Interface, Vol. 6 (1977), pp. 57-72.

A Composer's Introduction to Computer Music*

by

William A. S. BUXTON

ABSTRACT

A historical survey of computer music is presented. As the title suggests, this survey is written for the composer, in order to facilitate his access to the tools currently offered by technology. Approaches both to computer composition and to sound synthesis techniques are considered. These are discussed in terms of their music-theoretical implications, modes of man-machine communication, and hardware configurations. Under computer composition, the various degrees to which a digital computer can participate in the compositional process are discussed. As regards sound synthesis, both digital and hybrid techniques are considered. Throughout, the presentation is based on a discussion of specific systems, which provide a historical review of the field. In addition, extensive references are made to the existing literature, in order to direct the reader to additional information.

INTRODUCTION

Since the appearance of Hiller and Isaacson's *ILLIAC Suite for String Quartet* (1957), computers have played an important role in the realization of many musical compositions; nevertheless, this area of contemporary musical life is still treated with a widespread lack of understanding. Consequently, this paper intends to provide a general introduction to computers as they relate to the production of music. The approach taken is that of a general overview. Our goal is to present the conceptual and theoretical background which would enable the reader to evaluate and compare the various systems extant, and provide the basis for further discussion on the subject. It is hoped that this will result in the promotion of discussion and opinion based on information, rather than on misconception, as is too often the case now.

* Revised version of "Computers and the Composer: an Introductory Survey," published in FAIRE 4/5, Editions GMEB, 1975.

THE COMPUTER AS TOOL

Before beginning to discuss musical applications per se, it would be worthwhile to investigate briefly those attributes of computers which have made them attractive to musicians in the first place. Digital computers may be characterized by their two main properties: the ability to accurately remember (i.e., store) large quantities of information for an indefinite period, and the ability to perform a sequence of — albeit simple — instructions in a very short period of time. Differences between various computers relate to these two attributes; that is, the amount of information which can be stored, and the nature and speed of the instructions possible. Each of these two processes is centered in a reasonably autonomous part of the computer: the MEMORY and the CENTRAL PROCESSING UNIT (CPU), as diagrammed in figure 1.

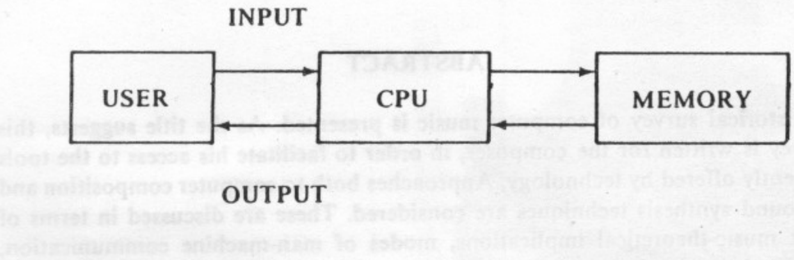

Fig. 1.

Notice in the diagram that all communication between the outside world (the USER) and the memory takes place via the CPU. Thus, the CPU takes on the role of the "nerve center" of the device, supervising the movement of all information.

In the above discussion, it is intentional that nothing has been stated concerning the paths labelled "INPUT" and "OUTPUT" in the diagram. In the conceptualization which we are trying to establish, the computer should be considered as a modular device which is "plugged into" a particular application. The size and shape of this "module" (i.e., the computer), along with the languages and devices used for input and output, are then considered as properties of a particular application rather than of computers in general. While this seems a trivial distinction, it becomes increasingly important to keep separate the concepts of "device" and "task," for the sake of both design and evaluation. We will therefore discuss languages and peripheral input and output devices later in the paper, in terms of their appropriateness to particular applications.

In summary, it is clear that the computer can benefit the composer only insofar as one is able to define tasks or applications which can be served by a computer's memory and/or ability to execute a series of predetermined instructions (i.e., a PROGRAM) As we shall see, such applications range from the automation of all or part of the compositional process, to the acoustical (viz., sonic) realization of a musical composition.

MUSIC SYSTEMS IN GENERAL

The multiplicity of approaches to "computer music" are such that the potential composer-user is frequently overwhelmed by their diversity. Thus, in order to impose some order on our presentation, we shall commence by establishing certain criteria whereby various systems can be compared. To begin with, our discussion will present the material in terms of two main application areas: the use of the computer in the compositional process, and the generation of acoustical signals. The reader should be aware, however, of the bias implicit in this separation of abstract musical structures on the one hand, and sound on the other. This is a bias which is neither reflected in all of the systems to be discussed, nor is entirely justifiable in terms of music theory. Keeping these misgivings in mind, we use this approach for ease of presentation.

In addition to the above mentioned separation of topics, three other considerations should be introduced in order to facilitate our discussion. These are:

1) What is the theoretical basis or "model" on which the system is founded (implicitly or explicitly), and what are the resulting musical assumptions or restrictions imposed on the user?

2) What is the hardware configuration on which the system is implemented; that is, what equipment is used and how is it set up?

3) What is the mode of man — machine communication; that is, how do the composer and the system interact?

While these criteria are neither mutually exclusive nor all encompassing, they do provide a basis for comparison among systems of interest. We may now proceed to discuss these systems according to the two application areas mentioned above: computers and the compositional process, and computer aided sound generation.

COMPUTERS AND COMPOSITION

Historically, there have been two main trends in the use of computers in the compositional process. These can be characterized as those programs which on being initialized, would generate "musical" structures without further intervention by the composer (composing programs), and those which serve as "aids" to the composer in carrying out lower level compositional tasks (computer aided composition). Since each approach gives rise to interesting peculiarities, we will deal with them separately.

1) *Composing Programs*

Much of the initial use of computers for musical purposes was in the writing of programs which, on being initialized with the appropriate data, would generate a completed musical structure. Early examples of such usage which are of historical importance include: Hiller and Isaacson's work at Illinois (Hiller and Isaacson, 1958, 1959 and Hiller, 1959) which resulted in the *ILLIAC Suite for String Quartet* (1957); the ST programs of Xenakis (Xenakis, 1971), from which the composition *Atrees* (1962) was produced; and Koenig's PROJECTS 1 & 2 which resulted in, for example, *Uebung für Klavier* (1969).

In each of these systems, it is the embodied model of the compositional process which is of prime importance. In order to be implemented, each demanded that the author first formalize, and then program the "rules" of a particular theory of composition. This is true even if the author is unaware of it. *Every* program for composition embodies a specific set of such rules. Thus, it is the nature of this "theory" and its implications for the user which we shall investigate. While these are the only three systems we will deal with under this heading, it must be realized that many other models have been proposed or implemented for the generation of musical structures. These include models based on linguistics, cybernetics, systems theory and so on. See for example (Clough, 1969), MUSICBOX (Wiggen, 1972), (Moorer, 1972), and MUSCOMP (Rader, 1977).

The goal in the early experiments of Hiller and Isaacson was to have the computer undertake the composition of quasi-traditional counterpoint. The results of their first four experiments constitute the movements of the *ILLIAC Suite for String Quartet* (1957). The first of these experiments involved the generation of simple diatonic melodies as well as of two and four part polyphony. In the second experiment, four par' first species counterpoint was produced. A more modern idiom was chosen for the third experiment. Here, chromatic music based on tone rows was produced. Finally, the fourth experiment involved the production of "Markovian" music, that is, music where the notes are generated randomly, but where the probability of any particular note being chosen is dependant on the last note(s) selected.

Throughout these experiments, the basic technique or "model" used is what is known variously as the "Monte Carlo" technique, a "finite state" machine, "table driven" generation, "generate and test", or "information theoretical" model. Regardless of terminology, what is meant is quite simple. It can be described in terms of three basic steps: initialization, generation and testing. To begin with, the user of this technique must set up a table of "rules" or "conditions" which define which combinations of notes are considered "legal". This constitutes the initialization. Thus, in experiment two, the rules for voice leading, etc. for first species counterpoint were specified. Once these "rule tables" have been initialized, a composition can be begun. The process is as follows: a note is generated at random (the generate step). This note is then tested for acceptability against the "rules" which were specified in the initialization phase (the test step). If it is accepted, it is appended onto the score. Otherwise, a new attempt is made to generate an acceptable note. Thus, via repeated iterations through the generate and test procedures, a composition is gradually built up.

In dealing with the "generate and test" technique, there are certain significant points to be considered. First, it is important to note that the nature of the rules, which is of prime interest from a musical point of view, is completely arbitrary from a technical viewpoint. Thus, different stylistic traits, for example, can be generated simply by having the composer define his own set of rules. This is not, however, as useful a property as might be at first imagined. To begin with, this technique is "left to right". That is, a composition is "through-composed" from start to finish. As a result, there are severe stylistic limitations on the types of musical structures which can be generated. In addition, changing the "rule table" is a non-trivial endeavour, which significantly limits the composer's freedom.

Xenakis refers to his computer generated compositions as "stochastic" music. In realizing these works he makes heavy use of the science of probability and statistics. It is from this field that the term "stochastic" derives. Generally described, stochastic music implies simply that random variables, selected according to certain probabilities, are utilized in the calculation of a musical structure. In order to get a better feeling for how such calculations function in the ST programs, it would be worthwhile to investigate briefly Xenakis's ideas on the perception of musical structures. These ideas center on the concept that what is of highest musical importance in such structures are the composite "groups" of sounds, rather than the individual sound events. Thus, each "group" of sounds which is perceived as a structural entity can be thought of as a sound "cloud". The speed, colour, density, and shape of this "cloud" then give a means of characterizing the group as a whole. This is preferable to having to describe the cloud note-by-note via its constituent elements.

In adopting a theoretical basis in which the isolated event is secondary to the group, Xenakis then strove to evolve a meta-language for music which reflected this approach. Given his ideas about "clouds" of sound, it is not surprising that he turned to statistics and probability, which are well suited for such description. The essence of the ST program, therefore, is that it enables the user to describe the characteristics of the clouds of sound in a musical structure, using the terminology of statistics and probability. The program then uses "stochastic" procedures to calculate the elements of these clouds according to the user's specifications. The musical implications for the prospective user, therefore, are that he must accept Xenakis' formalization concerning "clouds" of sound, and be prepared to specify his ideas in terms "understandable" by the program. While there are definite limitations imposed on the composer by Xenakis' system, it is one of the few which has resulted in tangible musical results.

The work of Koenig, as illustrated by the programs PROJECT 1 and PROJECT 2, is based on an extension of serial technique which was prevalent in the 1950s. PROJECT 1 (1964) leaves little room for influence by the user. Basically, the same process generates each piece, with only random variation. The program outputs information for manual transcription concerning the following parameters: timbre, rhythm, pitch class, octave register, and dynamics. Each composition thus generated consists of seven "form-sections." The central idea behind the program is a play between the "periodicity" and "aperiodicity" over each parameter. In terms of PROJECT 1, periodicity implies a sequence of similar values while aperiodicity means dissimilar values. For each parameter there is a scale of seven levels of periodicity. Thus for any particular parameter, each "form-section" has a different degree of periodicity (i.e., one scale degree for each form-section). The sequence in which the degrees of each parameter's scale appear in the "form-section" is random and may be different for each parameter. Thus, PROJECT 1 can be seen as a program which generates compositions according to a very narrowly defined compositional model. Since it was written for his personal use, this is not a drawback as long as the idea works musically, which it does, in this author's opinion. The biggest problem in this approach, however, is the investment required to produce a program with such limitations.

Based on his experience with PROJECT 1, Koenig attempted to write a compositional program which would be of general application. The result was PROJECT 2. Basically, the attempt in PROJECT 2 is to enable the user to specify the compositional rules whereby each of the various parameter values are selected throughout the piece. Principles such as aleatoric, series, ratio (weighted alea) and tendency are available, for example. As a result of this increased flexibility, however, the user is confronted with the somewhat formidable task of understanding the framework of the program in which his input data functions. As well, he must format this data in the appropriate manner. Once done, however, the program is able to produce compositions of quite diverse natures. Currently an interactive version of the program is being developed. With it, PROJECT 2 shall not only become more accessible to composers, but probably fulfil its promise as a tool for research into problems in computer composition.

While some compositions of interest and musical merit have been produced by composing programs such as those discussed, certain questions do arise. The prime one is this: given that decision making is undertaken by the program, to what extent do we posess sufficient knowledge of the musical processes involved to program the knowledge base on which these decisions are made? Each of the projects mentioned represents an attempt to deal with this problem. These efforts have brought to light several previous misconceptions concerning music, just as attempts at automated speech translation did in linguistics. The central issue was the inadequacy of traditional music theory to deal with the "musical process." That is to say, we are severely limited in our current ability to establish a knowledge base for computerized musical decision making. Consequently, those systems which have had musical success, such as those of Xenakis and Koenig, have of necessity been highly specialized in that aspect of music with which they dealt, and therefore have been highly personalized. The writing of a "generalized" system for the composition of music would presume a complete understanding of a "grammar" for music; however, it is doubtful that such an understanding can exist. Thus, while programs such as PROJECT 2 are valuable in exploring a particular theory, at this stage they cannot, by their very nature, be of general application.

In terms of hardware, each of the above mentioned systems was initially implemented on a large-scale computer (the Xenakis program, for example, on an IBM 7090). Man-machine interaction involved the preparation of the initial input data, and collection of the final results; there being no composer intervention during the actual realization of a composition due to the automated nature of the programs. In these early systems, the completed composition was output by the computer in the form of alphanumeric symbols. This encoded version would then be manually transcribed into common musical notation (CMN) for performance by traditional musical instruments. It is clear, however, that given appropriate facilities, the musical data could have been output directly in the form of CMN. This could be done without affecting the compositional aspects of the program, while significantly improving the man-machine interface. An example of such a program has in fact been written (Byrd, 1974), which automatically transcribes data produced by Xenakis' program. In addition, it is clear that man-machine communication would be further enhanced if the output of compositional programs could be in the form of an acoustic realization of the completed work. This is supported and demonstrated in the following discussion of computer aided composition and sound synthesis techniques.

2) *Computer Aided Composition*
The above discussion has brought to light two main problems concerning composing programs. First, it was illustrated that the more knowledge and power that is built into a program, the less general its musical application (see also Truax, in press). Second, attention was drawn to the limitations in our ability to formalize a basis for musical decision making. As a result of these limitations, in many systems an alternative to composing programs has been taken. This we will call "computer aided composition." We acknowledge that in the general sense, this term could cover the use of any computer system (from sound synthesis to composing programs) in the creation of music; however, for the purpose of this survey a more limited scope is intended.

The key feature distinguishing computer aided composition from composing programs is the degree of interaction between the composer and the program during the realization of a composition. In brief, computer aided composition implies only limited decision making on the part of the computer, which is subject to the composer's intervention and control. Such intervention and control takes the form of a dialogue between the composer and the program, and its nature is extremely important in the evaluation of such systems.

One approach to computer aided composition is illustrated by the SCORE program developed at Stanford University (Smith, 1972). SCORE is primarily a program which enables a user to input, in music oriented terms, the pitch and rhythmic data to a sound synthesis program. The effect, therefore, is to render the technology more accessible to the musician. The user may not only create motives, but easily transpose or otherwise transform them. As well, he may introduce various degrees of randomness over note sequences. All this is accomplished using an easily learned (for musicians) alphanumeric command language. One of the drawbacks with SCORE, however, is that while the specification of the data to the program is interactive, its acoustic realization is not necessarily so. As was stated above, SCORE is a program to input data to a sound synthesis program; however, the type of synthesis program generally used with SCORE is of the MUSIC V type (see discussion of digital synthesis, below). Unfortunately, programs such as MUSIC V do not easily lend themselves to interactive sound synthesis.

In terms of compositional power, the active role taken by the SCORE program is quite minimal. It is primarily a tool of convenience which has proven its value in actual practice. Its compositional power can be augmented, however, as can that of most sound synthesis programs of the MUSIC V type. This is accomplished by combining the basic program with special compositional programs or "subroutines". The use of such programs to generate parts of a composition has been described by Howe (Howe, 1975). Typically, the composer would write such subroutines himself, to generate certain of the musical data for a composition. The problem is that the composer is then generally obligated to learn computer programming — a not altogether musical endeavour. Furthermore, it is seen that this approach to composition has more in common with composing programs than with interactive computer aided composition, as described below. Such an approach to computer composition has proven useful to many composers, such as Howe; nevertheless, for the composer just beginning to utilize computers, it is important to realize that alternatives do exist.

As we saw, one of the key drawbacks of the SCORE system was that it did not necessarily enable the user to interactively audition all or part of the composition in progress. In recent years many researchers have been developing systems which, to varying degrees, overcome this problem. For the most part, rather than the large computer (PDP-10) used by SCORE, these systems have been implemented on inexpensive minicomputers. By using such machines, it becomes economically feasible to have the entire system dedicated to serving a single musician-user. Such a dedicated machine thus enables the prompt response (acoustic or otherwise) to the composer's commands. Thus, the tools are provided, throughout the entire compositional process, for the "intervention" and "control" associated with computer assisted composition. Examples of such systems are the NRC system in Ottawa (Pulfer, 1970 and Tanner, 1972), the GROOVE system (Mathews and Moore, 1970), POD (Truax, 1973 and Buxton, 1977), and that of the Experimental Music Studio of M.I.T. (Vercoe: Nov. 1976).

Each of the systems mentioned enables the composer to mould his materials in a way somewhat analogous to a sculptor. With the NRC and M.I.T. systems, the composer expresses himself in terms of common musical notation. The GROOVE system, on the other hand, utilizes a convenient form of graphical notation. To a greater or lesser extent, each system enables the user to deal with groups of sounds at a time, thereby going beyond the note-by-note approach of most sound synthesis programs. In many cases, especially in the POD programs, the system augments the simple transformations possible with the NRC system. This program has the ability to generate groups of sounds according to criteria similar to those seen in the ST program of Xenakis. Here, groups or structures can then be easily played back, augmented, and modified, thus defining the gradual evolution of a composition.

We see then, that the role of the computer aided system extends beyond that of an, albeit powerful, musical scratch pad, to what could be considered a composer's "assistant". All of this does not come without certain drawbacks, however. As was stated earlier, the main design criteria of such systems is to optimize, on a musical level, the communication between such an assistant and the composer. In so doing, certain sacrifices as regards sound quality or diversity must usually be made. Given a system appropriate to his needs, however, the composer is usually well compensated for such drawbacks, most of which are being overcome by current advances in technology.

In summary, the main attraction of computer aided composition systems is the potential for learning and communication made available to the user (Truax, 1976).. That is, in working with a program such as POD, the user is freed to concentrate on problems of composition: the design of well formed musical structures, rather than computer programming. Finally, it could be stated that it will most likely be through the experience gained in working with such systems that composers will come to better understand the compositional process, and thereby enable the development of better technological tools for their craft (Laske, 1975).

SOUND PRODUCTION WITH THE AID OF COMPUTERS

With sound synthesis, one must keep in mind the main task. This is the creation of an electrical signal, which is the analogue of the acoustic pressure function defining the sound to be produced. Simply stated, the goal is to produce a voltage comparable to that output by the stylus of a record player. Once produced, the electrical signal can be fed into an amplification system and converted into sound. In attempting to generate such a signal, however, one runs into several problems. To begin with, the pressure function associated with most sounds of musical interest is extremely complex (Risset and Mathews, 1969, Grey, 1975, and Benade, 1976). Thus, it is necessary to find a less complicated representation of the sound, before such a signal can be generated. Describing and synthesizing sound via its formant structure (as in much speech synthesis), or component sine waves (as in Fourier synthesis) are two examples of such representations or "acoustic models" of sonic phenomena. Regarding such models, the interested reader is referred to the excellent, if somewhat technical, survey in (Moorer, 1977).

Assuming the existence of such a model, it then remains to be asked, "how is the model seen by the user?". That is, given that a user wishes to define the characteristics of a sound to be synthesized using that model, what is the "description language" that he must use to do so? Thus, it is important to distinguish between the mathematical model being employed (the "internal representation") and how it is seen from the outside (the "external representation"). In some systems, such as those using Fourier synthesis, there is little difference between the two; however, in cases where the model is very complex, it is clear that some sort of meta-language more suitable to the musician is desirable. The purpose of such a language is to render the acoustic model "transparent" to the user. Through such a language, the composer is able to specify information in terms oriented to his application (music), letting the system translate this data into a form more appropriate to the acoustic model in use. While not directly related to computers, the "Solfege" developed by Pierre Schaeffer (Schaeffer, 1966) represents an important effort in the development of such a music oriented description language for sound. More recent research, such as that of Kaegi (Kaegi, 1973, 1974), is now oriented towards the implementation of such description languages in interactive computermusic systems.

Computers have many advantages over conventional modes of sound synthesis, such as the traditional electronic music studio. Generally stated, the computer is well suited to deal with the complexities involved. In this regard, both its memory and calculating power play an important role. With a suitable computer, one can efficiently simulate and test various acoustic models. The development of digital F.M. by John Chowning of Stanford University (Chowning, 1973), which has had such an impact on electro-acoustic music, is a case in point. Furthermore, it is precisely with the computer that we have the flexibility to develop description languages that make the resources of such acoustic models more accessible to the composer.

Some of the systems discussed below offer extreme flexibility, but often at the expense of increased cost and complexity. Others are easy to work with, but limited in sonic repertoire and quality. Trade-offs must be made, and these are largely user/application dependent. To date, there have been three main approaches to using computers in the sound generating process. These are: digital synthesis, hybrid systems, and mixed digital systems. Each of these approaches is presented below, with appropriate examples.

1) *Digital Synthesis*

This is the "classical" technique of sound synthesis first developed by Max Mathews of Bell Laboratories. It is the technique used in the MUSIC IV & V programs (Mathews, 1969), and their derivatives, including MUSIC 4B & 4BF by Howe and Winham (Howe, 1975), and MUSIC 360 (Vercoe, 1973, 1975). As well, it is used in the system of the C.E.M.A.Mu. (Xenakis), the IRMA system (Clough, 1971), and POD (Truax, 1973). While a complete discussion of digital synthesis is beyond the scope of this paper, the basic concepts are outlined below. For a more detailed treatment, the reader is referred to Mathews (1969).

Sound is perceived due to variations in the atmospheric pressure, as sensed by the ear. Each different sound is characterized by a unique pattern of pressure variation. Assuming that for a given sound we knew what this pattern was, we could then generate, on a digital computer, a sequence of numbers whose magnitude fluctuated in a way analogous to the pressure pattern under consideration. If the variation in the numbers' values is an adequate representation of the desired acoustic pressure variation (what is considered adequate will be discussed below), we can output the "samples" of the number sequence from the computer, through a device known as a "digital to analogue (D to A) converter" (Kritz, 1975; Freeman, 1977). This device produces a voltage whose amplitude is proportional to the magnitude of the number given as input. It is clear, therefore, that the voltage output by the D to A converter will then be analogous to the variations of the pressure pattern, just as is the sequence of numbers given as input. This fluctuating voltage can then be fed to an amplification system in the manner already discussed, thereby producing the desired sound.

Inherent in digital synthesis is an important trade-off. Information theory tells us that in order to adequately represent the bandwidth of audio (circa 16 kHz), the minimum number of numerical samples needed to represent one second of sound is 32,000 (Mathews, 1969). Even with the most powerful computers, this factor renders the calculation of all but the shortest and simplest compositions extremely expensive. This is where the nature of the acoustic model used is very important. The MUSIC V class of programs (which dominate the field) utilizes a model which digitally simulates the workings of apparatus found in an electronic music studio. While offering generality and complexity (one can simulate any "idealized" studio set-up with this system), one must pay in terms of long turnaround (i.e., typically a day between the time that data is submitted and the time when acoustic output is returned); furthermore, the complexity of the calculations involved dictates the use of a large general purpose computer, such as an I.B.M. series 360. This implies expense, sharing the system with other users, and generally working in a "batch" (card readers, etc.) environment, none of which is conducive to creative work. On the other hand, the general availability of such computers, the portability of the software, and the generality offered, makes such systems attractive to many users. Furthermore, the replacement of the "batch" approach by timesharing has improved the man-machine interface of some systems, such as that at Stanford University.

One can, however, take an alternate approach. Due to recent technological developments, small low cost (under $20,000) "mini-computers" are a viable alternative for music systems. While these machines have neither the calculative power nor the memory capacity of their larger brothers, they do make it economically feasible for an entire system to be dedicated to a single musician-user. This makes it possible for the first time to have computer music systems tailor made to meet the composer's needs. In digital synthesis, the price one pays for these advantages is a loss in generality and sound quality; a mini-computer can simply not do as much in as short a time as an I.B.M. model 360, for example. However, by choosing an acoustic model which is computationally efficient, these drawbacks can be largely overcome, with the added benefit that sounds can be auditioned immediately, in "real-time." Two examples of such systems are those of Truax (1973) and that at the Xerox Research Labs at Palo Alto, California (Saunders, 1975; Kaehler, 1975). Each of these systems is highly interactive, and capable of producing complex sounds with time-varying spectra. The results of such interaction are systems in which the potential for learning is very great.

While the systems of Saunders and Truax are in some ways limiting, such limitations are largely technical, and are rapidly being overcome by current technology. One should examine carefully their advantages from a musical viewpoint: both offer a wide palette of timbres through their use of the F.M. technique, and both are easy to learn. Furthermore, one must reconsider the terms in which we mean "loss of generality" for such systems. While the POD system is far less flexible than MUSIC V in its capacity for sound generation, its implementation enables the synthesis portion to be combined with an interactive compositional system, thus offering the composer a complete music package.

In summary, the trade-off with digital synthesis is generality and sound quality vs interaction and cost. The choice between the two is largely dependent on whether the system is composition or research oriented.

2) *Hybrid Systems*

In hybrid systems, sound production is carried out by analogue generators (oscillators, synthesizers, etc.), rather than by a computer. The computer in this case is used as a device to control the operation of the peripherals. Examples of such systems are PIPER (Gabura and Ciamaga, 1968), GROOVE (Mathews, 1970; Mathews and Moore, 1970), the Yale synthesizer (Friend, 1971), MUSYS (Grogono, 1973) and EMS (Wiggen, 1972).

In using peripheral sound generators, the computational demands are greatly reduced, as compared with digital synthesis. Whereas digital synthesis requires a minimum of 32,000 samples per second, hybrid systems only need approximately 100 for each device being controlled. As a result, smaller, and therefore less expensive computers can be used (for example, MUSYS utilizes a PDP 8, and EMS Stockholm a PDP 15). Furthermore, since sound quality is dependent on the quality of the devices being controlled, interactive systems can be implemented without the resultant loss of quality seen in digital synthesis.

By the very nature of hybrid systems, the acoustic model (and description language) typically used is a reflection of the apparatus being controlled. Such is the case with both the EMS and MUSYS systems; for example, in the EMS1 language, one specifies a sound in terms of the connections, settings and timings for the actual apparatus of the analogue studio. The computer then "plays back" the events as specified, to be accepted or modified by the user. The GROOVE system has expanded on this concept by introducing a graphics oriented control language which enables the user to modify the values of previously defined parameters during playback. The user's role during playback thereby becomes analogous to the conductor's in orchestral music.

Smaller, portable hybrid systems are now appearing, which are suitable for performance in concert situations. Examples of such systems are the HYBRID IV system of Ed Kobrin (Kobrin, 1975; Smith and Kobrin, 1977), and the systems commercially available from Donald Buchla Associates. Whether used in the studio or in concert, the main appeal of hybrid systems is the ability to perform, in real time, compositions made up of complex control and timing functions, and patching sequences, thereby bypassing the previous dependence on audio tape in auditioning the complete composition; furthermore, the utility of many systems extends beyond this use of the computer as an expanded sequencer, in that the user is able to invoke previously defined compositional procedures during actual performance. Two interesting approaches to this type of system are presented in (Rosenboom, 1975) and (Pinzarrone, 1977).

There are however, several drawbacks to hybrid systems. While the quality of the sound output by an analogue device may be quite high, the stability and accuracy can in no way match that of the digital device. Furthermore, whereas with a system such as MUSIC V, one can hypothetically simulate any number of analogue devices in any configuration, with hybrid systems one is restricted by the number and type of actual devices available.

3) *Mixed Digital Systems*

Mixed digital systems are those systems in which a computer is used as a control device for a digital sound generator, such as a digital oscillator. Existing examples of this approach to sound synthesis are the Dartmouth synthesizer (Alonso et al., 1975), the VOCOM system (Zinovieff, 1972), VOSIM (Tempelaars, 1976), the University of Illinois (Beauchamp et al., 1975), the I.R.C.A.M. system (Di Giugno, in press), and (Chamberlain, 1976). This type of system is perhaps the most promising in terms of the future of interactive computer music systems.

A conception of mixed digital systems can be gained by considering that any procedure that can be realized as a program (software), could theoretically also be realized by appropriate apparatus (hardware). That is, one could build a special processor to execute any programmable task. This processor is in turn controlled by the CPU of the main computer. While expensive, in realizing a complex procedure in hardware rather than software, one gains in one area in particular, that of time. Thus, if one developed an especially good — but time consuming — paradigm for sound generation using MUSIC V, for example, the paradigm could then be realized in hardware, enabling it to function in real time. Consequently, much research is now being carried out (by Kaegi, for example), to develop models which lend themselves to such implementation.

Mixed digital systems take the best from both worlds. One has the speed (and associated convenience) of an analogue hybrid system, combined with the accuracy and stability of digital synthesis. This is emphasized by the sampling rates of about 50 kHz possible with this technique (Zinovieff), or the ability of the Dartmouth synthesizer to output 16 voices of high quality F.M., in real time. The one shortcoming, as with the analogue hybrid systems, is the limitation set by the hardware configuration of the "synthesizer". This factor merely emphasizes the need to evolve an adequate acoustic model before realizing it in hardware; however, mixed digital systems have a distinct advantage over analogue hybrid installations. With the advent of inexpensive "micro-computers", it is becoming feasible to construct systems which can be quickly reconfigured in order to realize various alternative paradigms for sound generation.

SUMMARY

In summary, we have seen that there exist several different approaches to computer music. The various degrees to which a digital computer can participate in the compositional process have been discussed, together with the various types of such participation. Both composing programs and computer "aided" composition were examined. In addition, we have seen that there exist several approaches to generating sound for musical purposes using a computer. Techniques considered included digital, hybrid, and mixed digital synthesis. It has been shown that the various systems extant can be compared in terms of how acoustic phenomena are represented to the machine and to the user, and furthermore, in terms of the method of obtaining sounds from these representations.

From the above survey, it can be seen that the current trend in computer music is towards systems which are more "accessible" to the composer in the physical, economic, and music-theoretical sense. This is seen, in the mind of the author, as a tendency towards small interactive systems. It is felt that with mini-computer based mixed digital systems, coupled with well thought out modes of communication (graphic languages, etc.), the full potential of computers will be felt in both the composition and performance of music. Regardless of the accuracy of this conclusion, it is hoped that the preceding discussion will prove useful to those either actively or passively interested in the field.

ACKNOWLEDGMENTS

The author would like to thank Leslie Mezei, David Barnard, and G. M. Koenig for their many helpful suggestions and corrections to the manuscript. Any errors or misconceptions which remain are completely my responsibility. Finally, I would like to thank the UNIX operating system, which made the preparation of the manuscript almost enjoyable.

REFERENCES

Alonso, S., Appleton, J., Jones, C., "A Special Purpose Digital System for the Instruction, Composition, and Performance of Music," 17-22, in *Proceedings of the 1975 Conference on Computers in the Undergraduate Curricula,* Fort Worth, Dept. of Computer Science, Texas Christian University, 1975.

Alonso, S., Appleton, J., Jones, C., "A Special Purpose Digital System for Musical Instruction, Composition and Performance," *Computers and the Humanities* 10 (1976): 209-215.

Barbaud, P., "LVDVS MARGARITIS VITREIS," *Rapport de Recherche*, l'Institut de Recherche d'Informatique et d'Automatique, 78150 ROCQUENCOURT (France), 1976.

Beauchamp, J., Pohlmann, K., Chapman, L., "The TI 980A Computer Controlled Music Synthesizer," in *Proceedings of the Second Annual Music Computation Conference*, vol. 3: Hardware for Computer-Controlled Sound Synthesis, University of Illinois, 1975.

Benade, A. H., *Fundamentals of Musical Acoustics*, New York, Oxford University Press, 1976.

Baxton, W., *Manual for the POD Programs*, Utrecht, Instituut voor Sonologie, 1977.

Byrd, D., "A System for Printing Music by Computer," *Computers and the Humanities* 8 (1974): 161-72.

Chamberlain, H. A., "Experimental Fourier Series Universal Tone Generator," *J.A.E.S.* 24.4 (1976): 271-276.

Chowning, J., "The Simulation of Moving Sound Sources," *J.A.E.S.* 19.1 (1971): 2-6.

Chowning, J., "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation," *J.A.E.S.* 21.7 (1973): 526-34.

Clough, J., "Computer Music and Group Theory," *American Society of University Composers Proceedings*, 4 (1969): 10-19.

Clough, J., "An Interactive System for Computer Sound Generation," *American Society of University Composers Proceedings*, 6 (1971): 22-26.

Di Giugno, G., "A Real-Time Computer Controlled Digital Oscillator Bank," *Proceedings of the First International Conference on Computer Music*, M.I.T. Press, in press.

Freeman, D. M., "Slewing Distortion in Digital-to-Analogue Conversion," *J.A.E.S.* 25.4 (1977): 178-183.

Friend, D., "A Time-Shared Hybrid Synthesizer," *J.A.E.S.* 19.11 (1971): 928-935.

Gabura, J., Ciamaga, G., "Computer Control of Sound Apparatus for Electronic Music," *J.A.E.S.* 16.1 (1968): 49-51.

Grey, J. M., "An Exploration of Musical Timbre," *STAN-M-2*, Stanford, Dept. of Music, 1975.

Grogono, P., "MUSYS: Software for an Electronic Music Studio," *Software Practice and Experience* 3 (1973): 369-83.

Hiller, L., "Computer Music," *Scientific American* 201.6 (1959): 109-120.

Hiller, L., Isaacson, L., "Musical Composition with a High-Speed Digital Computer," *J.A.E.S.* 6.3 (1958): 154-160.

Hiller, L., Isaacson, L., *Experimental Music*, New York, McGraw-Hill, 1959.

Hiller, L., Ruiz, P., "Synthesizing Musical Sounds by Solving the Wave Equation for Vibrating Objects," *J.A.E.S.* part 1: 19.6 (1971): 462-470; part 2: 19.7 (1971): 542-551.

Howe, H. S., "Composing by Computer," *Computers and the Humanities*, 9 (1975): 281-290.

Howe, H. S., *Electronic Music Synthesis: Concepts, Facilities, Techniques*, New York, W. W. Norton, 1975.

Kaegi, W., "A Minimum Description of the Linguistic Sign Repertoire," *INTERFACE*, part 1: 2 (1973): 141-53, part 2: 3 (1974): 137-57.

Kaehler, T., "Some Music at XEROX Parc," *Proceedings of the Second Annual Music Computation Conference*, vol. 1: Software Synthesis Techniques, University of Illinois, 1975.

Kobrin, E., *HYBRID IV Users Manual*, La Jolla, unpublished manuscript, Center for Music Experiment, 1975.

Koenig, G. M., "A Program for Musical Composition," *Electronic Music Reports* 2 (1970): 32-44.

Koenig, G. M., "Project 2," *Electronic Music Reports* 3 (1970): 4-161.

Kritz, J. S., "A 16-Bit A-D-A Conversion System for High-Fidelity Audio Research," *IEEE ASSP* 23.1 (1975): 146-149.

Laske, O., "Towards a Theory of Musical Cognition," *INTERFACE* 4 (1975): 147-208.

Mathews, M., *The Technology of Computer Music*, Cambridge, M.I.T. Press, 1969.

Mathews, M., "The Electronic Music Studio of the 1970's," *La Revue Musicale* (1970): 129-41.

Mathews, M., Moore, F., "GROOVE — A Program to Compose, Store, and Edit Functions of Time," *Comm. ACM* 13 (Dec. 1970).

Moorer, J. A., "Music and Computer Composition," *Comm. ACM* 15.2 (1972): 104-113.

Moorer, J. A., "The Synthesis of Complex Audio Spectra by Means of Discrete Summation Formulas," *J.A.E.S.* 24.9 (1976): 717-727.

Moorer, J. A., "Signal Processing Aspects of Computer Music — A Survey," *Computer Music Journal* 1.1 (1977): 4-37; also to be published in: *Proceedings of the IEEE* July, 1977.

Pinzarrone, J., "Interactive Woman-Machine Improvisation or Live Computer-Music Performed by Dance," *Creative Computing,* 3.2 (1977): p. 66.

Pulfer, J. K., "Computer Aid for Musical Composers," *Bulletin of Radio and Electrical Engineering Division* 20.2 (1970): 44-48.

Rader, G. M., "The Formal Composition of Music," *Technical Report No. 77-1,* Dept. of Computer Science, University of Ife, Ile-Ife, Nigeria, 1977.

Risset, J. C., Mathews, M. V., "Analysis of Musical-Instrument Tones," *Physics Today,* 22.2 (Feb. 1969): 23-30.

Rosenboom, D., "A Model for Detection and Analysis of Information Processing Modalities in the Nervous System Through an Adaptive, Interactive, Computerized, Electronic Music Instrument," *Proceedings of the Second Annual Music Computation Conference* vol. 4: Information Processing Systems, University of Illinois, 1975.

Saunders, S., "Real-Time Digital FM Audio Synthesis," *Proceedings of the Second Annual Music Computation Conference,* vol. 1: Software Synthesis Techniques, University of Illinois, 1975.

Saunders, S., "Improved F.M. Audio Synthesis Method for Real Time Digital Music Generation," *Computer Music Journal* 1.1 (1977): 53-55.

Schaeffer, P., *Traité des Objets Musicaux,* Paris, Editions du Seuil, 1966.

Schaeffer, P., "Music and Computers," *La Revue Musicale* (1970): 57-92.

Smith, J. B., Kobrin, E., *Kobrin: Computer in Performance,* Berlin, DAAD, 1977.

Smith, L., "SCORE -- A Musician's Approach to Computer Music," *J.A.E.S.* 20.1 (1972): 7-14.

Tanner, P., "MUSICOMP, an Experimental Aid for the Composition and Production of Music." *ERB-869,* Ottawa, N.R.C. Radio and Electrical Engineering Division, 1972.

Tempelaars, S., "The VOSIM Oscillator," *Proceedings of the First International Conference on Computer Music,* M.I.T. Press, in press.

Truax, B., "The Computer Composition — Sound Synthesis Programs POD4, POD5, & POD6," *Sonological Reports* 2 (1973).

Truax, B., "A Communicational Approach to Computer Sound Programs," *Journal of Music Theory,* 20.2 (Fall, 1976): 67 pp.

Truax, B., "The Inverse Relationship Between Generality and Strength in Computer Music Programs," *Proceedings of the First International Conference on Computer Music,* M.I.T. Press, in press.

Vercoe, B., "The MUSIC 360 Language for Sound Synthesis," *American Society of University Composers Proceedings,* 6 (1971): 16-21.

Vercoe, B., *Reference Manual for the MUSIC 360 Language for Digital Sound Synthesis,* Cambridge, unpublished manuscript, Studio for Experimental Music, M.I.T., 1975.

Vercoe, B., *Man-Computer Interaction in Creative Applications,* Cambridge, unpublished manuscript, Studio for Experimental Music, M.I.T., November 1975.

Wiggen, K., "The Electronic Music Studio at Stockholm; its development and construction," *INTERFACE* 1 (1972): 127-165.

Xenakis, I., *Formalized Music,* Bloomington, Indiana University Press, 1971.

Zinovieff, P., *VOCOM: A Synthetical Engine,* London, Technical Report of E.M.S. London Ltd., 1972.

Received June 1977.

William A. S. Buxton
Computer Systems Research Group
Sandford Fleming Building
University of Toronto
Toronto, Ontario,
Canada