Chroma

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

Number 6
October 90
ISSN 1034-8271

In This Issue

Australian New Music Conference 1990

The Australian New Music Conference (formerly the Australian Composers Conference) was held in Brisbane from 17-20 August at the Queensland Cultural Centre. Sponsored by Sounds Australian (Australian Music Centre), it was held in conjunction – well, simultaneously with the Musica Nova Festival.

The main themes were those one can read about in recent issues of the Sounds Australian Journal – tertiary music education in Australia and the "Complexity Debate" in music composition, with some very interesting sessions on music and technology and libretto writing.

There were over eighty delegates - composers, performers, teachers, arts and education administrators, representatives from the ABC and even a few interested members of the general public. Despite the variety of interests and opinions there was a general feeling of goodwill, tolerance and fellowship among the delegates. However, many attended only for their own speeches or specialist sessions and unfortunately missed the remainder of the conference. The conference was rich in information and enthusiasm – with perhaps a touch of frustration and even desperation about the state of Australian music. Actually, that's not quite correct. The frustration seems to be with the how Australian music is regarded (or largely ignored) by those in government or influential postions.

There were concerts every day and night, with a special concert of live electroacoustic music on Saturday night titled *Murmurs of Chaos*, preceded by Greg Schiemer's UFO music. This was an impromtu performance given by

Participants in the New Music Festival Concert *Murmurs of Chaos*. From left to right - Ian Fredericks, Greg Schiemer, David Worrall, Graeme Gerrard, Virginia Read, David Hirst, Warren Burt and Rodolphe Blois.

a number of delegates using Greg's electronic oscillators housed in plastic containers. These were suspended on strings and swung around in a style that nicely echoed Sara Hopkins' earlier performance, swinging swimming pool tubing above her head, whirly-whirly like.

The concert included *Think Out Loud* by Graeme Gerrard (Melbourne) for prerecorded digital tape, sampler and effects; an improvisation by Ian Fredericks (Sydney) using an Atari computer and Warren Burt (Melbourne) on Casio MIDI horn; a stunning soundscape study - *Etude de Paysage*, by Rodolphe Blois (Brisbane); two movements from Warren Burt's *Chaotic Research Music*, a piece based around chaos, 1/f melodies, self-similarity and the like; an improvisation by Ian Fredericks using his own software – *Iansmuse*, on an Atari and David Hirst (Melbourne) with his program *Spray* on the Macintosh; and finally *Ravin' like a Writhing Disk* by Virginia Read and David Worrall (Canberra) running their *EventMaker* program on a Macintosh, and featuring an audio/MIDI feedback loop using a Pitchrider pitch to MIDI converter. The ABC recorded the concert and several of the pieces were broadcast by John Crawford on his program *Random Round* a couple of weeks later.

The papers were very broad in content, ranging from descriptions of software composition systems, hardware designs and synthesis software, to considerations of algorithmic composition and the place of music technology in secondary and tertiary education generally. The proceedings of the conference will be available from Sounds Australian, so you can read the papers in detail.

There was some good humoured and bloodless jihad in the "my programming language is superior to yours" war (not including Fortran, Pascal or Basic) and a surprise appearance and talk by composer and pioneer computer musician James Penberthy, about his work in computer composition in the mid 1960s.

Discussions with Dick Letts of Sounds Australian resulted in an arrangement being worked out for the distribution of electronic music on tape. The preferred format is high quality (e.g. metal) cassette copies with Beta/PCM masters, for the time being. They will also soon be able to handle DAT tapes, as this format gains in popularity and availability. So, if you want your music to be available and distributed to users of the Sounds Australian facility, and there are many, send your tapes in to them at PO Box 49 Broadway 2007. (You get a 15% royalty on all rentals of your tapes).

The next Australian New Music Conference is to be held in 1992, possibly in Melbourne, and if the 1990 conference is any guide, participation is recommended.

Chroma is edited by Graeme Gerrard.

The Relevance of Copyright to Synthesis, Sampling, and Computer Generated and Assisted Compositions, Part II.

COPYRIGHT AND SYNTHESIS

- Shireen Tippett

Patch Copyright

To date, the area of copyright and sound synthesis has received little attention from lawyers and government bodies, and naturally musicians or programmers themselves tend to be more interested in their art than in the legal aspects involved. The need for patch copyright in synthesis is quite clear - programmers spend many hours creating and refining their patches, only to find that they are being regularly ripped off. "...the fruits of their labor are being freely shared among musicians, and in some cases even repackaged and sold by unethical hustlers who fail to give the original programmers credit. In neither case does the programmer get paid for the work that went into a patch." (Tomlyn, 1986, 11). If this situation continues, it is highly likely that programmers will cease to find programming worth the time and effort, and the quality of programmed sound may suffer severely. With the complexity and variety of synthesisers today, unlike in the past, it is frequently necessary for musicians to employ professional programmers because it is no longer always feasible for a good session musician to also be a proficient programmer. As it is, not all professional programmers are adept at programming all types of synthesiser, and many actually specialize in a specific type of synthesis.

In many instances, if the software hasn't been pirated exactly, adaptations are made. Bo Tomlyn (1986) feels that credit should be given where it is due, and believes that if an adapted patch is sold, 70% should go to the original programmer, and 30% to the person who sold the modification. No doubt as synthesis technology advances, more problems will arise and the courts will of necessity need to address these.

It is not at all clear from the Australian Copyright Act as to what level of protection, if any, is afforded to sounds themselves. In the United States, it is possible to follow legal precedents, but to date in Australia there have been no legal cases. Therefore we should observe with great interest the growing number of overseas examples in order that a solution to this problem may be found.

The main aspect of synthesis and copyright is that of patch copyright, or more specifically, the ownership of individual or banks of sounds. No mention whatsoever is made of this in the Australian Copyright Act, and according to a representative from the Australian Copyright Council, there are currently only two ways in which a sound itself may be copyrightable :-

a) If the sound itself is considered to be music or an integral part of music, in which case copyright is afforded to the musical work; or

b) where the sound forms part of an actual master sound recording, in which case no part of that recording may be copied or reproduced.

Because of the situation in the United States, I thought it

possible that the Australian Copyright Act might also allow copyright of patches in the same manner as it permits the protection of computer programs, but this is not the case. According to the Copyright Council, the main problem with copyright of a sound in Australia, is that the sound on its own is considered under copyright law (although not specifically mentioned in the Act), to be an 'idea' only, and 'ideas' themselves, unless they fall within the two categories listed above, may not be copyright protected at all.

Surprisingly in Australia, there are few articles debating this important issue, and there haven't been any legal precedents. In the US however, with larger numbers of professional programmers, the problem appears to be receiving a considerable amount of attention, particularly in *Keyboard Magazine*.

In the United States, "Computer-controlled analog and FM synthesis programs are partially copyrightable because they are operating control code for microprocessors", according to Bryan Bell of 'Synth Bank' (Alvaro, 1986, 10). But,

> "...the problem isn't necessarily with the copyright law itself. The law says that, for a work to be copyrightable, it must be an 'original work of authorship' ". Whether or not a particular work is copyrightable depends on the interpretation of this phrase. 'Original' means only that the person actually created the work rather than copying it; it is not necessary that a work be 'novel' to be 'original'. 'Authorship', however, requires that a "certain minimum amount of creative expression" must have gone into the development of the thing being created. Furthermore, an idea by itself is not copyrightable; it must be expressed in a unique way, and it is the expression that is copyrightable. ...The finished product of this process is copyrightable, unless the author's way of doing it was the only way in which it could be done." (McCaffrey, 1989, 12).

Therefore, in developing a patch, the programmer takes an idea and uses a creative process to transform that idea into the sound. Programming according to Tom McCaffrey is without a doubt a creative activity because, in taking into account the number of parameters and values on a DX7 for example, the number of possible patches would be 100 to the 78th power. "With such a vast number of choices, it's easy to see how unlikely it is that two independent programmers will come up with exactly the same patch." (McCaffrey, 1989, 12).

In the United States, for $10 a bank of patches may be sent in for copyright registration to the Library of Congress in Washington D.C. on form 'PA'. This registration should be titled "compilation of synthesiser patches", and should also be accompanied by data sheets. Copyright protection will not extend to individual patches, nor will it extend to the names of these patches. This is because the United States Library of Congress considers that the work required in creating an individual patch is not significantly creative enough to warrant copyright, and under the US Copyright Act, names or titles are not copyrightable. Nevertheless, registering a bank of sounds is worthwhile in that it may prevent someone from pirating the entire bank (McCaffrey, 1989).

In the United States, one known legal precedent in a similar area appears promising for synthesiser programmers. In 1986, Atari won a major lawsuit against a mail order catalogue for taking video game data out of Atari's cartridges and selling it in a cheaper cartridge. Bo Tomlyn (1986,11) believes this case "establishes that synthesiser sounds, like video data, are copyrighted work."

Although it would appear that synthesiser patches are copyrightable in the United States, in reality it will be extremely difficult to benefit from this 'legal right'. The cost of taking an infringer to court is approximately US$50,000, and the time taken in court proceedings would no doubt be more productively spent programming. Even if the case is won, it will only be the defeat of one of many infringers. Bo Tomlyn's own solution to the problem is to appeal to consumers to question where their bought patches have really come from, and before buying, to make sure they are getting 'good' patches. "Let's stop using second-hand sounds" he says, "and start supporting the people who sit in studios programming all day. Good programmers are crucial. Let's not exile them from the industry." (Tomlyn, 1986, 154)

COPYRIGHT AND SAMPLING

The issue of sampling and copyright involves three distinct sections:-

1) copyright of original samples;
2) the use of pre-recorded material, and
3) the use of live material.

Copyright in Original Samples

The problem of copyrightability of original samples is very similar to that of patch copyright in synthesis, except that initially, samples are digital recordings, and it is only when they are modified that they become computer-controlled programs. Therefore, the raw (unmodified) sample on its own may not be copyright protected unless it forms and integral part of a musical work. Once it has been processed however, it may be protected under the United States Copyright Law as a computer-based literary work because of its microprocessor control code. As with synthesis, this area is somewhat unclear, and certainly not specifically mentioned in the Australian Copyright Act or many others for that matter.

Sampling and the Use of Pre-recorded Material

The use of pre-recorded material in samples is clearly a breach of copyright unless royalties are paid, permission has been obtained, or the person sampling owns the rights to the particular recording. This area is quite clear in the Copyright Act, because sampling from a recording constitutes an infringement under the rights laid down for copyright in sound recordings (as outlined earlier). It is also an area which has been receiving much publicity in the recent years with the developments in Rap music, Dance music, Hip Hop, and to a lesser extent, both Rock and Rhythm and Blues. 'MTV' in Australia recently showed a short program

on current problems in the United States. This demonstrated many examples where fragments of earlier recordings had been sampled and used in not only new recordings, but new works. Some of these are :-

* Cold Cut sampled Ofra Hazi, a female Israeli singer from a then obscure record.

* Public Enemy *The Rebel* "borrowed" from James Brown's *Funky Drummer*. Their justification was that James Brown was old hat and now they have made him new again!

* Ton Loc *Wild Thing* - the introduction clearly resembles the introduction in Van Halen's *Jamies Cryin'*.

* Robert Plant *Tall Cool Mama* "borrowed" from Led Zeppelin's *Black Dog*, with the lines "Hey Hey Mama".

* Janet Jackson with *Rhythm Nation* "borrowed" from Sly & the Family Stone's *Thank You*.

* Bob Base *Get on the Dance Floor* "borrowed" from Michael Jackson and *Shake Your Body*.

* Dela Soul *Say No Go* "borrowed" a riff from Hall & Oates' *I Can't Go For That*. (Shtein, 1989)

Dela Soul were recently involved in two legal cases. The first, a US$1 million law suit for taking a large part of a recording (*You Showed Me*) by the Turtles. The producers for Dela Soul believed they had processed the sample sufficiently for it not to be recognised, but in actual fact, the similarities are quite apparent. This case was settled prior to going to court, which is perhaps unfortunate in that no legal precedent was set. In the second case, Hall & Oates sued Dela Soul for "borrowing" from their (Hall & Oates') song *I Can't Go For That*. This case was also settled out of court.

The British band Black Box in *Ride on Time* admitted to "borrowing" without permission Loleatta Holloway's vocal sound. Holloway received no royalties from the millions of dollars that were made by the infringers, and she was unable to take the case to court because of the high costs involved. (Patterson, 1990)

Philip M. Cowan (Shtein,1989), a United States copyright lawyer believes that the reason Rap is so successful is purely because of the background (i.e. recognisable hits) rather than for the Rap itself. Many companies and individuals believe that unauthorised sampling from a record is not only theft, but a clear breach of copyright, and many recording companies, musicians and composers would like to protect their music from such theft. Many feel that copyright is still inadequate as a result of the high costs involved in taking legal action. Nevertheless, the growing number of legal cases, and the accusations that sampling is theft are beginning to have some effect. In finding material to sample, more musicians and composers are applying for permission from copyright owners, and/or paying royalties. In this regard, copyright is beginning to take effect and is therefore quite relevant. However some, like Frank Zappa, believe Copyright is not conclusive enough. He now prints a notice on his recordings, prohibiting unauthorised sampling.

In order to prove a breach of copyright through sampling, the plaintiff must first prove ownership of the copyright in the sound recording. Secondly, they must prove that the sampling used in the defendant's work was a direct dubbing of the actual sounds as opposed to an 'independent fixation' of other sounds. And finally, they must prove that the defendant's recording substantially and materially copies their own (McGiverin, 1987). The first clause raises some interesting problems in that frequently the owner of the recording is not the owner of the music. In which case, if the owner of the music wishes to put a stop to the breach, they must first convince the owner of the recording to take expensive legal action. Secondly, in order to prove that the sampling was in fact a direct dubbing, the owner of the recording must show obvious similarities to the court. Finally, it is up to the courts to determine whether that which was 'stolen' constitutes a substantial part of the music - this can be difficult. In the United States Judicial system this is usually decided by a jury, whereas in Australia, the decision is usually made by the appointed Copyright Tribunal. It is debatable which of these two systems is best. In the United States, judgement is in effect made by representatives of the consumer, whereas in Australia, the decision is made entirely by professionals.

Sampling and the Use of Live Material

The use of live material in samples is a complex issue in that the rights of performers may be breached when the appropriate authorisation is not obtained. Not only that, such a practice on a large scale could effectively put session musicians out of work. Standard procedure in music recordings has been to pay a session musician to realise the music, simply because these professionals have spent years mastering their instrumental skills. However, it is naturally less expensive to sample a performance of say a violinist, and to use these samples to replace the cost of employing a session violinist. "Just a few seconds of sampling from a compact disc, for instance, allows us to orchestrate a composition with the string section of the Chicago Symphony - without paying the exorbitant price of a sampling session. Thanks to technology, stealing from other musicians has become a simple operation" (Oswald, 1988, 12). Without the permission of the performers, such a practice is quite unethical and a breach of either the Copyright Act in Australia, or the Performing Rights Acts in the United Kingdom and the United States.

Until the Australian Copyright Amendment Act of 1989, performers in Australia were afforded no protection whatsoever of their performance. In 1985, a report was put together by the Copyright Law Review Committee on whether there was a need for the legislative protection of performers in respect of their performance, and if so, what form this legislation might take. Some interesting arguments both for and against were presented, but generally, the move towards more protection for performers received overwhelming support. One exception was that several members of the committee could not believe unauthorised use of performances ever occurred in Australia (Copyright Law Review Committee, 1985). Nevertheless, it was decided that such a move was in fact necessary, and the Committee's proposals were incorporated into the Copyright Act in 1989 as Part XIA - Performers' Protection, Section 248A-V. This means that in Australia, a person may no longer make an unauthorised use of a performance, with several exceptions.

The Performers' Protection division of the Copyright Act (1989, s.248G) lists the following as constituting unauthorised use of a performance :-

"(1) A person makes an unauthorised use of a performance if the person, at any time during the protection period of the performance and without the authority of the performer:

a) makes a direct or indirect recording of the performance;

b) broadcasts or re-broadcasts the performance, either directly from the live performance or from an unauthorised recording of it; or

c) causes the live performance, or an unauthorised recording of it, to be transmitted to subscribers of a diffusion service.

(2) A person makes an unauthorised use of a performance if the person, at any time during the protection period of the performance and without the authority of the performer:

a) makes a copy of a recording of the performance that the person knows, or ought reasonably to know, is an unauthorised recording;

b) makes a copy of an exempt recording of the performance, being a copy that the person knows, or ought reasonably to know, is not itself an exempt recording;

c) makes, for use in a sound-track, a copy of an authorised sound recording of the performance and the person knows, or ought reasonably to know, that the making of the sound recording was not authorised for the purpose of use in that or any other sound-track;

d) has in his or her possession a recording of the performance that the person knows, or ought reasonably to know, is an unauthorised recording;

e) sells, lets on hire, or by way of trade exhibits in public or offers or exposes for sale or hire, a recording of the performance that the....;

f) distributes a recording of the performance for the purpose of trade, or for any other purpose to an extent that will affect prejudicially the financial interests of the performer or performers in the performance..."

Although no specific mention is made at any time of unauthorised sampling from a performance or an unauthorised recording of a performance, it should be understood that to do so would be a clear breach of the Act.

[In the next issue, the thorny area of copyright of Computer Assisted and/or generated Compositions is addressed. Is the author of such works the programmer, the compiler of the data base, the computer operator, or the computer itself? Or, do all of these have some right in the output?]

Three More Interactive Composing Programs for the IBM-PC

-Warren Burt

Several months ago, I wrote about *M* and *Sound Globs* interactive composing programs for the IBM-PC. Since then, things have changed somewhat. *M* for the Amiga, Atari, and Mac is now distributed by Dr. T's software. *M/pc*, the IBM version, is still distributed by Voyetra. *Sound Globs* is now distributed by Cool Shoes Software, which is the company set up by Russ Kozerski, one of the authors of *Sound Globs*. Addresses will be found at the end of this article. I would now like to continue and look at three other programs. Each one is different in its orientation. *Drummer* is a prepacked drum machine program with algorithmic composition abilities. *Ravel* is a complete language, oriented towards MIDI implementations. *CAL* is an extension language accessory provided with Twelve-Tone Systems *Cakewalk Pro* sequencer. Each one gives you resources you otherwise wouldn't have, providing you with more ways of thinking about composing, more glimpses into how other people think composing works.

In one sense, *Drummer* written by Russ Kozerski, and marketed by his company Cool Shoes Software, doesn't belong in this article. *Drummer* is basically a very nice, and very inexpensive program that emulates the function of commercial drum machines. In terms of algorithmic composition, "it does only one thing, but that one thing it does very well", to use Harry Partch's marvellous phrase.

In *Drummer* you make one measure patterns. These patterns can have from 1 to 8 beats at 1 to 8 divisions per beat. You can have 20 MIDI note numbers specified, that is, have a choice of 20 of any note number on any MIDI channel. Each pattern can play at the tempo you specify or at any ratio of that tempo from .001 to 9.999 times it with a bottom tempo of 8 beats per minute and a top tempo of 240. Any note can have one of ten user specified volume levels. You can have 25 patterns and select between them in real time or assemble a "score" which is a predetermined ordering of patterns. You can have up to 2000 patterns in a "score", which can be saved as a MIDI file. So far so good, and at $79.95 US, every IBM owner with a MIDI interface should buy one and keep Russ Kozerski afloat, in order to subsidize his work *Sound Globs*. But now *Drummer* gets interesting – because of one feature. A "fill" feature has been added, which allows you to specify what percentage of the time a particular note will randomly play. Each of the 20 pitches can have a different percentage and loudness level, and these can be changed in real time. For example, if a note is specified to happen 50% of the time, the result is an interesting, unpredictable, but quite bouncy rhythm that clearly emphasizes the underlying pulse. If 2 notes are specified with each having 33% probability, the result is an interesting hocket reminiscent of African bell patterns. Having 20 independent lines, each consisting of one sound each, and each having its own probability of occuring strikes me as being a new kind of counterpoint. Although the lines

don't affect each other, in sound result this strikes me as being similar in result to what a cellular automata might produce. The implications of this are rather staggering, both of working with Drummer and in other languages. And since each pattern can have its own tempo, beat and subdivision, Drummer can also be an ideal tool for exploring sudden tempo changes and metric modulations. I have also found it to be extremely useful for sound sculptures and sound installation work. And at the price, who with an IBM will *not* want to have it around?

There's also one cute feature that should be mentioned. If you want to save a *Drummer* "score" as a MIDI file, you don't have to listen to it- it will save all the MIDI information very rapidly. If you have random "fills" specified, each MIDI file produced from the same "score" will be unique. Because of this faster than real-time saving of information, it could be possible to specify a piece, save it to MIDI file, load that file into a notation program and convert it to notation in less time than it would take to play the piece. This notion of compositional automation carried to the extreme, of "hyper-real time", if you will, strikes me as happily, joyously ridiculous.

The other two programs are not prepackaged environments, but full-blown languages. Jim Binkley's *Ravel* is a fully integrated programming environment. It is a modular, somewhat object-oriented language that has some features of C and some of Pascal. *Ravel* has a 350 page manual, written in a somewhat pedantic programmer's manner. However, the manual is very thorough, and is fairly self-contained. That is, it would help in learning *Ravel* if you knew C, but it's not essential. As one who hates learning new languages with a passion, I can say that *Ravel* is relatively easy to learn, though learning it *is* time consuming.

Ravel *and* Cakewalk *are not prepackaged environments, but full-blown languages.*

In *Ravel*, you write a program using your own text editor. Then you compile that using "mc", its compiler system. Then, once you've successfully compiled your program, you play it using "mos", *Ravel*'s MIDI operating system program. A *Ravel* program is structured with global variables specified first, then "riffs", (subroutines) are specified, and finally "vco's", or voicelists. Each "vco" is a parallel program which specifies MIDI information on (usually) one MIDI channel. A *Ravel* program can have up to 32 voicelists operating on all 16 MIDI channels.

On hearing the output of your program, you go back and rewrite it, recompile it, rehear it, etc. Salt to taste. Since *Ravel* is modular, you can develop tools with it which can then be linked into other programs in the future, such as my strange attractor generator, which gives a stream of random values using the Verhulst/Feigenbaum equation $x_{next} = rx(1-x)$. *Ravel* has a number of good composing tools of its own as well. Many of the one dimensional and two dimensional matrix manipulation features described in the Dodge and Jerse *Computer Music* book are implemented, as well as a variety of random number generators including – "fractal" 1/f noise, evenly weighted random, and "triangle distribution" random, which is a poor man's bell shaped curve.

Ravel provides a "windows" facility, so that you can specify a number of windows which will serve as real time controls of the operation of your program. With *Ravel* you can gradually build up an interactive MIDI environment suited to your needs. *Ravel* is not without its limitations. Its rhythmic precision is limited to 24 pulses per quarter note, but since the fastest tempo allowed is 240 beats per minute, this has not seemed like such a problem, though I haven't yet used it for precisely metric music. After each performance, *Ravel* also lists what are called "timewow errors", which are the number of clock ticks it dropped while it was engaged in computation. This gives you an indication of the efficiency of your program writing for real time operation. *Ravel* also only implements single-precision fixed-point 16 bit arithmetic, so all of the skills you developed in FORTH for faking floating-point arithmentic may be of use here.

Since *Ravel* allows input and output of standard ASCII files, it's useful for other things than real-time music. I've used it, for example, to generate printed texts for a sound poetry performance, and also for direct non-real time sound synthesis, generating *Sample Vision* format sound files which I then loaded into the *Sample Vision* sound editor for transfer to an Akai S900. In my piece *Strange Attractors and the I-Ching: Concatenating Logics, not Shapes*, I used both the strange attractor equation and an I-Ching emulation to synthesize sounds. These sounds were then transferred to the S900, and the same equations used to select which samples, at which pitches and loudnesses would be played. Aspects of this selection were controlled by me in real time. *Ravel* currently only imports MIDI files. However, it also imports and exports MIDI information in the *Cakewalk* ASCII file format, so if you have the *Cakewalk* sequencer, you have a path available for notation or use of *Ravel* output in sequencing, etc. All in all, *Ravel* is a pretty powerful environment for researching interactive composition, one that I get to like better each time I use it. Jim Binkley has recently put *Ravel* in the public domain, and has said I can be the "Australian distributor" if I wish, so anyone wanting a copy should contact me. It's on 3 floppy disks and has a 350 page manual. Copying the disks will probably not be a problem for you, but you may want to prepare yourself for a long session at the photocopier.

The *Cakewalk* sequencer from Twelve-Tone Systems is a relatively inexpensive, very fully featured sequencer package. It has a wide range of standard sequencer features, is easy to learn and easy to use. It also has a very good user interface which enables it to be used as an interactive performance tool. Three of the 10 pieces of my *Chaotic Research Music* use it as a performance device. In addition, the latest update of Cakewalk has added a Lisp-like extension language, *CAL* or *Cakewalk Application Language*. This is a non-real-time facility which enables you to change events in *Cakewalk* event lists in ways the program doesn't specify. It is a fairly versatile language. I was surprised how

many features were jammed into it, although again, it only does 16 bit single precision arithmetic. After learning *Ravel*, *CAL* was very easy to learn. Anyone with a bit of programming experience should be able to learn it fairly quickly. With *CAL* there is an implied, built in loop. This loop takes *Cakewalk* events, one at a time, and performs the desired modifications on them. Since one of the modifications is "insert", however, you can also use it as an event generator. To give an example of how *CAL* can be used, in my piece *A Fig Tree (In a Post-Coltrane Environment If is Too Highly Correlated)*, I first wrote a *CAL* program to generate phrases using the Verhulst/Feigenbaum equation. Then I used *Cakewalk*'s facilities to extend those phrases into contrapuntal episodes. Then I wrote another *CAL* program to transpose each of these episodes into the just intonation scales of my choosing, specifying the proper MIDI bend number to accompany each pitch. I then used *Cakewalk* during real-time performance to select which of these episodes, at what tempo, and with what timbres would be playing. Used in this way, *Cakewalk* becomes more than a simple sequencer, but becomes a very powerful composing/performing environment. The idea of extensibility is one that hopefully, more and more programmers will build into

Contributions

Being so late in the year, the next issue of *Chroma* will be another double issue, like *Chroma* 3 & 4. To make it a double, we need contributions!

What new software is available for your machine?; care to describe a recent piece you have made?; did you go to CyberArts in San Francisco?; what about a studio description/report?; review of the latest thingummy XR5?; Is all this technology is constraining musical innovation? If you don't think you could manage a whole article, what about a letter?