COMPUTER MUSIC COURSES USING SUPERCOMPUTERS
AN EDUCATION EXPERIMENT AT NCSA/UIUC

Sever Tipei, Robin Barger, Ben Cox, and David Kelley
Computer Music Project
School of Music, University of Illinois
2136 Music Building
1114 W. Nevada Street, Urbana, Illinois 61801

ABSTRACT: In collaboration with the EDUCATION PROGRAM of the
NATIONAL CENTER for SUPERCOMPUTING APPLICATIONS the School
of Music of the University of Illinois has offered last year two computer music
courses: "Supercomputer-assisted Composition" and "Musical Applications On
Supercomputers". These courses showed that a state-of-the-art computer environment
is conducive to experimentation and greatly beneficial to composers as well as to the
Teaching of composition.

Besides considering grant applications for supercomputer time from individuals, the National Center
for Supercomputer Applications located at the University of Illinois at Urbana-Champaign, is
offering access to its three supercomputers (CRAY X-MP/48, CRAY 2, and the CONNECTION
MACHINES) for selected class use. NCSA's EDUCATION PROGRAM is aimed at intro-
ducing the next generation of users to one of the most sophisticated computer environments of
today. The Composition-Theory division of the School of Music was one of the first units on the
University of Illinois at Urbana-Champaign campus to take advantage of this offering.

Two courses scheduled during 1989/1990 academic year have provided students with access to the
CRAY X-MP: "Supercomputer-assisted Composition" and "Musical Applications On Supercomputers". The first one has as a prerequisite a course on "Music
Formalization" which provides contact with some of the abstract ways of thinking used in
different music composition. The course involves three meetings a week and introduces the
students to the following:

MODERN ALGEBRA (15 hrs.): Set theory, congruence relations modulo m, groups and
vector spaces as applied to music: complementary mutes, pitch classes, mappings, slice
theory, group theory in Notakiti "Nosns Alpha", musical parameters.

PROBABILITIES (11 hrs.): Probabilities, conditional probabilities, stochastic distributions
and Markov Chains; their use by Xenakis and others: patterns, The Monte-Carlo method.
Granular Theory and micro-composition.

INFORMATION THEORY (3 hrs.): Music as a communication process, information vs.
redundancy, Bifidhoffs formula.

GAME THEORY (3 hrs.): Game matrices as an alternative to aleatorism.

CATASTROPHE THEORY and COMPLEX DYNAMIC SYSTEMS (4 hrs.): Aurolo Stroe's
"Morphogenetic Music".
FRACTALS and CHAOS (3 hrs.): The Mandelbrot and Julia sets as used by Dodge, Austin and Brian Evans; non-linear systems as a source of complex deterministic organization.

SCIENCE AND MUSIC (6 hrs.): The persistence of Newtonian concepts in the arts; the possibility of mutual influences between Music and Science. Relativity and the concept of simultaneity. Quantum Mechanics, Schroedinger's cat and MANIFOLD compositions [1]. Time asymmetry — entropy and the arrow of time.

Topics discussed in the Computer-assisted Composition class (which also meets three times a week) range from a brief historic review and aesthetic considerations to a detailed overview of a program (Sver Tipel's MPI) [2,3] and also includes brief introductions to UNIX (as well as its variant AIX - for the IBM RT's of the Computer Music Project and UNIXOS - for the CRAY) and FORTRAN. Following is a summary of the course:

BRIEF HISTORICAL OVERVIEW OF CAC (3 hrs.)

AESTHETICS OF COMPUTER MUSIC (2 hrs.)

HOW DOES A COMPUTER WORK? (4 hrs.): Hardware architecture, operating systems, software, Supercomputers — optimization, vectorization, parallel processing, Object-oriented programming.

INTRO TO UNIX/AIX/UNIXOS (6 hrs.): File systems, shells, the VI editor, pipes and filters, the Common File System (CFS) on the CRAY, the NQS queueing system and the CDBX debugger.

INTRO TO FORTRAN (6 hrs.): Variables and constants, operators, declaration statements, arrays, control structures, subroutines and functions, common blocks; I/O formats, sequential and direct access files. Programming style. FORTRAN 8X.

ANATOMY OF A PROGRAM — MPI (6 hrs.): General overview, files, interactive programs preparing the data. Parts and parameters, stochastic distributions vs. specialized routines dealing with sieves and patterns.

BRIEF SURVEY OF OTHER TYPES OF CAC PROGRAMS (3 hrs.): KYMA and HMLIL.

USEFUL TOOLS (4 hrs.): David Kelley's interface for automatic music notation, ADAGIO, Robin Bargh's Music Visualization program, the Cellular Automata digital instrument.

NOVEL APPROACHES TO COMPOSITION USING CAC (2 hrs.): MPI's "Time Machine" [4], relativistic points of view in a composition [5], indeterminacy and uncertainty.

CLASS DISCUSSION OF INDIVIDUAL PROJECTS (7 hrs.)

In the future it is planned to establish a concurrent tutorial section in which students taking various computer music courses will have the opportunity to learn or brush up on their UNIX, C or FORTRAN skills.

The choice of FORTRAN is not a casual one: the FORTRAN compiler is the most robust compiler on the CRAY. It allows for optimization, vectorization, and parallel processing, and
offers the best debugging facilities on this particular supercomputer. The imminent adoption of the FORTRAN 6X standard with all its enhancements, makes it also very competitive with languages like C.

The second course, "Musical Applications On Supercomputers" is a seminar intended for people with proven programming abilities who want to pursue independent creative or research work for which supercomputers are necessary. During last year, all students taking the course enrolled for both semesters because they felt they needed more time to develop their ideas and produce something of significance. Following is a brief description of these class projects which ended up as full-fledged research topics.

1. Robin Bargar's MUSIC VISUALIZATION includes a group of interactive programs for sound synthesis and computer graphics, an environment for simultaneous composition of sounds and images. The project is based on two premises: a) decisions made in either medium can effect both graphic and sound output, and b) the implementation of the first premise can be composed. The investigation was directed toward three areas: the use of graphics to analyze and represent compositional output; graphics as a representation of compositional input; graphics as an independent expressive medium. Crossovers can be identified between the first two issues ("functional" graphics) and the third ("computer art").

To explore the visualization of musical output, a system was developed to read a computer music score and represent its parameters visually. The composer designs the visualization of a particular score, selecting from a palette of mappings of audio into the visual domain: 3D space, color, shape, and time-variant playback can be used to represent sound parameters as an n-dimensional vector. The program produces real-time animated graphic playback of the score.

The use of graphics for compositional input is exemplified by the common project of Robin Bargar and Ben Cox: it allows the user to graphically define functions or relationships between functions and generate forms on any scale, from the level of sound waves to the overall structure of a piece or group of pieces. It is hoped that this program will be useful not only to musicians but also to mathematicians, engineers, and anyone else interested in the relations between functions and their various attributes. Specific uses could be circuit analysis, calculus, and analytic geometry, among many others.

The design of interactive music representation is the first step in a broader project undertaken by Robin Bargar: the linked representation of compositional choices in both sound and image. What is wanted is a single system of "notation" that stands for both image and sound processing. The goal is to develop a method for joining sound and image that does not require completion of one medium before the other. Instead both media will be rendered from a single representation; a single set of composed instructions. This will provide an alternative to "post-production scoring" by composers within the motion picture production system, and an alternative to "reverse" post-production concepts such as "cutting on the beat".

2. MCC is a digital sound generation language designed by Ben Cox. It is an interpreter which accepts object-oriented syntax and function declarations and statements, and processes files of floating-point numbers which can be converted into an analog sound signal. It is sufficiently abstract so as to be usable for higher-level composition (such as note and musical form) as well as for producing samples. MCC includes a number of "primitive" functions, iterators, and delayed assignments. Its few language structures were chosen to maximize flexibility.

3. The AUTOMATIC MUSIC PRINTING program conceived by David Kelley is an interface
between the alphanumeric output of computer-assisted composition programs (such as MP1) and the commercially available SCORE written by Leand Smith. SCORE, a notation program, was selected from a number of alternatives because it is able to accommodate the most demanding and unconventional notations found in contemporary music scores. The interface takes in parameters describing sonic events and produces input files for SCORE which describe musical events. The program is capable of producing complex rhythmic divisions, including an arbitrary number of ties, which can be written in any metric pattern -- from 4/4 to (3+4+2)/8, for example.

4. David Kelley has also designed a CELLULAR AUTOMATA DIGITAL INSTRUMENT based on a musical paradigm whereby a group of individuals become socially organized through mutual interaction. This model has one of its roots in the Markov Chain type of algorithms where successive transformations of an initial set lead to statistical stability. Each dimension of values isstreamed to a unique harmonic of a rich tone allowing an arbitrary degree of time-variances spectra and inharmonicities. Musical events are specified at "I" cards, a collection of which constitutes a score file which is input to MAC (a C variant of 4BF, available at the Computer Music Project).

Among the three courses described here, "Music Formalization" provides the students with a minimal background in the mathematical and logical tools available to composers, it encourages them to be rigorous in their thinking and to experiment. The "(Super)computer-assisted Composition" course introduces future composers to state-of-the-art technology, presents some basic facts about hardware and software and describes how computers have been and are employed in the writing of "serious" music. Although such topics as: vectorization and parallel processing or group theory and game theory can be presented only as an introductory level, they are offered with the hope of stimulating the interest for a deeper understanding in the future. Both courses strongly discourage the students from becoming thoughtless musicians who just happen to stumble upon tricks canned for them ahead of time by the manufacturers of technological marvels. (The moronic "gore fishing" paradigm: a simplistic becomes excited at the sight of the catch of the day he/she did not expect and does not know how to cook).

A special case, the seminar in "Musical Applications On Supercomputers" quickly bloomed into an exciting forum where original ideas materialized into useful products. Although the applications dealt with in the seminar required only a short run time on the CRAY compared with scientific problems (minutes vs. dozens of hours), our experience showed how the supercomputer allows the composer to consider projects which would have been impractical on any of today's mainframes. Sadly but truly, inexpensive equipment produces sounds of the same kind while complex and sophisticated music does not come cheap. We are just starting to realize how much art, like science, could benefit from such an unique technological environment especially when it is complemented by intellectual surroundings which are equally special.

REFERENCES: