MASSACHUSETTS INSTITUTE OF TECHNOLOGY
A. I. LABORATORY

Artificial Intelligence
Memo No. 243

October 1971

USING THE EUTERPE MUSIC SYSTEM

Stephen W. Smoliar

Work reported herein was conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract Number N00014-70-A-0362-0002.

INTRODUCTION

This memo describes the practical implementation of programs written in the language EUTERPE. Details of this language are given in the author's thesis ("A Parallel Processing Model of Musical Structures") and will not be treated here. We shall only be concerned with the preparation and processing of a EUTERPE source program. Sample programs are given in their entirety in the thesis or may be read off the author's file directory (SWS;). Notational conventions are those of Dowson's guide to the A. I. Lab timesharing system (A. I. Memo No. 215).

The EUTERPE system consists, essentially, in five stages of operation. These are, in summary, as follows:

1) Preparation of a source file. This is basically the "EUTERPE program," i.e. the representation of a piece of music in the EUTERPE language.

2) Assembly. In order to be run, a EUTERPE program must be assembled into a binary file which is actually loaded into the computer.

3) Debugging. A EUTERPE binary is normally loaded into the PDP-6; and as it is running, this computer plays the music represented by this program. The execution of the program involves a real-time interpretation of EUTERPE's six voice programs which is monitored by DDT, a debugging facility. (This DDT is not to be confused with the DDT which monitors the ITS timesharing system; it is a special DDT which runs only in the PDP-6.) This DDT has facilities to aid the debugging of EUTERPE programs.

4) Compiling. Because this binary file runs interpreted, the sound which is produced has certain distortions due to the timing of the intepreter. These may be overcome by preparing a compiled version once the program is debugged.

5) Playing compiled music. The PDP-6 is again the "instrument;" however, the monitor for this player runs in the timesharing system. It is also compatible with the music compiled by Peter Samson's MUSCOM (A. I. Memo No. 107).

PREPARATION OF A SOURCE FILE

A EUTERPE source program should be written as a TECO file which will then be assembled by MIDAS. In order for the program to be properly processed it is necessary to assemble the interpreter for EUTERPE instructions along with the source program. This is achieved by beginning the source file with the command:

```
.INSRT EUTERP;EUTERP >
```

This will supply the latest working version of EUTERPE. (Recent not-necessarily-working versions will have user name SWS.) Programs for the six voices may be inserted following this command. Since a program for voice n is to begin at address VOICEn, this program should be preceded by the MIDAS pseudo-operation LOC VOICEn. For example, here is the beginning of a EUTERPE program:

```
TITLE MODULIERENDER ZIRKELCANON
.INSRT EUTERP;EUTERP >
LOC VOICE1      ;CANTUS FIRMUS
        WAVE ALL,SQUARE ;ALL VOICES SOUND SQUARE WAVES
        MOVEI 1,6       ;ACCUMULATOR COUNTS SIX REPETITIONS
        K C 4D          ;NOTE WORDS FOR CANTUS FIRMUS
        K D 8T
```

Finally, the file should end with the command:

```
END TUNE
```

ASSEMBLY

The actual assembly of a EUTERPE source file is accomplished by MIDAS in the same manner as the assembly of an ordinary assembly-language file. Persons unfamiliar with MIDAS may easily assemble their programs by typing the following commands from DDT:

MIDAS<u>ctrl</u>K
filename1_filename2cr

This will create a binary file called filename1 BIN which may either be run interpreted on the PDP-6 or run in ITS to produce a compiled file.

MIDAS error messages are listed in Peter Samson's MIDAS memo (A. I. Memo No. 90); however, there is one error for which no message is given. If a EUTERPE program is too long, there may not be enough room for it in the PDP-6. When MIDAS is completed, it gives the addresses of the core assigned to generated constants. If this area extends higher than 30000, there is a danger that it will conflict with DDT; and the program should be broken into smaller segments.

DEBUGGING

The interpreted execution of a EUTERPE binary plays music using two 9-bit d-to-a converters. It is loaded into the PDP-6, along with a DDT, by TENLOD. By way of example, the EUTERPE binary file filename1 BIN would be loaded by typing the following to the timesharing DDT:

```
TENLODctrlK
L filename1_BINcr
F
```

The last command actually starts the DDT in the PDP-6 running. TENLOD may type out one of two critical error messages. "PDP-10 NOT AVAILABLE" means that some other user has the PDP-6. "RUNNING?" may be typed in response to the F command. This may or may not mean the PDP-6 is not running. Check the RUN light on the PDP-6 control panel (below the microtape drives; it is labeled RUN). If it is on, TENLOD is probably confused. If it is off, depress the INSTRUCTION STOP switch (and restore to center position) on the PDP-6 control panel; and do the same for the IO RESET switch. Set the address switches to 34000, and depress the START switch. DDT should now be running in the PDP-6.

The best way to communicate with the DDT in the PDP-6 is to use T4, the teletype at the PDP-6 console. On the left side of the 340 display is a toggle switch in the up position. Depressing this switch causes this teletype to communicate directly with the PDP-6, rather than with ITS. When this switch is in the down position, the user is ready to type commands at the DDT in the PDP-6.

Unfortunately, there is no available documentation for the PDP-6 DDT. The novice would do well to consult the section on DDT in DEC's PDP10 Reference Handbook; most of the basic debugging facilities mentioned therein are identical to those of PDP-6 DDT. Also, Tom Knight's A. I. Memo No. 147 lists most of the DDT instructions but also includes commands for the timesharing monitor. (N.B. For purposes of using PDP-6 DDT, A. I. Memo No. 147 is far preferable to its revision, A. I. Memo No. 147A; get it before it disappears.)

Before running a EUTERPE program, the amplifier and data switches have to be set for monitoring the d-to-a converters. Turn the amplifier on and set it to channel AUX2 with relatively high volume. It is desirable to lower the treble and boost the bass because of the rather nasty sound which these d-to-a's currently make. Finally, the address switches should be set to 20.

To start the EUTERPE program playing, type TUNE (ESC) G; the program will now play to its completion, after which control will be returned to DDT. On subsequent occasions, the player may be started by typing SETUP (ESC) G. The player may be stopped by typing a space.

EUTERPE has eight breakpoints which cause a temporary interruption of the program. To enable these breakpoints, type EUBRK(AA) (ESC) B. To assign an address to a breakpoint, type BREAKS+n/, where n ranges from 0 to 7. Then type the address which is to be the location of the break. When any voice program encounters that address, the playing will stop; and control will return to DDT, which will type out the address of the break. To resume playing from that breakpoint, type (ESC) P.

Some bugs are more fatal than others. Some, for example, may prevent returning control to DDT. DDT may be restarted from the PDP-6 console by hitting INSTRUCTION STOP, IO RESET, setting the address switches to 34000, and hitting START. (Don't forget that the address switches must be set to 20 for listening to music.) If this succeeds, the teletype will respond with a reassuring carriage return and line feed. If it fails, DDT has been clobbered and it is necessary to reload the PDP-6. Flip the teletype toggle switch up again, and type ctrlX. This

restores TENLOD's attention, and you can proceed exactly as if you had typed TENLODctrlK. (Short cut: if you just type L cr, TENLOD will load the same file it loaded on the last L command.)

If you do not have access to T4, the situation is not hopeless, although it is much more painful. You will have to be in the vicinity of the PDP-6, not only so that you may hear the loudspeakers, but also to be able to readily access the PDP-6 console. From TENLOD first type 10 (this is the letter O); TENLOD will respond with a carriage return and line feed, after which you type an L command as above. Then type PF instead of F, and your teletype is now communicating with the DDT running in the PDP-6. Things may now proceed exactly as above, except that hitting a space will not interrupt the PDP-6 playing music. To do this, you have to stop the computer from its console and restart at 34000 (following the same procedure given above in case the program hangs up). To get back to TENLOD, type ctrlX; and to return to the PDP-6 DDT from TENLOD, type PS.

Once you have become accustomed to debugging EUTERPE programs, you will probably have occasions when you will not have to actually hear the program to debug it. In such a case, the program does not have to be run on the PDP-6 but can be run as a job on the timesharing system. Suppose you have such a program assembled with file name filename1 BIN. From

(timesharing) DDT, create a job by typing:

jobname (ESC) J

Next, load the file by typing:

(ESC) L filename1_BINcr

The program may now be started by typing NOSIX (ESC) G, and it will type out .VALUE 0 when it stops. On subsequent occasions, it may be started by typing SETUP (ESC) G.

It is also possible to play a EUTERPE program through the Minsky music box hooked into a terminal which outputs 30 healthy characters per second. (Noise in either the terminal or the line may cause disasterous lossage.) This output rate will, unfortunately, distort most rhythmic subtleties; but it will provide a rough approximation for debugging purposes. To run a program in this manner, create a job and load the file, as above, and start the program by typing BOX (ESC) G. The appropriate control characters are transmitted for turning the box on and off and silencing and restoring typeout; and these control signals alow for breakpoints in the EUTERPE program. Any other interrputions, however, may lose in any number of ways. In this case, the user may have to supply control chaacters from the

terminal:

lower-case @: box and typeout on

lower-case a: box on, typeout off

lower-case b: box off, typeout on

COMPILING

Once one has a debugged version of the program, one may compile a much more euphonic performance. This is done in HACTRN (rather than in the PDP-6) by creating a job in ITS, loading the binary of this debugged version, and starting at the address COMPIL (i.e. typing COMPIL$ G). The program will echo with an asterisk, whereupon the user prescribes conditions for output -- device name, user name, file names. These have the following default values: DSK:user name;MUSIC filename1.

While it is not recomended practice, it is possible to take a debuged EUTERPE program directly from the core of the PDP-6 and compile it. Starting in PDP-6 DDT, return to TENLOD, type ctrlZ, and then type PDP6$ J. Type 33776/ followed by a ctrlY. Dump out this core image using$ Y. Now proceed as above, loading in this dumped file instead of an assembled binary. (Any file which is already loaded in the timesharing system, e.g. programs using the Minsky box for debugging, may also be started at COMPIL and need to be dumped only if the user wishes to save them.)

PLAYING

Compiled music is played by APOLLO which is loaded by typing APOLLOctrlK. When it is loaded, APOLLO echoes an asterisk unless the PDP-6 is not available, in which case it flushes itself. APOLLO commands are analogous to those for MUSRUN and are as follows:

L is the command for Loading files. When it is typed, APOLLO echoes with a space, after which the user may type a file name. If the file is not found, APOLLO replies with a question mark. Default values are initially DSK:EUTERP;MUSIC OUTPUT, but once a file has been loaded, its specification determines default values. However, if only one file name is given, it is taken as the second; and the first is assumed to be MUSIC. APOLLO will not only load files compiled by EUTERPE, but also those compiled by MUSCOM which were originally intended to be loaded by MUSRUN. If the file is completely foreign to APOLLO, it will type "CAN'T FIND DATA" followed by another asterisk. If the PDP-6 is not running, APOLLO will type a message to that effect. If this is the case, hit INSTRUCTION STOP followed by IO RESET, set the address switches to 100, and hit START.

If a compiled file is too large to fit in the PDP-6 core, APOLLO will load as much as possible and then type "MORE."

To continue after the loaded portion is played, type A. If there is still too much for one load, another MORE message will be given, and so on until all of the file has been digested.

P is the Play command. When it is typed, the PDP-6 starts playing. For this player, the address switches and data switches should be set to zero and the amplifier should be set to channel AUX1.

S is the Stop command. Typing it will cause the PDP-6 to stop playing.

Q is the Quit command. Typing it will flush APOLLO.

T is the Tempo command. When it is typed, APOLLO echoes with an equals sign and the tempo constant. If it is preceded by a number, then the tempo constant is set to that number. (Do not type a space between the number and the T.) Files compiled by EUTERPE have the tempo constant 434.

D is the Detune command. It is preceded by a six digit octal number which sets the detuning constants for the six voices (see A. I. Memo No. 107).

? causes APOLLO to list these commands.

The APOLLO player may also be run from the PDP-6 data switches in the manner described in A. I. Memo No. 107.