Joel Chadabe
P.O. Box 8748
Albany, New York 12208 USA

Interactive Composing: An Overview

Introduction

Interactive composing is the name I have given to a method for using performable, real-time computer music systems in composing and performing music. The concept of interactive composing has grown in my work since 1967, at which time I proposed the design for a sequencer-programmable analog synthesis system, subsequently built by the R. A. Moog Co. and installed in the Electronic Music Studio at State University of New York at Albany in December 1969. That system's special capability in automating controls made possible the techniques that led to several compositions, in particular *Ideas of Movement at Bolton Landing* (1971) (Chadabe 1972), which involved the control of timbre and rhythm interactively with automated controls generated by sequencers, and *Echoes* (1972) (Chadabe 1975; 1977), in which the sounds performed by an instrumentalist are delayed for a few seconds, then transformed by a sound-processing system, then distributed to various loudspeakers. The transformed sounds appear as distant echoes to which the performer reacts in deciding how to play the next notes.

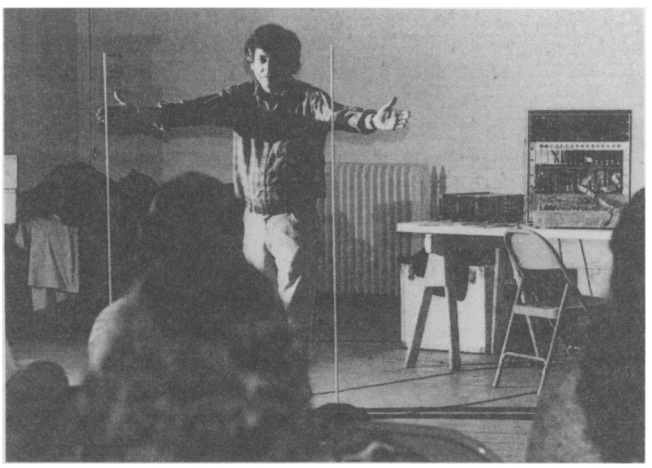
Echoes is an example of what I referred to at that time as a *design-then-do* procedure for composing, that is, the creation of a system followed by its functioning. That procedure was also the basis of other compositions, such as *Flowers* (1975) (Chadabe 1977) and *Settings for Spirituals* (1976), where the do stage, because of the impossibility of transporting the equipment to a concert hall, was performed in the studio and resulted in tapes. It was the basis also for *Play*, a computer program for control of an analog synthesizer, written by myself and Roger Meyers in the spring and summer of 1977. *Play1*, the first version of the program, was described at the time as "functioning in two stages: (1) a design stage, where the composer designs a specific compositional process, using any of the modules available in the program, and (2) an operation stage, where the composer's process plays back and the composer interacts with the playback according to the design" (Chadabe and Meyers 1977).

In 1977, I began to compose for and perform with a small computer and digital synthesizer system, which I carried and still do carry to concerts. To my knowledge, it was the first instance of a completely digital system used in musical performance. Two compositions have grown out of those performances, *Solo* (1978) and *Rhythms* (1980), which are the first computer music statements of interactive composing (Chadabe 1982).

The computer's function in *Solo* is to compose automatically the notes of a melody, its accompaniment chords, and other aspects of the music, and to interpret the positions of a performer's hands in relation to two proximity-sensitive antennas (Fig. 1). As I perform, I move my right hand in relation to the right antenna, thereby controlling the speed of the melody by increasing or decreasing the duration of each note. As I move my left hand in relation to the left antenna, I control timbre by passing my hand through zones in which different computer-generated instruments, reminiscent of vibraphones, clarinets, and flutes, are playing. I cue two clarinets and two flutes to play slowly, for example, but I cannot foresee what chord they will play. Reacting to what I hear, I decide what to do next.

In *Rhythms*, the computer automatically generates melodies and rhythmic patterns, articulated in sounds reminiscent of Indonesian, Caribbean, and African percussion instruments. I perform by pressing keys at the terminal keyboard, thereby transposing chords, changing pitch relationships within chords, triggering melodic variations, altering rhythmic patterns, overlapping voices, and introducing random notes. But although I trigger each set of changes to begin, I cannot foresee the details of each change. I must react to what I hear in deciding what to do next. It is a distinctive characteristic of interactive composing that a performer, in deciding each successive performance action, reacts to information automatically generated by the system.

Fig. 1. A performance of Solo *by Joel Chadabe. The right hand controls the speed of the melody played by a digital synthesizer, while the left hand controls the choice of computer instruments. (Photograph by Carlo Carnevali.)*



Another Early Example

Although some of the elements of interactive composing are evident in works by other composers (automated decision making is an essential aspect of the music of John Cage, Lejaren Hiller, and Iannis Xenakis, for example, and performance with automated electronic systems plays an important role in much of the music of Giuseppe Englert, Morton Subotnick, and many others), Salvatore Martirano's *SalMar Construction*, built between 1969 and 1972 at the University of Illinois, exemplifies the primary characteristics of the approach. It consists of analog sound generators and modifiers controlled by digital circuits, a console with over 200 touch-sensitive switches, and 24 loudspeakers arranged throughout a performance space. The system automatically generates sounds with different timbres, pitches, and loudnesses, and routes the sounds along four paths through the arrangement of loudspeakers. Martirano performs in reaction to what he hears, manipulating aspects of the music, such as pitch, rhythm, tempo, pattern, octave, spatial distribution, and cycling.

A Definition of Interactive Composing

Interactive composing is a two-stage process that consists of (1) creating an interactive composing system and (2) simultaneously composing and performing by interacting with that system as it functions. Creating the system involves bringing together a programmable computer, synthesizer, and at least one performance device, and programming the computer with algorithms that function automatically and in real time to

Interpret a performer's actions as partial controls for the music
Generate controls for those aspects of the music not controlled by the performer
Direct the synthesizer in generating sounds

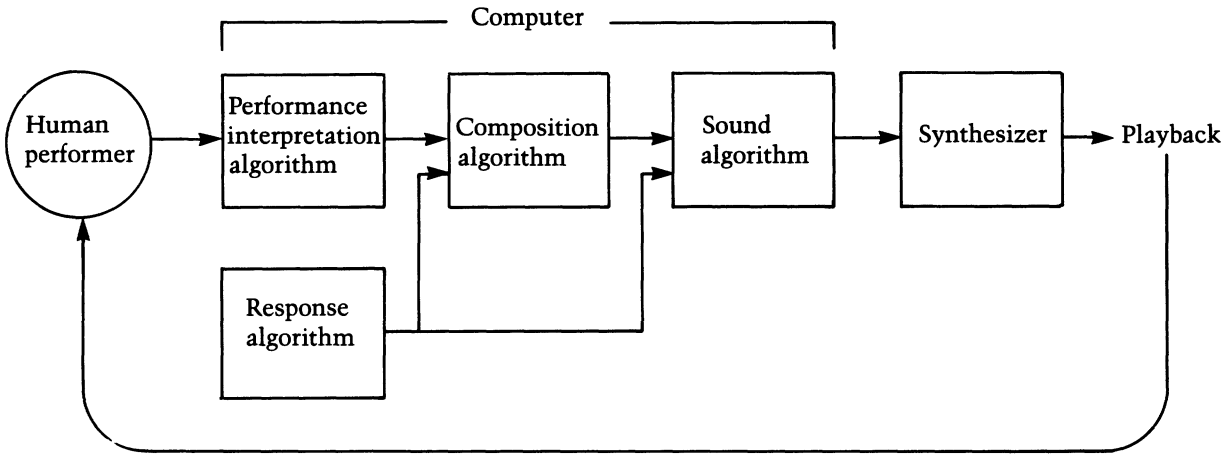
An interactive composing system operates as an intelligent instrument—intelligent in the sense that it responds to a performer in a complex, not entirely predictable way, adding information to what a performer specifies and providing cues to the performer for further actions. The performer, in other words, shares control of the music with information that is automatically generated by the computer, and that information contains unpredictable elements to which the performer reacts while performing. The computer responds to the performer and the performer reacts to the computer, and the music takes its form through that mutually influential, interactive relationship. The primary goal of interactive composing is to place a performer in an unusually challenging performing environment.

Description of the System

The organization of an interactive composing system is indicated by the block diagram in Fig. 2. The *performance interpretation algorithm* translates a performer's motions into specific control information. In *Solo*, the computer "sees" the space around each antenna as divided into concentric zones and interprets a performer's hand's distance from an antenna as being in a certain zone. In *Rhythms*, pressing a key interrupts one process and initiates another.

The *composition algorithm* consists of the variables that define the composing processes of the system. In *Solo*, the most important variables define the way the melody and accompaniment chords take form. In *Rhythms*, the important vari-

Fig. 2. Organization of an interactive composing system, from human performer to sound output.



Feedback loop contains information generated by computer in addition to information specified by performer.

ables define the way the rhythmic and melodic patterns occur.

The *sound algorithm* translates the variables that define the sounds into the format required by the synthesizer. The synthesizer used in *Solo* and in *Rhythms* is the Synclavier I, which is used to generate eight voices of frequency-modulated sounds. Each voice requires specifications for modulator and carrier frequencies, a modulation index, amplitude envelope, and loudness.

The variables of the composition algorithm are the most important determinants of how the system will function. In *Solo*, the melody algorithm, which is the basic component of the composition algorithm, functions as indicated in the flowchart in Fig. 3.

Stated as questions, the melody variables are the following:

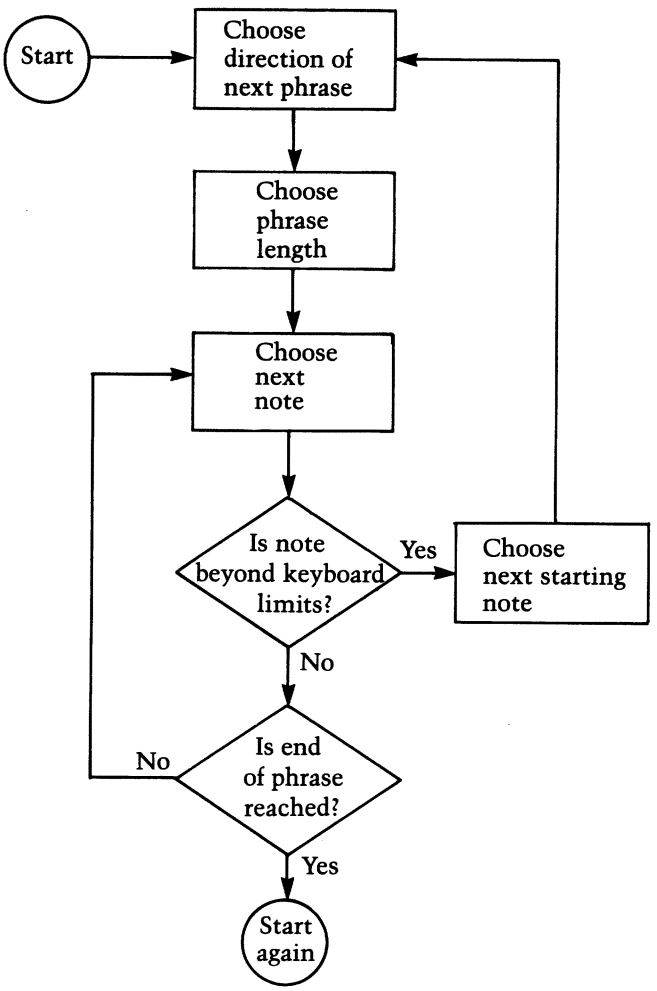
Will the next phrase go up or down?
What is the number of notes in that phrase?
What will be the next note, as determined by a jump measured in semitones from the present note?
If the melody exceeds the range of note possibilities, what is the new starting note for a phrase?

The notes of the accompaniment chords are determined by the specification of intervals higher or lower than each melody note.

In *Rhythms*, the important variables derive from two lists, the first a list of randomly placed ones and zeros, the second a list of randomly chosen notes. At the start of each variation, the following two questions are asked for each of eight voices: At what location in the first list shall a cycle begin? How long, from two to nine positions in the list, shall a cycle be?

Rhythms is based on synchronous beats, and the first list is used to determine whether or not a voice will sound on a particular beat. If voice 1, for example, were directed to start at position 23 in the first list and play a cycle of three, and if positions 23, 24, and 25 in the list contained 0, 1, and 0, then voice 1 would not play on the first beat, play on the second, not play on the third, and then begin the cycle again. Six voices repeat their rhythmic patterns on single notes that are chosen at the beginning of each variation. The remaining two voices read a cycle of notes from the note list, also by choosing a starting location and cycle length. Performer-selected routines call into play other sets of variables, such as continually changing notes within patterns, intervals of transposition, and tempo.

Fig. 3. Flowchart of the melody algorithm for Solo.



Intelligence in the System

In interactive composing, the system's response to the performer must be interesting and informative; that is, it must contain new, unexpected information. Yet the response must also be recognizably related to the performer's actions, because if the performer did not perceive clearly the effects of his or her actions, the act of performing would cease to have any meaning. The response must be as in a conversation, where the reply to a statement is related to the statement but not a repetition of it.

The seemingly contradictory qualities of newness and recognizability are brought together through shared control. The performer controls certain of the composition variables and the response algorithm controls others.

In *Solo*, the response algorithm determines the details of the melody and shares control of tempo and timbre with the performer. The primary tempo variable is the period allowed to pass between points in time that mark the beginnings of sounds. In specifying a particular period, the performer, with the response algorithm adding a bit of randomness, sets the envelope segments of each sound to expand or contract to fit within the specified period. The performer exerts a powerful control, determining the ranges within which the durations of the envelope segments will occur, and the response algorithm fills in the details with some unpredictability.

The timbre of the sounds in *Solo* results from the performer's specifications of which instruments are playing, from the nature of the sounds themselves (which is a function of the sound algorithm), and, most important for the performer, from the particular voicing and spacing of the accompaniment chords as controlled by the response algorithm. The performer may specify two flutes and two clarinets, for example, but without knowing the voicing and pitches of the resulting chord, the performer cannot predict precisely the timbre of the total sound. The performer is sufficiently surprised to have to react to the sound in deciding the nature of the next event.

In *Rhythms*, the performer triggers complex patterns of pitches and rhythms to change, but the details of each change are determined by the response algorithm. The successive intervals of a rising melodic pattern, for example, are not predictable, which causes the pattern to be different every time it occurs and propels the performer toward the next decision.

The response algorithm in *Solo* and in *Rhythms* consists of a random-number generator. Like other random-number generators, it is mechanistic and, by itself, does not suggest intelligence. The important quality of random-number generators, however, is what I have come to call complexity. In general, by *complexity*, I mean that the rules which determine a certain succession of events, as, for example, the numbers in the number series 15, 7, 11, 5, 10, 13, 6, 3, 9, 4, 2, 1, 8, . . . , are underlying and

elusive and not evident in the succession of events they generate. Throwing dice, for example, where the numbers that appear are the result of a multitude of underlying and elusive causes, such as the way the dice are held as they are thrown, the strength with which they are thrown, the balance of each die, the relative weight of each die as it strikes another, and so on, may be seen in this light as a method for generating a complex series of numbers. In contrast with *simplicity*, where the rules that determine a succession of events are evident in the events they generate, as in the number series 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, . . . , the events in a complex series seem to occur as if by chance.

The reason for thinking of a random-number generator as complex rather than random is that other complex generators, based on other operating principles, might be devised. A future response algorithm, for example, might embody the artificial intelligence model of the knowledge-based, self-aware, self-organizing system, responding to and evaluating a performer's actions in framing an interesting response.

For the present, however, one might generalize that complexity results in whimsy. It is certainly true that, from the point of view of the performer of an interactive composing system, aspects of the music that are determined by a complex procedure contain whimsical, unpredictable elements. When a complex procedure is used to provide a performer with new yet intelligible information (e.g., in *Solo* and in *Rhythms*), the random-number generator causes the computer to exhibit attributes of intelligent behavior.

The Significance of Interactive Composing

Interactive composing redefines composing and performing. Instead of composing a particular musical structure, as does a composer of traditional music, the creator of an interactive composing system composes a mode of functioning for computer system and performer that, in operation, generates a new particular structure in every performance.

The tasks of interactive composing are substantially different from those of traditional composing. The problems central to creating a mode of functioning are those of defining a composition algorithm and deciding which of its variables are best controlled by a performer and with what device. This is quite different from writing music for a traditional instrument, where a performer's actions are dictated by the mechanical construction of the instrument and always control the same sound variables. (In playing a piano, for example, depressing a key always controls pitch and loudness.) The creator of an interactive composing system specifies freely a performance action (by specifying an appropriate device) and specifies freely which composition variable or group of variables it will be used to control. The choice of performance device influences the behavior of the system (and the music it produces), because each type of performance action evokes a different musical sensibility. Each type of action, consequently, leads a performer to think and feel in a certain way about the music and about the specific musical variable that is being controlled. For example, proximity-sensitive antennas reinforce the feeling of performing *Solo* as if conducting an instrumental ensemble, because the motions of moving one's hands in the air resemble what a conductor does, and the musical effects of those motions, instruments entering and tempos changing, are typical effects of conducting.

Also different is the nature of a composer's control over the music. When a composer of traditional music creates a particular structure, the whole is known as the parts are made, and the parts can be made to fit. Music produced by an interactive composing system, however, unfolds only as the result of the specific functioning of the system in a particular performance, and the composition as a whole is known only retrospectively, after the parts have been made.

Further, a traditional composition exists apart from any particular performance of it. But in interactive composing, because the music and the performance are inseparable, the quality of the music itself, rather than simply the quality of its execution, is in large part dependent on the ability and talent of the performer. The performer is simultaneously a participant and an overviewer who functions interactively within the system, devising

strategies for using the unexpected to advantage, and also functions outside the system by supplying a more global perspective and guiding the progress of the music as a whole.

A Closing Remark

The ultimate significance of interactive composing is that it represents a new way for composers and performers to participate in a musical activity. I offer my nontechnical perception that good things often happen—in work, in romance, and in other aspects of life—as the result of a successful interaction during opportunities presented as if by chance; to that I would add only that it seems to me reasonable that such a perception should also find expression in music.

References

Chadabe, J. 1972. *Ideas of Movement at Bolton Landing.* Greenville, Maine: Opus One Records 17.

Chadabe, J. 1975. *Echoes.* (Jan Williams, percussionist.) New York: Folkways Records FTS 33904.

Chadabe, J. 1977. *Echoes, Flowers.* (Paul Zukofsky, violinist.) New York: CP2 Records 2.

Chadabe, J. 1982. *Rhythms.* (Jan Williams, percussionist.) New York: Lovely Music VR 1301.

Chadabe, J., and R. Meyers. 1977. "An Introduction to the Play Program." *Computer Music Journal* 2(1): 12–18. Revised and updated version forthcoming in *Foundations of Computer Music,* ed. C. Roads and J. Strawn. Cambridge, Massachusetts: MIT Press.