THE COMPOSERS’ DESKTOP PROJECT

Martin Atala, Andrew Bentley, Thomas Endrich, Rajmil Fachman, David Malham, Richard Orton, Trevor Wishart.

The Composers’ Desktop Project
11 Kilburn Road
York, YO1 4DF
ENGLAND

The origins, nature and goals of the Composers’ Desktop Project are described, along with a brief discussion about the technical problems involved in using the Atari ST as a basis for a digital sound editing system. The hardware configuration is minimal and consists of commercially available items (except for the Sounds2Steamer interface which is unique to the CDP). It makes it possible for the composer to make use of, in 16-bit digital format, a full range of electro-acoustic sound techniques: recording from analogue, computer-generated sound, signal processing, mixing and editing. Only the first steps have been taken so far. The project is designed to promote continued software development: as a cooperative research effort, extending to micro-owners the opportunity to share in the evolution of serious electro-acoustic music. Program development tools and libraries created by the CDP specifically for the Atari are made available for this purpose.

BACKGROUND

In its pioneering days, research into digital synthesis methods was undertaken, for the most part, in large American institutions on powerful mainframe computers. Considerable efforts were subsequently undertaken to make such languages portable, so that the programs could be used in alternative locations. The portability intended was however that between mainframe computers! Today new concepts of portability are emerging, and it is both conceivable and practicable to implement on some personal microcomputers software that was originally developed on machines designed for multiple users, in a large scale academic or research environment.

In the UK to date there has been no significant support for the kind of research going on in half-a-dozen institutions in the USA and at IRCAM in Paris. In the UK no mainframe computer is producing significant digitally synthesized music. Serious electro-acoustic music has been pursued largely in small analogue studios distributed throughout the country in Universities, Polytechnics and Colleges. Having no significant budgets for research, developments have been piecemeal and largely uncoordinated. Furthermore, once established, studios found themselves without significant funds for new equipment and were hence limited to buying whatever they could afford often from the commercial music world.

On the other hand, the crucial problem for students and composers has been that, on leaving the college environment, even these limited facilities became inaccessible. The only alternative, the commercial music world, offers closed 'black-box' technology or software packages tailored to the needs of multitrack rock recording and have essentially conservative musical assumptions built into them. For those lucky enough to be invited on a residency at one of the specialist research institutions outside the UK, initial enthusiasm might be denied by the realisation that experience and expertise gained would then have to be put on the shelf for a possibly indefinite period. Composers needed to have access to powerful

145 1987 ICMC Proceedings
In order to achieve our objectives a number of technical problems had to be overcome and these will be discussed more fully below. Furthermore, in order to enter into agreements with other parties to manufacture hardware for our system, and to be able to effectively administer the network of communications between individual and institutional software users and developers, the CDP has since formed itself into a non-profit making limited company which administers the CDP Club. Anyone joining the CDP Club is able to obtain the appropriate hardware for the CDP system as a special low price and to receive and to distribute software to other members. Our aim is to encourage cooperative research amongst serious computer-musicians both within and outside institutions and we are encouraged by the fact that in the UK institutional subscribers already include the Universities of Oxford, Glasgow, Edinburgh, York and Keele, whilst the Universities of Birmingham, City and Norwich have indicated their intention to participate. European subscribers (as at May 1987) include CMI Utrecht, as well as a number of individuals working at other centres of musical research.

THE CDP SOUNDSTREAMER

A primary concern of the CDP has been to keep the cost of the system as low as possible. The system philosophy is therefore to use powerful general purpose hardware (the desktop micro, rather than commercial "black-box" units which become rapidly obsolete) and to develop musical tools in software. We also needed an affordable means of storing digital sound in bulk. A small hard disk could be used by the average (low budget) user to manipulate soundfiles, but this would not be adequate to store even short pieces of music. We therefore opted for the SONY PCM system, for its combination of mass-storage of digital sound, and its affordability. It also provided a means to record digital sound for use in the Atari system and to play back the results of our composition or research.

In order to make such a system viable it was necessary to develop an interface between Atari and the PCM. This interface was designed by
David Malham of the University of York Music Dept, and is called the SoundStreamer. The unit itself is essentially a large first-in, first-out memory system with a parallel, 16-bit interface at one end and a serial interface at the other. The memory consists of 128K x 16-bit dynamic array made up from 64K x 4-bit chips. Because of the continuous, high data throughput no refreshing of the array is necessary although the controller has considerable work to do in arbitrating conflicts between read and write requests. The same memory can be used for both input and output transfers since time constraints prevent the computer handling bi-directional transfers at the required rate.

The choice of the PCM dictated a sampling rate of 44.1 KHz in stereo (though the CDP system also allows one to work in mono and at the subsampling rate of 22.05KHz). At this sampling rate an overall data transfer rate of 176 kilobytes per second is required. This poses a formidable problem on a personal computer such as the Atari ST. Since we do not wish to modify the Atari, we were limited to a choice of two ports, the DMA port and the ROM cartridge port. The DMA port is intended to couple up to a hard disk. The communications protocol used, AHDI, is rather, but not quite, like SCSI (Small Computer Systems Interface), which is an industry standard. Naturally, the differences arose in the worst possible place for our needs! After carefully examining all the various timings and transfer rates it became apparent that any performance gain from utilising the DMA was outweighed by the difficulties involved in implementing a connection to the AHDI bus.

The ROM port has a direct connection to the ST's 68000 processor. The sixteen bit data bus and the sixteen low order address lines are brought out as well as two strobes which select either the upper or lower 64K of the 128K of addressing space which it occupies. Missing, however, is a read/write control line and, indeed, any attempt to write data to the cartridge area results in a bus error being generated, at which point the machine may well crash. This problem, which at first seemed insuperable, was neatly solved by Martin Atkins of the University of York Computer Science Department, who has since joined the CDP as a software consultant. The technique, known as data offset addressing, involves adding the data word to a base address and then using a dummy read of the address generating so that the data is contained in the lower sixteen address bits.

With suitably efficient software this technique has remarkably little overhead when compared to a normal write data operation. We have measured data output rates of 384 kilowords per second. Since we have to supply data to the PCM at 176 kilobytes per second, we have about 77% of the available time spare within which to communicate with the disk drive.

Using the Atari software as supplied this is sufficient but the margins are certainly not enormous as the machine waits until disk operations complete before carrying on with any processes.

THE SOUNDFILE SYSTEM AND RESIDENT LIBRARY

In the CDP system, processing of sound data is normally achieved by operating on soundfiles stored on a hard disk. Atari's own hard disk system may be used for this, but we are recommending the use of SCSI disks, giving the user a wider choice of sizes and prices of hard disk.

Handling transfer of sampled data to and from the hard disk involved writing a new, specialised soundfile handler to get around certain problems in the existing operating system. In GEMDOS, parts of files are scattered all over the disk, wherever there is spare space available. Although it is well documented that properly designed scatter storage has a number of advantages for digital audio, this can only really by applied to the large, fast disks used in big systems. With the lower performance drives which are all that many of the potential members of the project could afford, a file structure is needed where sectors are written consecutively on the disk in order to reduce the amount of time involved in seeking for the required sectors and hence gain a sufficiently high average transfer rate.

The CDP soundfile system was written by Martin Atkins and stores sound in consecutive
sectors on the hard disk. The soundfile system is accessed through its own library of functions and, to the user, appears similar to the resident generic system. Functions to create, open, close, delete and rename soundfiles are provided, together with a comprehensive error-handling system. When creating a soundfile, its length must be specified. Alternatively a flag may be set which causes the system to grab the largest available area of consecutive sectors on the disk. After this has been utilized, the soundfile will be appropriately truncated to free any unused space. Once created, soundfiles cannot be expanded in size. Available space on the disk can be optimized using a SQUEEZE program which moves empty blocks to the end of the disk.

Data about the soundfile is stored in a header block at the beginning of the file in a linked list of properties each consisting of a property name (an ascii string) and the associated value. Functions are provided to put, get, create, delete, and list soundfile properties. For signal processing applications, functions are provided to permit sample-by-sample and buffered access to the soundfiles, similar to the CARL procem library.

During the development of the system, the soundfile system was revised a number of times. As soundfile functions are fundamental to any signal processing application this meant that such applications had to be rewritten or recompiled with each such revision. We foresaw that once the system became public, if we decided to make further revisions to the soundfile system we would need to revise and re-release all existing software. Similarly, all other program developers using the system would need to similarly update all their software. This was clearly unsatisfactory. To solve this problem Martin Atkins proposed, and subsequently implemented, a resident library structure for the soundfile system. This resident library will normally be loaded on system boot-up. Resident library functions are linked into existing applications through an indexed array of pointers to functions in a protected area of memory. A revision to the soundfilling system now implies only that a new version of the resident library be installed by the program developer. In all other respects applications

programs will remain unchanged.

PROGRAM DEVELOPMENT ENVIRONMENT AND USER INTERFACE

During the porting of Cmusic from the CARL package of software, some important problems needed to be overcome and these resulted in the development of a practical program development environment on the Atari. Thus Cmusic on the VAX works in a UNIX environment, utilizing pipes, and calling existing unix functions such as SORT. A number of unix-like utilities were written for the Atari (sort, gzip, cat etc), together with a system call, to enable one program to be called by another, hence bypassing the need for unix pipes. At the same time a MAKE utility proved essential to keep track of such a large program development task, and this was also written by Martin Atkins.

In order to make the system easily accessible to the musician user, a mouse-driven graphic interface using drop-down menus and dialogue boxes, and known as the CDI Desktop has been developed by Rajni Fischman. Commonly used functions such as Play, Record, Delete and Copy soundfiles etc are accessed via the drop-down menus. Other applications, including those developed by the user, may be accessed through a directory dialogue-box, and implemented via a command-line dialogue-box. This structure permits the software base to be expanded by individual users, and to remain compatible between different users without constantly needing to rewrite the graphic interface.

The flexibility of the Desktop System can be seen in the variety of ways in which sounds can be brought into the system: by analogue input (microphone or tape), by recording from video tape to hard disk, by computer-generated sound synthesis, and by sound input via MIDI.

Current program development has centred around the C programming language, due to its universal adoption by the international computer-music community. In the near future, however, the newly favoured LISP, and hence Midl-Lisp, will be available on the Atari. We
are also interested in attaching a floating-point co-processor or transputer to our system, in order to speed up calculation times (a transputer add-on for the Atari already exists) and anticipate that some of our members will already be working on this!

The CDP offers a program development kit to members wishing to develop their own music applications software and Andrew Bentley of the CDP has already developed applications for mixing, editing, speed-changing and filtering sounds.

On joining the CDP club, members receive copies of these application programs, together with Csound and the CDP Desktop environment, and during the first year we intend to release further modules from the CARL package, together with Csound (the new version of Barry Vercoe’s Music-11 written in C) and the CMU Midi Toolkit.