General

This is a report on work in progress at the Computer Audio Research Laboratory (CARL). CARL has existed since 1980 as the major technological research facility of the Center for Music Experiment (COME), which is an Organized Research Unit (ORU) of the University of California, San Diego (UCSD). COME is the only ORU in the 9-campus University of California system especially for research in the arts and related topics.

The development plan for CARL at its inception broke into three major phases:
1) a general-purpose timesharing computer system,
2) a dedicated realtime computer system, and
3) special purpose digital hardware for synthesis and interactive control.

Of these three phases, we have accomplished the first, have acquired the equipment for the second, and are working on the third.

CARL Facilities

CARL facilities now consist of two computer systems, a considerable amount of CARL software, and a number of user work locations, all of which are adjacent to COME facilities for analog recording, sound reinforcement, and a performance space.

The main research system is based on a VAX-11/780 timesharing computer system running the Bell Labs/berkeley UNIX operating system. The computer system is comprised of 2.25 megabytes of main memory, a floating point accelerator, an IM3 87-megabyte system disc, a TS-16 9-track magnetic tape drive, two CDC 766 300-megabyte sound storage discs, and a variety of interactive computer terminals, some of which have graphics capabilities.
A Digital Sound Corporation analog conversion system provides two channels of analog-to-digital conversion, and four channels of digital-to-analog conversion. All converters are 16-bit, capable of running at rates up to 50,000 samples per second per channel. This system is used mainly for software development, and software music synthesis, processing and analysis.

The second computer system is based on a PDP-11/55 dedicated computer system, also running a version of the UNIX operating system which allows realtime operations. This system consists of 64-kilobytes of 167-nanosecond bipolar memory, 256-kilobytes of MOS main memory, an 172-11C floating point unit, an LPA-11 analog and digital input/output controller, a DEC 9448 combination 40-megabyte Winchester and 16-megabyte removable cartridge disc subsystem, and a Cipher magnetic tape drive. This system is specially configured for realtime operations including interactive control via knobs, switches, keyboards, etc., and control of specially-built digital synthesis and processing hardware now under development.

The two sound storage disc drives are dual-ported in order to allow access by either computer system, and the two computer systems intercommunicate via RS232 and DMA connections.

**CARL Software**

CARL software, which has been distributed to a number of other centers worldwide, consists of programs for sound synthesis, sound processing, sound analysis, and sound storage (recording and playback).

The main music synthesis and processing program is named cmusic. It has been based on the classical MUSIC V program in order to minimize the amount of effort needed to learn it, but it also includes several improvements and extensions:

1. It is entirely written in the C programming language, making it relatively portable and adaptable to different programming environments.

2. cmusic's use of memory is entirely dynamic, making it well suited for execution in a timesharing environment.

3. It utilizes the C compiler's preprocessor, which implements a straightforward macro and file-include facility. A major advantage gained by using the same preprocessor for cmusic and C is that only one preprocessor need be learned and understood by CARL software users. This is a prime example of what we have dubbed "system coherence", in that anyone who has learned cmusic has already learned an important set of features of the C programming language, and vice-versa. In addition to the C preprocessor, cmusic itself provides a flexible syntax for arithmetic expressions which includes the standard arithmetic operators, transcendental and random functions, and post operators such as "+=" and "*=".

4. A rather complete set of unit generators - all coded in C - is provided for instrument design which includes arbitrary function generation and filtering, as well as extended functions such as soundfile
processing (soundfiles may be generated via synthesis or digital recording), and sophisticated spatial manipulation. The disadvantage of C-coded unit generators is, of course, that they compute more slowly than their hand-coded counterparts. However, the advantages of easier implementation (and debugging) of new and complex unit generators, together with the relative ease of recording in assembly language (for those in a hurry with a lot of time on their hands) offset the execution speed drawback very well. The spatial processing features of cmusic will be described in another talk at this conference.

In addition to cmusic, a number of sound processing programs based on the IEEE Programs for Digital Signal Processing have been implemented with interactive front-end drivers. These programs provide a variety of facilities such as sample rate conversion, Fourier analysis, computer graphics, and multichannel filtering.

A special file system for mass storage and manipulation of sample data has been developed over the last two years at CARL. Called the sound file system, it has just been released to the public, and is now also available in the CARL software distribution package. It includes the basic facilities needed to allow multiple users to simultaneously read and write contiguous sample data files on disk, and to record and play these files through the analog conversion system. It also contains several extended features, such as a realtime interactive mode for editing and comparing sound files, a "sound descriptor file" facility which allows arbitrary data to be associated with any sound file, and provision for non-contiguous as well as contiguous file storage. No modifications to the UNIX kernel are necessary to implement this file system.

The final important part of the CARL software package is its documentation. All programs are liberally documented with help files, tutorials, and UNIX Programmer's manual entries. Additional README files give instructions for the installation of programs on the host machine. All CARL software documentation exists in the form of on-line files immediately available to all users at any terminal. We expect printed wood pulp as much as possible in order to avoid documentation conflicts.

The CARL software package has been distributed for a nominal fee to more than 20 centers in the US, Canada, Australia, and Europe. The distributed software is essentially a "snapshot" of the current state of the CARL software, that is, we send only the current state of the system, no upgrades or bug reports. Since the software is free of charge, there is no warranty or support supplied by CARL for this software. It reportedly been made to run on an LSI-11/23 computer system at Queens College in Kingston, Ontario, on a PDP-11/40 in Salzburg, Austria, and of course on other VAX systems. (The sound file system may not run on anything but VAXes.) A similar sound file system was developed by Bob Gross while he was at Eastman School of Music for the PDP-11 family of computers. Our emphasis has been upon utilizing UNIX and C as a coherent structure that can be adapted directly to the tasks of musical signal processing. All programs are written in C, and no modifications to the VAX operating system have been needed to bend UNIX to the task of synthesizing, modifying, editing, recording and playing sound. We have experienced no degradation of the timesharing environment on our VAX even while running very high bandwidth realtime
Our current work focuses on
1) modeling the computer music problem based on musical rather than technological considerations,
2) developing prototype hardware which can be used in realtime to test these models,
3) carrying out the evaluation of these prototypes under musically realistic conditions, i.e., in actual performances, and
4) reducing the tested prototypes to easily reproducible forms.

In order to accomplish these goals, and with the support of the System Development Foundation of Palo Alto, California, we have undertaken a large research effort with other major US computer music and technological centers. Our principle aim is to produce a highly interactive computer music workstation over the next five years. Such a workstation ideally should have a few capabilities than any current computer music system, and cost less than a grand piano. The most important feature of such a workstation is that it should be capable of live performance, i.e., it should respond to realtime interactive control in the manner of a fine musical instrument. This work involves not only the integration of current technology, but the production of special VLSI integrated circuits for music processing.

One impact of such workstations on the field of computer music is that it would bring serious computer music activity out of the exclusive province of large research institutions. Another is that it would be possible for the first time to move from place to place and continue to work in a relatively constant environment which is currently impossible because of the lack of compatibility among major installations. Finally, it would allow the complexities of live performance to be studied in a full-blown musical context for the first time.

All of these goals are clearly worth pursuing. Whether they will be achieved is no longer a question of financial support, technological plausibility, or the availability of skilled, dedicated researchers. The problem is now conceptual: how to harness a rapidly evolving and complex technology for musical purposes. We have confidence that this problem is ripe for solution.