/sounds/mdlpbk25.zip>

This free utility is something every Windows MIDI user should have. It simply allows you to route the output of any MIDI program to the input of any other midi program. You can have multiple MIDI Loopbacks happening simultaneously, as well. For example, you could be generating MIDI information with a composing program, routing that output to an arpeggiator, taking the output of that to a MIDI mapper, which changes, say, which pitches are applied on which midi channels, and then route the output of that into a sequencer, to capture the whole mess as a MIDI file. What this means is that any Windows MIDI programs that use standard Windows MIDI drivers can now be thought of as not just separate entities, but as modules which can be patched together, with the information generated by each program being processed by the next. This program is essential. Download it today.

Part 2: Just downloaded! Too new to evaluate properly, but you might like to know about them. While researching this article, I came across the following pieces of software, and they look interesting enough to include here before learning or testing them thoroughly. Well worth checking out, for those interested in mathematical applications to music in general, is David Strohbeen's fractal music website: <http://members.aol.com/strohbeen/fml.html>. It lists quite a few programs that use fractals that I don't have time to include here.

### **13.ARBORRHYTHMS**

(version 3.1 beta) (shareware \$20 US)

<http://www.teleport.com/~mantra>

If ArborRhythms capabilities are anywhere near what the help file says, it will provide an extremely interesting and different approach to constructing musical logics. ArborRhythms is a program based on constructing musical grammars. It can either generate it's own MIDI information, or \*.WAV files, or process external MIDI input or mix \*.WAVs. The main grammatical structure is a tree function. Different logic processes (there are over 60 of them) are linked in hierarchical trees. When a tree is "solved", all the processes interact to produce or process musical information. Additionally, the tree structure allows for the possibility of recursion, so you can set up

---

---

processes which feedback into themselves. It can be used as a MIDI effects processor, or as a stand-alone music generator. Additionally, up to 40 of these "trees," or MIDI files, or \*.WAV files can be combined in the highest level of structure, the "song," and you can even have sets of trees and files which can be accessed using fuzzy logic. For those interested in a grammatical approach to musical structure, this looks like an extremely powerful tool.

#### **14.GBLOINK**

(shareware 20 British Pounds or \$30 US)  
<http://members.xoom.com/synaesmedia/gb.zip>

#### **15.ARTSONG**

(version 2.3) (shareware \$35 US)

#### **16.MUSICLAB**

(version 3.02) (shareware \$25 US)

#### **17.MUSICNUM**

(version 2.0 beta 5) (freeware)

#### **18.MAKE\_PRIME\_MUSIC**

(version 1.1.3.2b) (freeware)

#### **19.KATORSHA**

(version 1.41) (shareware \$10 US)

#### **20.OMAR'S FRACTAL**

(version 1.0) (mealware - if you like it, send the author the price of a loaf of bread and a jug of wine) all (15-20) at:  
<http://members.aol.com/strohbeen/fml.html>

Just brief mentions of all these programs. I like the irreverence of Gbloink's authors. They say their program is not a serious compositional tool. Rather, they say, it's a silly compositional tool. Like Windchimes, their program is an emulation of a physical model, in this case three balls gliding around a surface and bouncing into the sides and each other. You place objects in their paths, adjust speed, program changes, and volumes, and the movements of the balls determine pitches and rhythms. Sounds simple, but the musical results are delightful. The full version, they say, has 10 balls to play with. Now THAT should be something to warm the most ardent complexist's heart!

Artsong and Musiclab are two similar programs which use iterated function system fractals. They both allow sophisticated control over choice of ranges of pitch, duration, volume, articulation, choice of pitch scale, etc, and they both allow you to

write scripts to determine overall changes in the style of playing. Also, both allow you to determine the kind of IFS you're going to use. The difference is that Musiclab works off parameters of the IFS itself, while Artsong uses the pattern of the IFS to have points leaping about the surface of any \*.BMP image. The values of red, green and blue at each point leapt to can determine the values of any of the parameters.

MusiNum, Make\_Prime\_Music, Katorsha and Omar's Fractal are four programs in which simple equations produce musical information. In MusiNum, Make\_Prime\_Music, and Omar's Fractal, a very simple equation (a Morse-Thue fractal for MusiNum, a way of combining and reading successive prime numbers in Make\_Prime\_Music, and the logistic (Feigenbaum) equation for Omar's Fractal) is used to change pitches in melodies. In Make\_Prime\_Music, there is a script of values, which you can change in real-time, which can change pitch and duration parameters. In Omar's Fractal, there is a very well laid out graphic interface which makes live performance on its five voices quite easy. I think that the output from MusiNum would have to be processed by a sequencer in order to be really useful, but the parameters available in the script of Make\_Prime\_Music indicate to me that it could make much more sophisticated music, or parts of a piece of music, on its own, while Omar's Fractal seems best suited to live polyrhythmic improvisation. The harmonic controls of Omar's Fractal look quite sophisticated as well. Katorsha allows you to define equations and then look at graphs of them, and apply them to pitch and rhythm parameters. Each new equation is treated as a loop, and you then assemble up to 16 simultaneous tracks of equations. Scales can be defined as well, and the results stored as a MIDI file. On first glance, Katorsha seems to offer the potential for exploring the musical usefulness of a variety of equations.

## **B. SOUND SYNTHESIS**

Part 1: Graphics to sound conversion programs. Readers familiar with Xenakis' UPIC system, or with Metasynth or Phonogramme for the MAC will know that graphics to sound conversion is an incredibly powerful method of synthesis. A Musical Generator, and ArtSong, discussed above, have graphics to MIDI information

---

---

conversion algorithms, and here are three programs, all free, which use slightly different ways of converting \*.BMP graphics files into \*.WAV sound files. Additionally, Virtual Waves, discussed later in this section, also has its own internal method of converting drawn graphics into sound.

## **21.COAGULA LIGHT**

(version 1.0) (freeware)

<http://hem.passagen.se/rasmuse/Coagula.htm>

Stockholm's Rasmus Ekman has come up with two of the most useful sound generating and processing programs I currently own, Coagula and Granulab (see below). Coagula uses bitmaps (\*.BMP files), treats them as spectrograms, and turns the graphics into sound using a granular synthesis algorithm. Black = no sound; red = sound in the left; green = sound in the right; and yellow = sound in the middle. (Blue will indicate waveform from sine to band-limited noise in the next upgrade.) You set the highest and lowest frequencies you want, how loud you want the sound to be, how long, and the size of the graphic in pixels. The horizontal axis is read as time, and the vertical axis as pitch. Additionally, you can combine other graphics with your original file in the form of filters or overlays. The filters work very well. Coagula also provides brush tools to draw spectrograms with, (including a "boil" feature which uses cellular automata to generate interesting spectral detail), and a tool to duplicate lines or objects at different levels, so that spectra of exact or randomised partials can also be generated. On its own, it's very powerful. When combined with a graphics program, such as MSPaint, its even more powerful. I downloaded it in late November '98, and I've been obsessed by its abilities ever since. The delight is that it's in the early stages of its development. Promised added features in future upgrades include the ability to do sound to graphic conversion (\*.WAV to \*.BMP), allowing for manipulation of sound spectra in the graphic domain, different waveforms for partials besides sine waves, and MIDI output as well as \*.WAV output. And at only 77k, the download time on the program is almost negligible, even with my 14.4k modem.

## **22.BITMAP PLAYER**

(version 1.25) (freeware)

<http://www.the7hills.com/bmp2wav>

This is another \*.BMP to \*.WAV converter, which uses a different algorithm than

Coagula. It's written by Victor Khashtchansky, from Russia, and it uses black as silence, and white as sound intensity. The sound files are mono, and it reads the vertical axis as time, and the horizontal axis as pitch. It has the ability to do sound to picture (\*.WAV to \*.BMP) conversion, though the conversion is a bit wobbly at the moment. Nonetheless, it's the only PC graphics to sound program I know of at the moment that goes both ways, and the price makes it well worth downloading and using.

## **23.PHONOGRAMME SVELTE**

(freeware)

<http://www.ai.univ-paris8.fr/~vi/svelte/englishsvelte.html>

The third of our free \*.BMP to \*.WAV converters, this is Vincent Lesbros' ultra-light version of his MAC program Phonogramme. Again, it has different features than the two programs above. Here white is silence, red level is intensity, and black is full intensity. The horizontal axis is time, the vertical axis is pitch, but you can specify either a linear spread for the pixels (ie each pixel equals, say, 15 hz) or a logarithmic spread (say 792 pixels per octave). This program, unlike the other two, is a massive download (over 2 meg for the self installing version, and 800k even if you don't need the \*.DLL that makes it self-installing), but since its abilities are slightly different than the other two (and different abilities imply different compositional possibilities), you might as well also have it.

Part 2: Other sound synthesis programs. These programs each feature one different way of making sound, with the exception of Virtual Waves, which is a rather complete synthesis toolkit. I've used them all, and am quite delighted with the particular capabilities of each.

## **24.SMORPHI**

(version 3.0) (shareware - \$20 US or DM 30)

<http://www.tu-chemnitz.de/~aka/>

One of the delights of the current software development scene is seeing a program that does in software something that, years ago, you spent lots of time building hardware to do. Having spent several years in the early 1970s building devices to make waveforms that changed, or morphed, in real-time, I was delighted to discover Andre Karwath's Smorphi. I love Smorphi. It's a non-real-

---

---

time program that generates \*.WAV files in a manner similar to, but more powerful than, additive synthesis. You can have up to 32 partials in a wave. Each partial can be tuned to any frequency, with a resolution in cents. Each partial can then have two seed waveforms, each exactly 150 steps long. These seed waveforms can be drawn by hand, taken from Smorphi's extensive library of functions, or imported from a \*.WAV file. These two seed waveforms can then morph into each other in a number of ways. Each partial can be any duration, have any number of repeats, and can occur anywhere in the total duration of the waveform. Additionally, each partial can have its own envelope, it's own rate and depth of vibrato and its own lowpass filter settings. The vibrato and filter settings can also morph. Then all 32 partials can have global effects applied to them, such as graphic equalization, lowpass filtering, distortion, fadeins and fadeouts, etc. I've been using Smorphi for a number of things, the most recent of which was to generate waveforms of changing harmonic spectra. I then used these waveforms as sound sources in an analog synth emulation program, and let them frequency and amplitude modulate each other. The resulting dynamically changing noisebands were extremely exciting - I was 25 years old all over again. Smorphi might not let YOU rediscover your lost youth, but it will provide you with an extremely useful method of sound synthesis.

### **25.PHYMOD**

(version 2.0) (shareware - \$35 US or DM 49)  
<http://141.84.217.141/phymod> or  
<http://www.harmonycentral.com/Software/Windows/phymod20.html>

Physical modeling synthesis is a potentially extremely powerful method of synthesis that is still in its infancy. For those wanting to get in on the ground floor, this shareware program provides an ideal vehicle. In Phymod, you build "sound sculptures," which are collections of "masses" (each having its own user defined weight) which are connected by "links." These links can be activated by "hammers," "bows," or "gums," that is, they can be made to move in various ways. Strings, plates, resonant chambers, etc, can be set up, and they can be either passive or dynamic, continually changing in real time. They can either generate sounds internally, or external \*.WAV files can be processed through them. Additionally, if one uses Stelken's proprietary DSP system, the

system can be used as a real time processor of sound input. Physical modeling of this sort is quite a different way of thinking about sound synthesis, and I've just begun to scratch the surface of what this program is capable of. So far, though, in my limited experimentation with the program, I've already been producing sounds with an internal life that I hadn't gotten from any other method of computer sound synthesis. Phymod is another tool well worth having.

### **26.DEFRACTOR**

(version 0.97B) (freeware)  
<http://www.sonicspot.com> or  
<http://neurotix.303dim.com/>

Jenner Fusari, from Italy, offers this sound synthesis program. It's a particular realization of one 2-oscillator fm-like patch. Oscillator 2 can feed back on itself, and waveforms are selectable (including a user-specifiable one). The modulated waveform then goes through amplifiers with distortion capabilities, and a unique filter algorithm. Sounds made can then be saved as \*.WAV, \*.XI or \*.RAW formats. This is one of those programs that does only one thing, but that one thing can generate a wide variety of dynamically varying sound types.

### **27.ADSYN32**

(version 2.0) (freeware)  
<http://www.sonicspot.com>

For those who want a quick and accurate additive synthesis module without going to all the fuss and bother of learning CSound, this is the place to turn. Andy Bridle, from England, has made an extremely simple and elegant realization of a non-realtime additive synthesis engine. You can have up to 32 partials. Each partial can be either at harmonic or subharmonic relations with the fundamental, or you can have custom partials, where the frequency of each partial can be specified to any desired degree of accuracy. Each partial can be either a sine wave or one of two waveforms labeled "fractal" and "random" which feature a smattering of higher harmonics, or white noise. Each partial can then have its own envelope, and a mixer allows overall shaping of the loudness relationships of all the partials. Again, a one-purpose utility, but a very well made one.

### **28.VIRTUAL WAVES**

(version 2.22) (software \$199 US)

---

---

<http://www.synoptic.net>

Virtual Waves is, for this listing, relatively expensive, but what a program. Short of learning CSound, you're not likely to get as rich and versatile a collection of functions in one package anywhere else. And I found it so much easier to handle than CSound. VW is a non-real-time program which generates mono \*.WAV files, and the maximum duration of any one component of a "patch" is 10 seconds. Now, having described every aspect of the program I don't like, let's proceed onto the rave review. Many ways of generating sound are included in VW, including Additive, FM, Cellular Automata, Formant Wave Functions (FOF), Karplus-Strong, MIDI file (where Midi continuous control information can be used as a sound source or as a form of control), Mathematical Functions, Noise, Spectral Sketch Pad (an internal graphic to sound conversion module - it can't convert bitmaps, but your drawings are saved in its internal \*.DES format), a couple types of Physical Models (reed and plucked strings), and a "wave sequencer" which emulates a type of early digital synthesis. These sounds can then be patched into a large number of processes, including, but not limited to amplitude modulation, envelope following, transfer functions (waveshaping), frequency modulation, resonant filters, spectral modulations (one sound morphing into another), time-stretching, pitch-shifting, a wide variety of reverb algorithms, etc. A number of analysis tools are also included in the package, including oscilloscopes, short term FFT, sonogram, etc. All the modules can be patched together, analog-synthesis or MAX-style, resulting in a sound making toolkit that is very, very deep, and well worth the price.

## C: SOUND EDITING

### 29. COOL EDIT 96

(shareware \$50 US full **PRO** version \$399 US)

<http://www.syntrillium.com>

If I were to say that Cool Edit is to sound what Photoshop is to images, I'd be displaying a visual bias, but I'd be correct. What the two programs have in common is a depth of function that is breath-taking. What they don't have in common, though, is price. Even the full commercial version, Cool Edit Pro, is still less than half the cost of Photoshop. The manual (in \*.PDF format) for Cool Edit is about 230 pages. It's worth

printing out and reading every page. The number and kinds of processing this program can do are breath-taking. Cool Edit 96 has almost all the functions of the Pro version, except for multitrack mixing, and a couple of more exotic envelope follower and vocoding functions. But all of the wide variety of delays, reverbs, filters (including morphing fft filters), convolution modules, noise reduction functions, etc. are there. Cool Edit is almost a complete sound editing and processing facility on its own. I've also found it to be quite a handy tool for live improvisation. It's processing speed on some functions is so fast that I found I could easily record myself or another of the performers, modify the sound in some way, and then quickly play back the modified sound, repeating this process again and again, until the original character of the sound was totally lost. Then, with 64 levels of undo, I could retrace my steps, as it were, playing the modifications in reverse order until we arrived back at the original sound. There are a number of very powerful sound editors on the market, such as Sound Forge, and Samplitude. Choosing between these and Cool Edit is largely a matter of personal taste. After trying them all, though, my favorite remains Cool Edit.

### 30. SMS TOOLS

(version 0.8) (freeware)

<http://www.iaa.upf.es/~sms/>

Those who have been attending ICMC conferences over the past few years tell me that Xavier Serra has been talking about and demonstrating this program, and that he had been saying that the Windows version was about to occur. Here it is, and it's a winner. Briefly, the program analyzes and incoming sound, using a fast-fourier transform, into sine components and residuals (everything else). The parameters that can be set to get a good, or a useful bad, analysis are extensive. Then, transformations can be applied to the fft. This is accomplished with a rather sophisticated set of graphics. For each available analysis parameter, and there are many of them, one can draw a graph to change that parameter. Transformations can be applied to amplitude, frequency (I especially like the individual scaling of each partial transform), modulations, time stretching, and hybridizations between two files. Then the file is resynthesized, and the resulting sound is heard. The program also saves its analysis files, so those with a bent for programming should be able to decode

---

---

these and write their own routines to process these to do things the program can't. I spent most of the winter of '95 writing Basic programs which modified analysis files from another program, so on my list of things to do is rewriting those programs to handle these analysis files. This program is deep, complex, and seriously written. Those wanting to deal with the analysis-resynthesis paradigm should get this one.

### **31.HYPERPRISM PLUG-IN PACK**

(version 1.5.1) (software \$349 US)

<http://www.arboretum.com>

Hyperprism is a commercial product, and the price listed above is the manufacturers recommended retail price. I've seen it at other places (and bought it) for considerably less than that. Even at full price, however, its quite good value for money. The PC version comes as a series of Direct-X plug-ins, so you need some other program, such as Cool Edit, that accepts plug-ins, in order to access its features. The central feature of Hyperprism is its by now famous "blue window," which allows your mouse to control any two (or more) of the available parameters for the chosen effect. On the MAC version, your mouse movements can be remembered and played back. This facility is not available on the PC versions, because of the limitations of Direct-X. However, the mouse movements do work in real time. This has led me to use Hyperprism, in conjunction with Cakewalk Pro Audio, mainly as a live performance resource. I have any number of \*.WAV files available on several Cakewalk tracks, and using the mixing facilities of Cakewalk, route them through several Hyperprism effects. In performance, I select which \*.WAV files are to be routed to which effects, and change the settings on the effects. In version 1.5.1 there are 29 effects, some of which are simply utilitarian, others of which are quite creative. A vocoder, frequency shifter, pitch changer (different from the frequency shifter), low-pass and hi-pass filters, ring modulator, reverbs, panners, chorus units, compressors, etc. are all available. Most effects also have a modulation parameter as well, which allows you to change some aspects of the sound with a low frequency oscillator. Whether as a live-performance resource, or as a series of extensions to your sound-editing program, Hyperprism is well worth investigating.

### **32.HOG**

(version .013p) (freeware)

<http://shoko.calarts.edu/~bcassidy/hog>

Hog is from California Institute of the Arts, the home of the MAC based Sound Hack, and Brendon Cassidy, the author, had considerable help from Tom Erbe, the author of Sound Hack, in making this program. It does only four things, convolution, equalization matching, ring modulation and decimation, but there are a number of parameters available for manipulation within each function. The ring modulation has three different modulation algorithms to select from, for example. The program is also accompanied by some excellent, clear and helpful release notes. The "walk-through" of the convolution function features the clearest and most useable explanation of how convolution works that I've yet encountered. By now, you've probably realized that you can set up a unique and powerful sound processing and editing studio on your PC for very little money by simply using cleverly written utilities like these.

### **D: LIVE SOUND-PROCESSING**

These programs either generate sound in real time, or, if you have a full-duplex sound card, enable you to process sounds in real time. Even if you don't have a full-duplex card, you can still use them for live processing of \*.WAV files. As above, each of the programs does something different. I've used all but one of these extensively (Generator), and I can recommend them all. In fact, these programs have made my laptop my main performance instrument over the past two years.

### **33.AUDIOMULCH**

(version 0.7b6) (shareware \$50 US or \$50 AUS for Australian users)

<http://www.audiomulch.com>

Those of us in Melbourne have been watching the evolution, over the past several years, of Ross Bencina's AudioMulch. Now that Ross has moved to Adelaide, we can only follow developments at a distance, but c'est la vie. AudioMulch is in a state of

constant upgrading and development, and each new version has added capabilities. For example, a recent email from Ross said that he's nearing completion of MIDI parameter control. This will enhance the usefulness of the program greatly. It's hard to control all

---

---

those parameters with just your mouse and keyboard! The last upgrade had 3 new beta modules, a "bubble blower," a fast, but limited implementation

of the older and more complex granular synthesis module, and an 8x8 and 4x4 matrix mixer, which allow you to route sounds to various outputs in real time, with control of the fade times on the switching. AudioMulch uses a patching paradigm. The screen is divided into two areas. In one, you connect over 20 functions together to make the kind of processor you want. In the other, you arrange the control panels of the modules to suit your taste, making a custom performance area. Because of the nature of some of the modules (loop player, drums, bassline), AudioMulch has gotten a reputation as a techno-oriented program. It can do techno-music elegantly, but it's much more powerful than that. For example, in one patch, I've been using 4 loop players with changing-harmonic-spectra waveforms made with Smorphi, mixing them and putting them through the stereo delay module. In performance, changing the overall tempo of the loops, or the number of bars they play changes the speed of playback in whole number ratios. I use this to assemble a rich drone of continually changing harmonics tuned in the small-number ratios of just intonation. The use of the delay line allows the newly selected pitches to combine harmonically with the old. AudioMulch, like many of these programs, does require a powerful computer to do many simultaneous functions. My Pentium 133Mhz laptop struggles when more than three functions are called for, but even under that limitation, very sophisticated things can still be done. (Ross reckons that a 166Mhz machine is really called for to run the program well. He himself was using a 266Mhz machine and getting all the power he needed for pretty complex setups.) The sound spatializer module deserves particular mention. It allows you to draw a trajectory for your sounds to move thru space, with selectable speed, doppler shift, etc. Ross can hold his head up high as part of the long tradition of Australian electronic music designers, starting with Percy Grainger, and continuing through such folks as Tony Furze and Greg Schiemer. Already impressive, I'm eagerly awaiting each new upgrade to AudioMulch.

#### **34.MELLOSOFTON II**

(version 2.5) (shareware \$29 US)  
<http://www.polyhedric.com/>

There are a number of programs which claim to be able to turn your computer into a sampler. Some are very expensive. Others, like Tommy Anderberg's Mellosotron, are cheap and powerful. The last non-laptop-based piece of equipment in my studio I used regularly for gigs was my venerable Ensoniq EPS sampler. At 20 kilos, though, carrying it to the gig was quite a chore. Since getting Mellosotron, my level of back pain has lessened considerably. I now take 4 kilos of laptop and auxiliary equipment to the gig instead. Because it uses Direct-X, the latency problem in Mellosotron is almost non-existent. For those of you unfamiliar with it, latency is the delay between when you tell a computer to do something, and when it actually does it. For some applications, like word processing, a delay of 1/10th of a second is not a problem. For music, it's disastrous. Having played with a number of programs in the pre-Direct-X days (Direct-X is Microsoft's free set of sound playing utilities which speed up sound playing and response time considerably), I can say that the advent of a genuine low-latency sampling program in my price range has been a real boon. Mellosotron enables you to have up to 16 "programs" available simultaneously. Each program consists of any number of samples (\*.WAV files), each with its own key range, response characteristics, envelope settings, and loop settings. The envelope parameters can be edited by hand, or graphically, but Anderberg's unique method of graphing envelopes may take a little getting used to. These "programs" are then combined into "maps," combinations of programs which can be switched between. If you don't have MIDI input, the program also has a small mouse-activated keyboard, and a set of bend, mod, volume and pan sliders available. I've been doing pieces recently where, using a computer keyboard to midi utility, I play a number of samples from the computer keyboard, letting my programming shape the sound's dynamics. It sure beats carrying 20+ kilos of gear to the gig....

#### **35.VAZ**

(version 1.5 - freeware) (version 1.6 - shareware \$35 US/ 25 British Pounds)  
(MOD - version 1.0 - software 129.66 British pounds (AUS\$326.96 - NZ\$387.23))  
<http://www.software-technology.com>

There are a number of analog synthesizer emulations floating around at the moment.

---

---

Many of these are mono simulations of vintage analog gear. VAZ started out that way, and the freeware version 1.5 and more powerful shareware version 1.6 are exactly that - a good analog synthesizer emulation. It has been upgraded however, in the slightly more expensive modular version, into a very powerful synthesis and sample playback system. For me, one of the chief joys of the program has been its ability to use \*.WAV files as sound and modulation sources. (This ability also exists in the earlier freeware and shareware versions.) This means that I can make complex waveforms in any of a number of other programs, and then use them in VAZ as frequency and amplitude modulators of each other. Some extremely complex changing timbres can be made in this way. Suddenly, the analog patching paradigm of sound design seems exciting again. I also like the fact that VAZ allows modules to be plugged together in feedback loops. This makes recursive patches, with their quasi-chaotic behaviour possible. In addition to the standard complement of oscillators, amplifiers, filters, etc. there are a number of other interesting functions in VAZ modular, such as frequency dividers, wavehaping, a variety of exotic envelope generators, envelope following, panning and a host of logic functions, such as comparators, inverters, slew limiters, logic gates, etc. There is also an analog sequencer emulation lurking behind the panel of each MIDI input panel, so the program can generate its own internal control signals, as well as responding quickly (thanks again to DirectX) to a complete range of incoming MIDI signals. Finally, there are rudimentary effects that the sounds can be routed through - chorus, delay, flanger, and reverb. The delay units feature 2.6 seconds of delay, and can be chained together. One patch I use connects 3 of these together to get a 7 second delay, which I added to a patch already straining the limits of my computer. The additional load on the CPU was negligible. You can, of course, save the output of the program as a \*.WAV file, too. In the year I've had it, VAZ has become one of my most used programs. Software-technology, by the way, has a website that is able to charge you in your home currency at the current exchange rates. The prices above were relevant on 2 March, 1999. Since then, anything could have happened.

### **36.GRANULAB**

(version T (transitional))(shareware - \$20 US individuals \$50 US institutions)

<http://hem.passagen.se/rasmuse/Granny.htm>

Rasmus Ekman, whose Coagula was mentioned above, also has Granulab, a MIDI controllable granular synthesis engine, which allows great flexibility in real-time control of its many parameters. Like Coagula, Granulab is currently very much under construction. A freeware, one voice only, version 1.0 currently exists on the website. Pay him the registration fee, and you'll get the 8-voice transitional version mentioned here, as well as the future version 2, which promises more levels of control. In the transitional version, each of the 8 voices can have its own \*.WAV file or internal sine wave, its own control settings, and its own presets. Latency is a bit of a problem here, but the program displays the latency time in a small window, so at least you know the kind of control delay you're going to get. Each voice can have up to 20 presets (settings of all parameters), and the presets can be morphed into one another. The rate of changing from one preset to another is also controllable. Some of the best sounds I've gotten out of it have been during these transitions from one state to another. The program is capable of some wild distortions, but can also be quite subtle as well. In a recent collaboration with San Francisco writer Elizabeth Block, for example, it was clear that her text would always have to be understandable. Granulab gave me plenty of scope for making subtle, yet interesting manipulations of her voice. Another interesting feature of the program is that it will take certain aspects of the sound, such as its amplitude, and use those to control other aspects of the sound, such as panning. This provides another level of interest to the program. A final attractive feature of the program is its size - about 64k. Even on the slowest modem, that means download times are negligible. If you want a program that does granular synthesis and sound modification, and that does it cheaply and elegantly, you should check out Granulab.

### **37.MIXMAN STUDIO**

(version 1.5) (software - \$49 US - seen in Melbourne shops for \$99 AUS)

<http://www.mixman.com>

Mixman is a fun program for combining samples in real time. It has wonderful hipper-than-thou graphics in its interface, and with its proprietary \*.TRK format, you can change the tempo of its samples without changing their pitch. It can also use \*.WAV

---

---

files, but these will change tempo and pitch together. You can have up to 16 looping samples available at any one time, and these are controlled from the computer keyboard. Additionally, the mouse can be used to control the pitch, volume and panning of each sample, and the overall tempo at which the loops are played back. You can also save a record of all your actions as a \*.MIX file, and when you get a \*.MIX you like, you can export it as a \*.WAV file. It also has an excellent and clearly illustrated manual, which assumes practically no prior musical or computer knowledge. Mixman is aimed squarely at the techno, house, and club remix culture, and for that world, its superb. Practitioners of other musical styles of reassemblage may also find it quite friendly and useful. I know I have.

### **38.BACKGROUND NOISE**

(version 1.6) (shareware - \$10 US, printed manual \$5 US extra)  
<http://members.aol.com/pjsoftware/>

This is one of a number of utility programs that can play a variety of sound formats. It is, however, the only one I know of which allows you to randomize the order of play of the files in such a way that each chosen file is played just once before a new random ordering of the set is made. It can also play sound files in a purely random order, allowing for the possibility of multiple repeats of a file, and can, of course, play things in sequential order. It was this random feature that made it absolutely invaluable to me last year when I did a series of sound installations in Victorian regional art galleries. Using Background Noise, I could select a vocabulary of sounds, select the percentage of silence and sound I wanted, and then record as many different cassettes or CDs of the sounds as I needed to play as part of the installations. The program plays \*.AVI, \*.WAV, and \*.MID files, and can also play tracks off any CD in the computer's CD-ROM drive. This is an example of a program which does one particular function, but it was just the function I needed at the time.

### **39.GENERATOR**

(version 1.5.5) (software -\$298 US)  
<http://www.native-instruments.com/>

I haven't used this program much, aside from playing a bit with the demo version to get the feel of it, but last November, in Santa Fe,

New Mexico, composer David Dunn gave me a thorough demonstration of it. He was using it to make extremely complex real time iterative function chaos patches. The 180 modules of Generator allow you to set up heirarchical patches which can loop back on themselves, have a wide variety of customizable controls, and can be controlled from external MIDI sources. To my eyes, it looks like the most powerful of the current crop of patchable synthesizer emulation programs. It will obviously take a while to learn, but like the old modular analog machines, the amount of time put into learning it was, in Dunn's case, richly repaid in the abilities he then had with the program. The download of the demo is big - several megabytes, but if you want to work with a system that makes no apologies for its complexity, and rewards you with increased abilities, you should at least try Generator.

Enough already! I now want to actually compose (!) with some of these programs!

---

### **The ACMA Committee for 1998/1999**

**President:** David Worrall  
David.Worrall@anu.edu.au  
**Vice President:** Andrew Brown  
a.brown@qut.edu.au  
**Secretary, acting Treasurer:** Warren Burt  
waburt@melbourne.dialix.oz.au  
**Membership Officer:** Garth Paine  
garth@creativeaccess.com.au

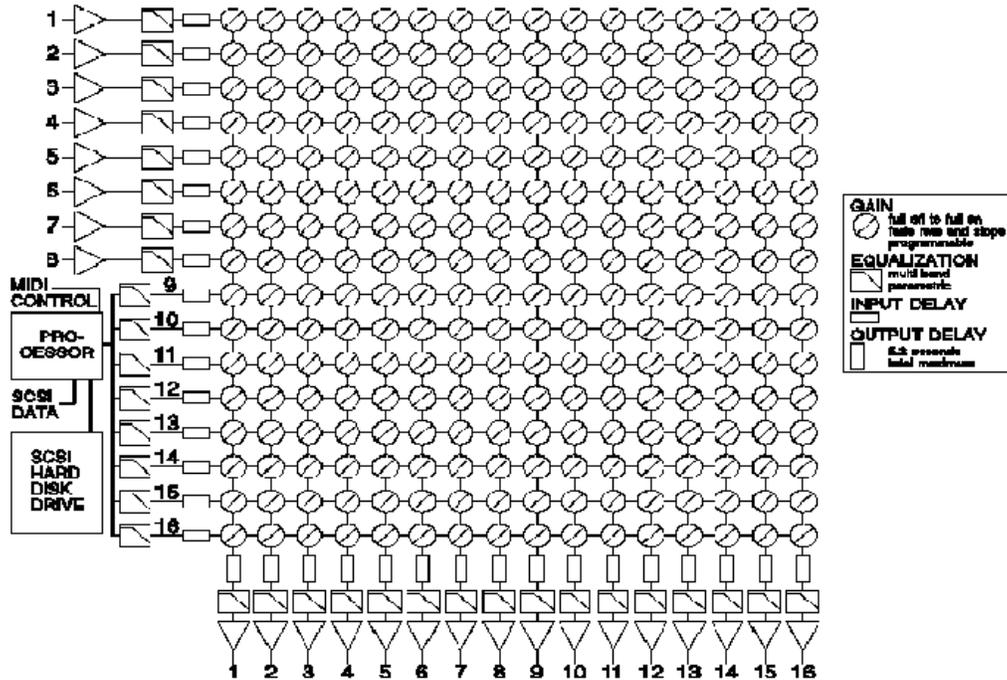
#### **Regional Representatives**

**New South Wales:** Anthony Hood  
ahood@conmusic.usyd.edu.au  
**New Zealand:** John Young  
John.Young@vuw.ac.nz  
**Queensland:** Daniel Horwood  
dp.horwood@student.qut.edu.au  
**South Australia:**  
Christian Haines  
cmh@cobweb.com.au  
**Western Australia:** Leigh Smith  
leigh@cs.uwa.edu.au

---

---

## AudioBox DM1616HD Signal Flow Diagram



---

## CONTACTING ACMA

To contact ACMA, see our website: <http://farben.latrobe.edu.au/ACMA/>

Oz-Computer-Music is a listserv based at La Trobe University in Melbourne. It acts as a centre for discussion amongst electroacoustic composers in Australia and beyond.

To subscribe to oz-computer-music:

Write to  
LISTSERV@LUGB.LATROBE.EDU.AU and,  
in the text of your message (not the subject line), write:

SUBSCRIBE OZ-COMPUTER-MUSIC



---

**Notice:** Anyone who has CDs that they would like reviewed / mentioned / discussed in the next bumper issue of Chroma (July) should get in touch with Warren or Garth before the end of May.

---

---

## Australasian Computer Music Association

### Application for Membership

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
- signal processing
- MIDI
- music notation
- software and hardware, including commercially available products
- algorithmic composition and strategies
- Macintosh, SGI, PC etc.

Membership entitles you to receive and contribute to Chroma, the Association's newsletter, and participation in all ACMA events for 12 months. Back issues of Chroma are available at \$3.

Annual Membership Fees (in Australian Dollars):

- Unemployed/students: \$ 15
- Employed: \$ 25
- Institution: \$100
- Assembly CD: \$ 20
- Machine Messages CD: \$ 10 (only 30 left!!!!)
- Overseas members add extra postage expenses: \$ 15
- CD purchases in Australia add postage: \$ 3

Please complete the details below and forward with your cheque to:

The Secretary  
Australasian Computer Music Association, Inc.  
PO Box 284  
Fitzroy, Victoria 3065  
Australia

Cheques should be made to the Australasian Computer Music Association, Inc.

Name: \_\_\_\_\_

Is this a renewal of membership? YES/NO

Address: \_\_\_\_\_

\_\_\_\_\_

Telephone: \_\_\_\_\_

Fax: \_\_\_\_\_

Organisation: \_\_\_\_\_

E-mail: \_\_\_\_\_

Particular areas of interest/equipment/software used: \_\_\_\_\_

\_\_\_\_\_

Do you agree to allow your name, address, telephone number and interests to be circulated to other members of the Association? YES/NO

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

