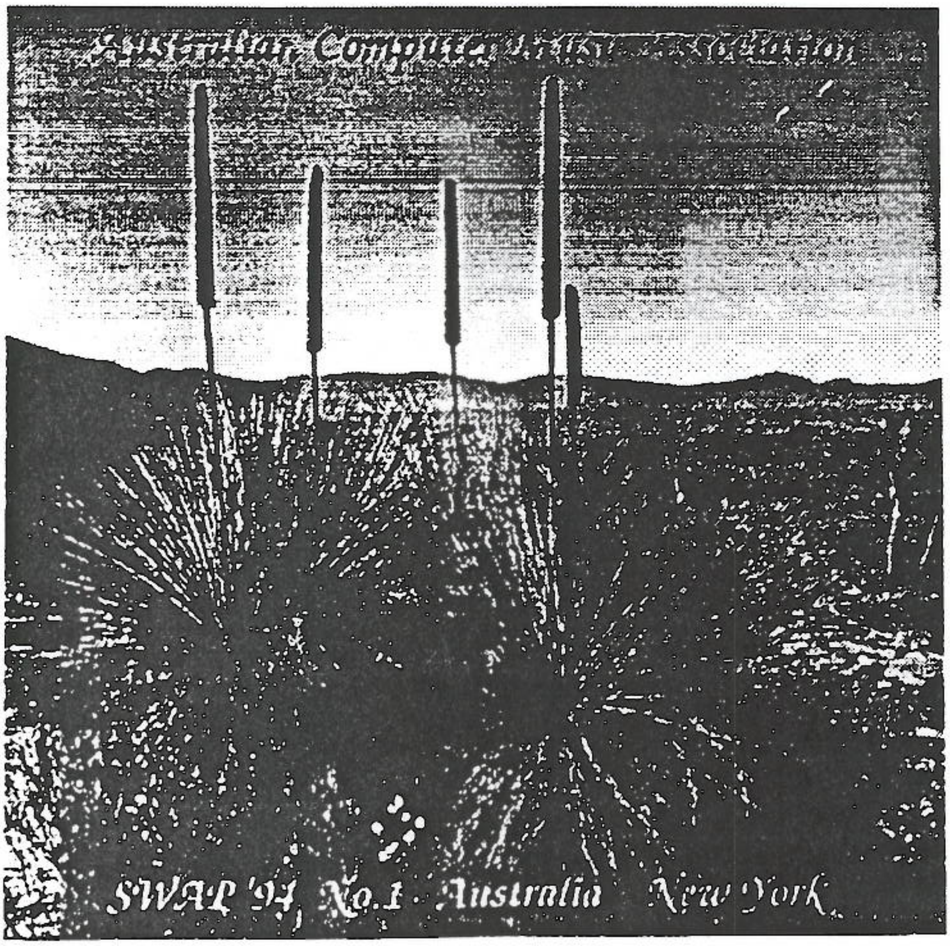
Chroma

Newsletter of the Australian Computer Music Association, Inc.
PO Box 186 Post Office Agency La Trobe University VIC 3083

Number 17
June '94

ISSN 1034-8271

In This Issue...



Contents

News and Info

ACMA Swap CD

May saw the publication of the first in a series of *Swap CD* projects. These are a form of musical exchange wherein a CD containing works by ACMA members is sent for performance overseas, while we in turn perform works from the overseas institution as part of an ACMA concert. Only a couple of custom CD's are pressed for the purpose, being much cheaper than a production run of discs.

Our inaugural volume, produced for exchange with Princeton University in New York, consisted of works by Gordon Monro, Tim Kreger, Warren Burt, Alistair Riddell, David Hirst, Steve Adam, Steven Campbell and Graeme Gerrard. This CD was then used for a concert there in late May.

We are already working on the next Swap CD, which is planned for a Canadian performance in October of this year. All submissions and expressions of interest are welcome, with the cost to each contributor likely to be around $20.00. Please contact any of the committee members if you are interested.

ACMA Concert Series

We are pleased to anounce that the Melbourne ACMA concert series is underway, with our first concert scheduled for the weekend of 27th and 28th August (exact date to be confirmed), to be held at North Melbourne's Elm Street Hall. During this concert, some material from the swap CD project will be performed, as well as new material from ACMA members. All submissions and ideas for live performance or tape pieces are welcome, so please contact the committee if you feel that you can contribute.

We have a second concert date confirmed for Sunday November 27th, at a venue to be announced, as well as plans for a third Central Business District concert in Spring. Once again, all expressions of interest are welcome, and we will keep you posted as details come to hand.

ACMA on Radio

Starting the first week in August, ACMA will be on-the-air with a regular radio programme. The show is part of Hard Listening, on 3PBS. ACMA, in association with Melbourne Improvisers Association will be taking over the show from Warren Burt. Warren will be out of Melbourne for around 12 months, visiting and working in Canberra and the U.S.

ACMA will present a one hour show each fortnight of electroacoustic music from Australia and overseas. An opportunity for new presenters to join the show is being offered by ACMA and 3PBS. PBS requires all new presenters to attend a training course to familiarise them with the studio and aspects of broadcasting practice in Australia. The cost of the course is $100 and is covered in three, two hour sessions. If would like to do the show, contact Lawrence Harvey on (03) 387 8474. Calls for material will be made in Chroma, so watch this space and respond!!!

Membership Renewals

Just a reminder that all memberships are due for renewal as at the end of February. We have had a steady stream of renewals, but we need to get the stragglers in now too, to ensure that we are not sending out Chromas to lapsed memberships. Given this, this Chroma will be the last posted to members who are non-financial for this year.

ACMA's Fifth Birthday

The committee at ACMA are proud to celebrate the fact that June 1994 sees the fifth anniversary of the foundation of the Australian Computer Music Association, Incorporated. In these years we have achieved many things, including the production of seventeen Chromas, two CD's and twelve concerts. To mark the occasion, we have included a chronology of ACMA's evolution and an index to the first sixteen issues of Chroma in this issue.

Chroma is edited by Roger Alsop, Ross Bencina and Thomas Stainsby.

Chroma is published bi-monthly and back issues are available at $2.50 each, with a compilation of issues 1-9 available for $18.00.

Acknowledgements:
Thanks to La Trobe University Music Department & Richard Lewis for the use of their computers in compiling this news letter.

Program for
SYNAESTHETICA '94

Symposium to be held at

Australian Centre for Arts and Technology
Australian National University
July 1 - 3

Friday 1st July

3.00pm - 6.00pm	Registration
6.30pm	Official opening by Barry Jones of Exhibition 'Contours of the Mind' a celebration of Fractals, Chaos and Feedback.
7.15pm	Performance by ACAT staff and students.

Saturday 2nd July

8.00am - 9.00am	Registration
9.00am	Keynote Address
9.30am - 11.45am	Session 1
11.45am - 1.30pm	Lunch
1.30pm - 3.00pm	Session 2
3.00pm - 3.30pm	Afternoon Tea
3.30pm - 5.00pm	Session 3
5.30pm - 7.00pm	Concert 1
8.30pm - 10.30pm	Concert 2

Sunday 3rd July

9.00am - 12.00pm	Session 4
12.00pm - 1.00pm	Lunch
1.00pm - 3.00.pm	Session 5
3.00pm - 3.30pm	Afternoon tea
3.30pm - 5.00pm	Session 6: Open Forum
5.00pm - 6.00pm	Session 7: Plenary

Session 1

Interactive Audio via the MIDI Tool Box
Greg Schiemer

A Practical System for three dimensional sound projection
Kimmo Vennonen

New Sounds for old forms: the Contribution of Changing Instrumental Technology to Film Music Scoring
Ian Whalley

Digital Sound Manipulation for Film and it's role in Affecting Emotions
Stephen Joyce

Session 2

Reaction-Diffusion Data in Musical Event Structures
Andrew Martin

Modelling & Control of Flagella and Cilia for Locomotion of Artificial Life
Alan Dorin & Justin Martin

Musical Applications of Typogenetics
Tim Kreger

Session 3

Creating Music from Speech
Roger Alsop

Exploring Recursive Function Systems
Stuart Ramsden

Evolving Fractal Industrial Music
Ben Goertzel

Session 4

Tunings for Nonharmonic Sounds
William A Sethares

A perceptually ordered geometric model of sound inspired by colour theory
Stephen Barrass

Modern Synthesis and Sampled Sound
David yeeOn lo

Static & Dynamic Sound Warping
Jon Drummond and Gordon Munro

Session 5

The Virtual Performer- Approaching Meaning
David Boyle

The Interface Composed
Alistair Riddell

The Absent Body: The Impact of Technology on Music
David Rodger

Morphing the Oz Indie Film Scene Paradigm or Why we need the Digital Media Commission
David Cox

Rasa of Algorithmic Composition
David Worrall

Session 6: Open Forum
Synaesthetica 94 delegates are invited to submit in no more than 10 lines a question or statement outlining an issue pertaining to the field of endeavour that they consider currently important. An open forum for questions and discussions relating to these ideas. Due date June 17.

* Program subject to change without notice.

Venue - School of Music

Accommodation

Lakeside Hotel
$105 per room (single, double, twin)
includes a fully cooked buffet breakfast in Bobby McGees Restaurant

Lakeside Hotel
GPO Box 1450
Canberra Australia
Phone (06) 247 6244
Telex 24796
Fax: (06) 2573071
Toll Free: (008) 026169

Burgman College
$48 per night fully catered
$40 per night Bed & Breakfast

Burgman College
(ANU Campus)
GPO Box 1345
Canberra ACT
Phone: (06) 267 5222
Fax: (06) 257 2655

Registration details:

Fees (Australian Dollars)

Registration Fee	$120
Student & Unemployed	$90

Registration includes attendance at all cultural events, a copy of the proceedings. Lunch and Morning & Afternoon Tea on Saturday & Sunday are also included.

Call for Tapes
Swap CD No.2

Submissions are now invited for inclusion on the second ACMA Swap CD project, to be arranged in collaboration with the Canadian electroacoustic music community.

All submissions should be on DAT tape, preferably at 44.1 kHz, and must be no longer than 10 minutes in duration.

Tapes should include:
- 30s silence at start,
- Start ID 5s before start of piece;
- Counter readings in absolute time mode,
- Indication of peak amplitude in dB;
- Specification of intended peak amplitude in dB upon final mastering;
- Freedom from any clicks and glitches.
- Programme notes;

All enquiries should be directed to Lawrence Harvey on (03) 387 8474

Approaches to Interactive Computer Music Programming

Roger Alsop

Department of Music
La Trobe University
E-mail:ralsop@klang.latrobe.edu.au

This paper was given at La Trobe University on June 1 this year. It outlines some of the approaches for createing improvisation environments within a computer. Four approaches are shown, each based on a diferent philosophy and each yielding very diferent results. Max is the software used to create these programs.

What does INTERACTIVE mean?

In his book Interactive Music Systems Robert Rowe describes three classifications of interactive computer systems, these are: programs that are score or performance driven; programs that are transformative, generative or sequenced; and programs that follow an instrument or player paradigm (p 6-8).

Score driven programs match input data to stored data then output some form of interaction, Rowe suggests that this method provides a more traditional interaction because actions cause predictable interactions. A score driven program propels the musician through a piece, making changes as the piece is being played. In this case the composition drives the interactions and is considered hierarchically more important

The opposite is true for performance driven programs. These "do not anticipate the realisation of any particular score" and result in a more intuitive interaction because the computer's actions are based on the musicians input data. Here the actions of the performer drive the composition.

Transformative methods transform existing data or the musician's input data stream to create the interaction. Generative methods create interactive data from algorithms created by the musician. Transformative or generative programs create with and along side the performer. While interesting and challenging interactions may be produced severe constraints are usually placed on the program to ensure some sense of cohesion between the musician and the program.

Sequencing methods output pre-recorded or programmed data in response to input data. Here the composition's constituents and parameters are set before presentation, the musician ordering the events as he or she goes along.

Methods following the instrument paradigm aim at making an extended, or hyper, instrument. In this case the output is greater than the input yet is resultant of that input. Programs that follow this paradigm create challenges similar to those of a traditional instrument. Here designers are often concerned with the interface between musician and machine. The player paradigm creates a virtual musician, containing a "personality" and behaviour of its own. Of course this "personality" is usually programmed by the musician.

Constructing a "player" with its own "personality" is by far the most challenging for the musician, he or she must define what they want in a musical collaborator and then accept the results, which can range from boringly predictable to frighteningly unpredictable.

These classifications are given as guides to types of real time interaction, each has its uses, advantages and disadvantages depending on the result desired and can all, and often do, exist in the one interactive program.

Types of musical interactions

There are many different musical types of musical interactions, below I give a much too brief outline of them.

The first type of musical interaction is the one a musician has with the relationship the musician has with his or her instrument. In this case the musician's creativity must fit within the parameters of their chosen instrument. These parameters severely limit the creative, rather than interpretive, musician's expressive pallet, forcing them to create with a very limited variety of sounds. Fortunately most musicians come to terms with this problem intuitively and unconsciously during their development, bending their expressive desires and actions to fit the domain of their chosen instrument.

Interactions between musicians often follow highly predictable paths. For instance, in an orchestra all musicians are all interacting in very subtle ways. The goal of these musicians is to create the best possible interpretation of the given piece, be that what it may and to do this attention is given to the subtle nuances of pitch, timbre and phrasing, all under the guidance of the conductor or section leader.

In a common jazz band the variety of and types of acceptable interactions are increased but these interactions are usually less subtle than those within an orchestra. In a jazz band interpretation is considered a form of personal self expression and it is important that this self expression be unique and well developed. However it is essential that this self expression fit within the style and aesthetic of the band. It is also essential that a unified sense of rhythm and tonality exist amongst the members of the band.

In the more "experimental" areas of music the varieties of creativity and interaction subsume interpretation and therefore increase again. However these interactions are still confined by the stated and assumed aesthetics and agendas of the group and the aesthetics, agendas and abilities of the groups members.

This problem of aesthetics and agendas applies to orchestras, jazz bands and experimental groups. Prevailing traditions will usually define these aesthetics and agendas to a greater or lesser degree. In the case of orchestras and jazz bands there is a very strong, well defined and well known set of stylistic parameters which must be adhered to unless there is a deliberate and stated intention to break from the current traditions. In experimental groups it is often the participating musicians who create their own stylistic definitions, aesthetics and agendas. While there is an assumed set of traditions for "experimental groups and musicians it is usually accceptable to break through these boundaries.

What is not often explored in musical interactions are the styles and methods of other forms of human interaction. For example, the most common form of human interaction, conversation, is a series of short solos around a theme. The thematic course of a conversation often darts wildly from expositions to variations to recapitulations to transitions to developments to unrelated themes and all with apparent ease and acceptance by the participants.

Musical interaction usually occurs in three ways:

parallel interaction, where the participants strive for similarity in their actions, for example playing in the same mode or with a similar degree of rhythmic density;

contrary interaction, where the participants strive for dissimilarity in there actions, for example playing in a way to frustrate the other participants; and

oblique interaction, where one, or more of the participants supports other participants.

Of course there are no clear boundaries between these types of interaction and whenever boundaries are crated they are almost constantly in flux. This is the paradigm in which 'musical' interaction exists.

Should we try to emulate these processes when programming computers?

To emulate musical instruments is a goal of many electronic instrument makers. As one area of instrument emulation is covered another opens up. The first attempts centred around trying to create electronic instruments that sound like acoustic instruments. When this was achieved to an acceptable degree of success, problems with expressivity came into play. Solving this has been an ongoing problem which Yamaha and CCRMA claim to be close to solving this problem (the Computer Age: May 17, 1994).

The problem in creating a truly expressive electronic instrument is in the interface. When a human plays an instrument such as a violin a myriad of interactions between the musician and the instrument occur. For example, extremely slight variations in the pressure and position of the bow and the position of the fingers has a profound effect on what the musician expresses and what the listener perceives. Controlling these variations is the goal of practise and the more "expressive" musicians are those who control these details in a method that readily communicates to the listener. Creating an electronic instrument with this sensitivity to variation is extremely difficult, confronting both the physical and philosophical domains of electronic instrument production.

Computer emulation of human to human interaction, where invention is part of expression, is far easier than computer emulation of human to instrument interaction. The computer's abilities are not as critical and the network of variations need not be as comprehensive for meaningful human/computer interaction to occur. The program "Band in a Box" quite successfully simulates the styles of instrumentalists in a very wide variety of musical genres, however it does not allow the apparent randomness and unpredictability of human nature that exists in creativity. If aspects of human nature, such as moods and varying abilities, are considered acceptable in human to human interactions then the various inabilities of computers could also be acceptable in human to computer interactions.

However emulating human to human interaction is just one of the possibilities available when interacting with computers. It is possible to use the computer as a conductor or event schedular, deciding what is available to the musician at different times in the piece, as a composer whose efforts are guided and constrained by the musician, as a pallet of events to be constructed by the musician and as a cohort in creation, whose input has as much weight as the musicians. Below are four interactive programs I have created in the Max programming language. Each of these exemplifies one of the possibilities given above.

Four interactive programs

In the first three programs a midi guitar is used as a controlling and/or seeding device, this need not be the only interface. In the fourth the computer keyboard is the controlling device.

The first program, **MIDI guitar** (see diagram 1 below), uses a mathematical/compositional approach. Here a stream of numbers is put through a series of algorithms and effects resulting in a musical (MIDI) output. The program deals with cycles of musical data. These cycles can be of varying duration and event amount. For example, a cycle may repeat seven events over seven minutes or seventy events over seven seconds. Having two independent cycles allow for interaction between cycles. Two of the possibilities this creates are of phasing between streams and a perceived melody and accompaniment. It is possible to create two cycles and then improvise, in the traditional sense, with them in real time.

This program has been used extensively in improvisations with ancers. The rhythmic and periodic capabilities and the realtime interactive and automated approaches have produced interesting results. I have found that, as a general rule, longer cycles create more interesting music/dance interactions in that the dancers intuitively synchronise with the musical periodicity after the cycle has repeated a few times, this occurs when the cycle is heavily disguised and without the dancer being intellectually aware of the nature of their accompaniment.

The program for the piece "**Sometimes I lie awake at nights** ... , (see diagram 2) which was performed at the La Trobe Moat Concert this year, is an attempt to introduce some random aspects of human nature into the program. These revolve around the computer supplying variations to my input data. In this case the computer, or co-improviser, reacts to my input stream within a set of choices I deem to be appropriate to my goals for the piece. The computer chooses from a number of possibilities effecting: pitch, articulation, note density and timbre. I can set the amount and type of possibilities and how the computer steps through these possibilities. I consider this as akin to discussing the ambit and pathways of an improvisation with other musicians or to setting the traditions, that is aesthetics and agendas to which the improvisation holds.

The program for "**Reply**" (see diagram 3), performed at Linden this April, is compositionally based. In this piece I set a pathway through which I could improvise. These improvisations are stored and then replayed by the computer with variations made to pitch and timing. In some ways this program is similar to using a chord chart or some other form of static guidline to create a form within which to improvise. Becoming proficient improvising with this program requires the same exposure and the same amount of practise as one would give to become proficient improvising over a standard tune.

I have recently been examining processes of collaboration between musicians and dancers with graduate choreography students at the VCA. So far this year I have worked with the dancer John Utan to present a piece in a workshop type environment. First we spent a few hours discussing the philosophy behind what we wanted to present and then the method by which we would present it. The result was a number of dance and musical cliches, appropriating themes from different styles and genres then butting these themes against each other first sequentially and then in an improvised/random fashion.

The program here simply plays eight pre-programmed sequences. However it has the added ability of changing MIDI channels and transposing the pitches of the sequence as it goes along. This results in some interesting and weird permutations. In performance I used the program much as one would use a traditional instrument, that is, I was able to make an action and accurately predict the reaction of the instrument.

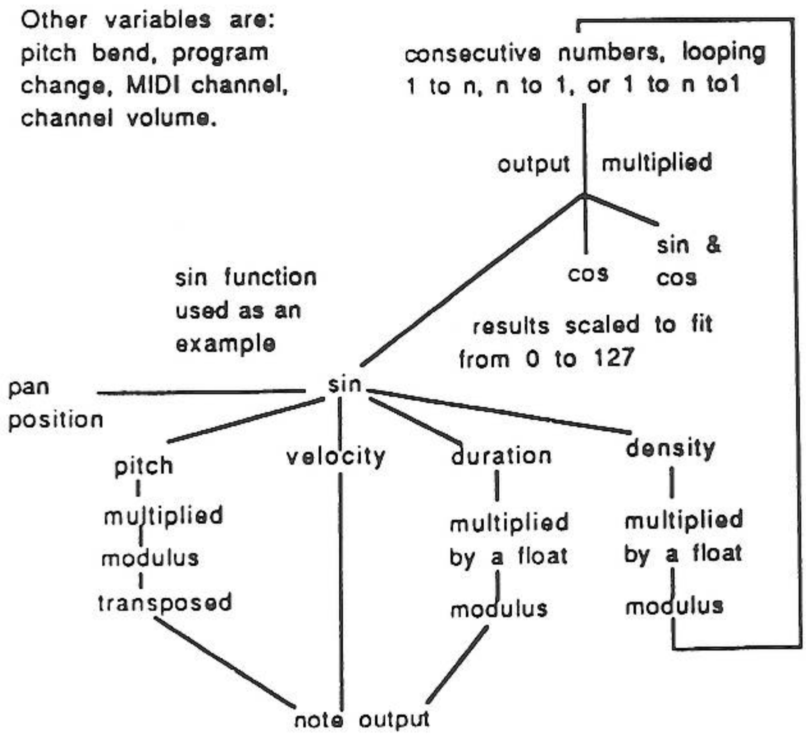
Conclusion

The problems in any interaction revolve around the purpose of the interaction and the hierarchies within that interaction. The programs above show ways of addressing this, each with a particular goal in mind. As each program is realized new or previously abscured problems arise requiring a deeper unerstanding of the interactive process.

Reference:

Robert Rowe, 1993. *Interactive Music Systems: Machine Listening and Composing*. MIT Press.

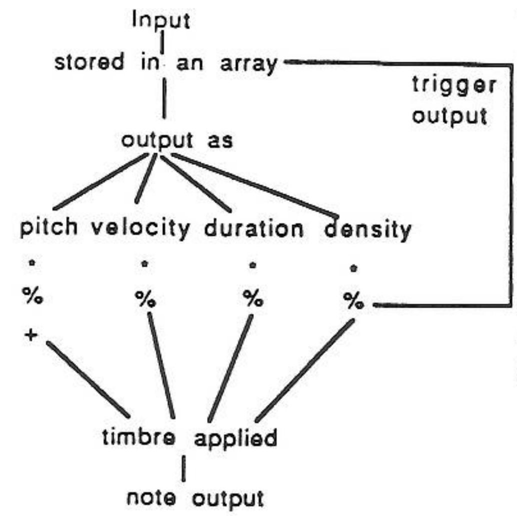
Diagram 1, MIDI guitar



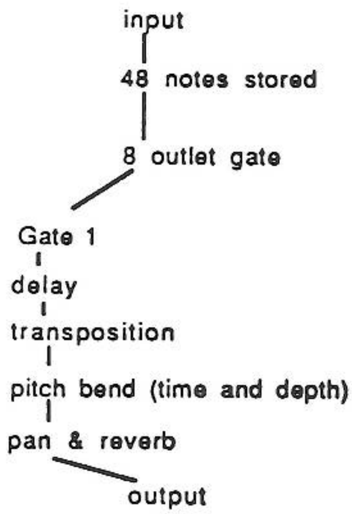
The processes that affect the sin function are the same as those that affect the cos and cos & sin funtions. The sin & cos function actually creates two opposing lines based on the same input.

This process is duplicated to produce two independent lines which can be juxtaposed as desired.

Diagram 2, "Sometimes ......



The *, % and + for each parameter are set from a constrained random field. Thes ize and area of this field is part of the pre-composition for each piece.



The size of the array, delay time, transposition and pitch bend time and depth can all be stored as presets in the precomposition. These presets can then be stepped through at set times during perrformance.

ACMA Contact List

To contact the committee, any of the persons mentioned in this issue or for any other information, Electronic mail can be sent to :

ralsop@klang.latrobe.edu.au or...
stainsby@klang.latrobe.edu.au

or fax:

(03) 479 1700 (c/- music dept. La Trobe Univ)

or write to:

ACMA, Inc.
PO Box 186
Post Office Agency
La Trobe University VIC 3083

1994 ACMA Committee

President: Alistair Riddell
E-mail: amr@farben.latrobe.edu.au

Vice-President: Lawrence Harvey
E-mail: Harvey@music.unimelb.edu.au

Treaasurer: Jane Walker
ph: (03) 801 9700 (W)

Secretary: Thomas Stainsby
E-mail: stainsby@klang.latrobe.edu.au
ph: (03) 497 4936

ACMA Sydney: Gordon Monro
E-mail: monro_g@maths.su.oz.au
ph: (02) 692 3814

AnthonyHood
E-mail: ianf@extro.ucc.su.oz.au
ph: (02) 882 8343 (W)

Dynamic Parametric Representations in Computer Music

Ross Bencina
Department of Music
La Trobe University
E-mail:rossb@klang.latrobe.edu.au

Western art music has often been criticised for it's dependence on notation as a basis for musical composition. While working directly in the sonic medium can avoid representation based compositional methodologies, paradigms of control imposed by a single synthesis / signal processing environment often limit the practical scope of parametric representation. Combining multiple representation and/or synthesis environments is a method often used to escape these limitations. Such an approach distributes control across multiple process-specific representations, the management of which often challenges the computer hacking abilities of the electroacoustic composer. The possibility of auditioning sonic materials during the composition process allows the composer to shape the musical result according to perceptual criteria. This 'shaping' is asserted through both the synthesis / signal processing methods employed and their respective control systems. It is the intent of the current discussion to examine one aspect of computer music representation; the relationship between the dominant 'note event' paradigm and dynamic parametric control specification.

Many synthesis and signal processing techniques utilise dynamic (ie. time varying) parametric specification as a means of shaping the morphology of resultant sonic objects; the simplest case being the 'envelopes' of classic synthesis techniques. In the electroacoustic domain, spectromorphological specification is often the dominant paradigm, temporal event organization being only one aspect of this. Unfortunately a large proportion of current synthesis and signal processing systems do not present a unified representational framework integrating both static and dynamic parametric specification of sound objects, those that do often utilise embedded control interfaces for specific process control.

Two common paradigms for computer music representation are MIDI and Music N style software synthesis languages. Both present different but divided representations of dynamic and static control.

MIDI makes available static parameter passing at event instantiation and termination (Note on/off velocity, midi note number), event dynamic control (polyphonic aftertouch) and channel global dynamic control (continuous controllers and sysex control commands). Due to the MIDI specification's low data throughput rate and the 7bit continuous control data format implemented by most manufacturers, severe restrictions are imposed on both the temporal resolution and magnitude of both static and dynamic control parameters.

The effect of continuous controllers on the sonic result is defined within individual MIDI sound generators, as are any stored dynamic parametric representations (envelopes). In general, the low bandwidth of the MIDI specification, and the generic global nature of dynamic control representation make it difficult to precisely control morphological attributes with the MIDI protocol. The more specific task of determining the morphology of midi triggered events is determined by operating systems and capabilities embedded in individual synthesizers. Synthesis control is conceptually separated from the control of a finite set of parameters via MIDI.

I will use Csound as an example of the Music N paradigm primarily as I have some degree of experience in it's use. The Csound score allows a large number of static parameters to be passed at event instantiation. There is no general facility for specifying event linked dynamic parameters at the score level. Dynamic parameters may be described by finite length static parametric representations, however this does not always provide a transparent interface for dynamic parametric representation. The score also allows the representation of continuously sampled 'function tables' containing abstract data whose role as dynamic control data may be expressed elsewhere in the representation. In Csound, dynamic control is generated and utilised at the 'instrument' level, it is generated by the sound synthesis processes possibly under the control of static parameters (note parameters) and/or abstract data (function tables). As with device embedded synthesis in MIDI, dynamic parameters associated with notes in Csound are primarily defined within the synthesis process rather than by the event instantiating protocol.

The computer music representations mentioned so far enforce methods of dynamic parameter specification that are counter to notions of dynamic control as a primary compositional concern. More complex approaches where the composer deals with a combination of software synthesis tools, all controlled by yet another representation (a more general process automation system). Under more primitive operating systems the composer is forced to 'hand assemble' sounds processed by multiple tools. Both approaches further diffuse the representation across multiple hierarchies of process control, 'event' and 'dynamic parametric' representation.

Specific purpose software synthesis tools often present a less divided approach to dynamic and static parametric representation. This is not surprising, as these systems are often designed to utilise dynamic control. A general class of software synthesis tools that utilise dynamic control are real-time hard disk multitrack mixing systems. Such systems include Digidesign's ProTools and Paul Lansky's "rt". While these systems utilize a number of different approaches to representing sound events and their associated dynamic parameters (amplitude, panning, pitch), they all provide access to dynamic and static control parameters at the same level of representation. Other examples of systems incorporating dynamic parametric specification as part of the event instantiation process are the UPIC system and the NeXT MusicKit.

Having identified properties of some current approaches to dynamic parametric control, some basic descriptions of their uses rather than of their implementation can be presented. Dynamic (and static) parametric representations may be classified by their scope, the scope of control specification in each case above is different. The scope of dynamic control may be unique to each event, it may be common to a family of events, or it may control parameters which are global across groups of events. An event may be an arbitrary categorisation based on computational necessity rather than formal delineation of the musical materials. Global dynamic parameters of a meta event may control generation of events at a lower level.

Following the examination of an aspect of some current computer music representations, I have formulated a list of criteria which I believe to be important in any computer music representation:

-Events could be defined hierarchically, allowing both formal delineation of material and a parametric control specification capable of representing control data for multiple layers of synthesis/reprocessing.

-Parameter representation is abstract in that the meaning of parameters is interpreted outside the representation. This is similar to one of the criticisms of MIDI above, the primary difference being that in the MIDI case, the range of possible values a continuous controller can take is narrow. Very few parameters could be directly mapped to the range 0 to 127. In the proposed representation, dynamic parameter values would have direct correlation with synthesis parameter values.

-Event definition would incorporate both static and dynamic parameter specification, with a rich set of tools for dealing with both real time and stored definitions of dynamic parameters.

-Static and dynamic parametric data should be representable as both individual data entities and as elements of note instantiation. For example, this would allow a dynamic parameter specification to be shared among a group of events, or for every event to be defined uniquely.

The representational system should be useful in dealing with all current paradigms, as well as future paradigms in dynamic parametric specification. Implemented as a scripting protocol with a set of utilities for extracting events and parameters, it could serve as a storage point for all event and parameter specification. This would enable the development of graphic tools for editing protocol conforming definitions, an area which is currently lacking in this age of advanced graphic user interfaces. After all, dynamic parametric representations are often easier to deal with graphically. In proposing a representation one must be careful to avoid attempting to include every element of the compositional methodology. Operational specifics of process automation, algorithmic composition and audio signal synthesis etc. have not been considered here, as these aspects are likely to be synthesis system / language / hardware platform dependent.

Some people seem to derive pleasure from making their computer music software jump through hoops with poorly integrated dynamic parametric specification systems. Owing to a shift in computer music aesthetics, the importance of dynamic parametric specification has increased. Representational schemes conceived thirty years ago are no longer as relevant as they once were, circumstances now demand that more attention be directed toward resolving todays computer music representation issues.

Thank You to Steve Adam for his advice during the preparation of this article.

ACMA Chronology

1989 * ACMA established as a non-profit organisation.
Graeme Gerrard, President;
Jim Sosnin, Vice-President;
David Hirst, Secretary;
Ann Shirley-Peel, Treasurer.
Interstate representatives, Peter Mumme, Victoria; Martin Wesley-Smith, NSW.
* Chroma Nos. 1, 2, 3 & 4.

1990 * 17 - 20 August, Australian New Music Conference, Brisbane; concerts and papers by ACMA members.
* Chroma Nos. 5, 6.

1991 * 10 May, AGM
Graeme Gerrard, President;
Warren Burt, Vice-President;
Michael Hewes, Secretary;
Ann Shirley-Peel, Treasurer.
* 10 October concert #1, Elm St Hall; twelve works presented, see Chroma No. 8 for details.
* 21 November concert #2, Elm Hall; twelve works presented, see Chroma No. 8 for details.
* Chroma Nos. 7, 8.

1992 * Machine Messages; CD#1 from ACMA
* 26 March, concert #1, Elm St Hall.
* July 1 -5, Australian New Music Conference, University of Melbourne; two paper sessions by

ACMA members.
* 3 July, Australian New Music Conference, Melbourne; concert at Elm st Hall.
* 4 July, Australian New Music Conference, Melbourne; concert at Melba Hall, University of Melbourne.
* 5 September, St John's Church Southbank; concert as part of the Melbourne Fringe Festival.
* 9 Nov. AGM elects committee for 1993
David Hirst, President;
Michael Hewes, Vice-President;
Thomas Stainsby, Secretary;
Andrew Brown, Treasurer.
* December, concert #2, Elm St Hall.
* Chroma Nos. 9, 10.

1993 * July 10, CompMusic, Sydney; a symposium and concerts by ACMA
* 26 June, Elm St Hall, Brigid Bourke presents music by Spanish composers for clarinet and tape.
* Novmber 29, AGM elects committee for 1994
Alistair Riddell, President;
Lawrence Harvey, Vice-President;
Thomas Stainsby, Secretary;
Jane Walker, Treasurer.
* Chroma Nos. 11, 12, 13, 14.

1994 * 5th anniversary of ACMA.
* SWAP CD #1; seven works presented in association with Princeton Studios, New York.
* Chroma Nos. 15, 16 & 17.

Index for Chroma Nos. 1-16

Number 1 May 1989

Number 2 August 1989

Numbers 3&4 Double Issue December 1989

Number 5 June 1990

Number 6 October 1990

Number 7 April 1991

Number 8 December 1991