CHAPTER 1

INTRODUCTION

For the purpose of this book, "automated composition" embraces all compositional processes which are systematic enough to be implemented as computer programs. As our experience with computers grows, more and more processes have been found to fall within this category. As recently as thirty years ago the qualifier "systematic" would have implied only numeric series such as occur in twelve-tone music; in the present book, "systematic" embraces diverse types of randomness, conditional selection, hierarchic generation, recursive decision-making, and many other procedures. The more sophisticated of these strategies transcend specific musical styles by simulating general human approaches to solving problems; still other methods completely bypass traditional notions of craft by directly addressing the listener's perceptions.

1.1 HISTORICAL OVERVIEW

In considering ways of using computers to compose music, we should always understand that "composition" means many things to many people. There is no correct method of producing a musical score; rather, there is a multitude of methods, and an important component of the total act is the process of selecting which among these methods are appropriate for solving whatever problems a composer might confront in a specific situation. Indeed, the wide diversity of backgrounds, influences, philosophies, motivations, and tastes which have distinguished composers throughout the history of music has been reflected more recently by an equally wide diversity of approaches and methods among practitioners of automated composition.

1.1.1 American Research: 1957 to 1964

Though several earlier attempts had been made (see Hiller, 1970), automated composition first gained public notice with the efforts of Lejaren Hiller (b. 1924) and Leonard Isaacson (b. 1925) at the University of Illinois, Urbana/Champagne. In 1957, Hiller and Isaacson undertook a series of compositional experiments using the Illiac computer which they arranged into an Illiac Suite for string quartet. Hiller and Isaacson used two basic approaches: 1) random selection constrained by lists of rules 2) Markov processes, also random, in which the relative likelihood of each option was conditioned by one or more immediately preceding choices.

In 1962, Hiller teamed up with Robert Baker to produce a composing utility called MUSICOMP (Hiller and Baker, 1964; the name is an acronym for MUsic Simulator Interpreter for COMpositional Procedures). The MUSICOMP approach greatly facilitates the process of developing new composing programs by providing libraries of compositional subroutines and other programming modules which individual composers may exploit or ignore as they please. These subroutines are linked together by a main program designed by the composer to meet his own idiosyncratic needs. The approach requires that a user also be a capable programmer, but in return it provides great flexibility. New subroutines may be developed to augment those already present in the library, and these new subroutines may themselves be added to the pool if they are judged to be of general application. Hiller and Baker first utilized MUSICOMP to produce their 1962

Computer Cantata, which incorporated approaches from the *Illiac Suite* in addition to serial methods then fashionable among the *avant-garde*. MUSICOMP has been employed by Hiller's colleague Herbert Brun (b. 1918) to create a number of works including *Sonoriferous Loops* (1963) and *Non-Sequitur VI* (1966).

Also associated with the University of Illinois during this period were faculty member John Myhill (b. 1923) along with James Tenney, who worked as Hiller's assistant. In 1961, Tenney was invited by Max Mathews to serve as the first in what came to be a series of 'composers-in-residence' at Bell Telephone Laboratories, an experience chronicled in Tenney's article "Computer music ~~experiments~~ experiences 1961-1964" (1969). Of the compositions realized by Tenney at Bell Labs over these years, all but the first is computer-composed. Tenney's work introduced gradual evolutions along with hierarchic procedures inspired by the Gestalt psychology of Max Wertheimer.

Among Hiller and his colleagues, a strong motivation for automating the compositional process has been the insight into creative activity which interaction with a computer can provide. For example, Hiller and Isaacson's book *Experimental Music* (1957) describes the *Illiac Suite* as a study of "those aspects of the process of composition..." which can be modeled "...by applying certain mathematical operations deriving from probability theory and certain general principles of analysis incorporated in a [then] new theory of communication called information theory". Hiller and Baker's article "Computer Cantata: A study in compositional method" (1964) expresses a similar attitude: "Since our primary purpose was to demonstrate the flexibility and generality of MUSICOMP, the Computer Cantata presents a rather wide variety of compositional procedures, some of which proved of greater esthetic value than others, and many of which could be improved by more sophisticated logic." In "From musical ideas to computers and back" (1970), Herbert Brun emphasizes the value of such efforts in the following terms:

> ... whereas the human mind, conscious of its conceived purpose, approaches even an artificial system with a selective attitude and so becomes aware of only the preconceived implications of the system, the computers would show the total of the available content. Revealing far more than only the tendencies the human mind, this nonselective picture of the mind-created system should prove to be of significant importance.

1.1.2 European "Algorithmic Music": 1960-1970

The first European composer to become associated with automated composition was Pierre Barbaud, who presented computer-composed pieces in Paris as early as 1960. Many of Barbaud's procedures are documented in a monograph entitled <u>Initiation a la composition musicale automatique</u> (1966); they include permutational methods applied to traditional harmonies, serial (twelve-tone) methods, and methods of random selection. The article "Algorithmic Music" (circa 1970) sets forth an aesthetic philosophy which holds that musical composition "consists in creating what scholastics called an <u>artifactum</u>, that is to say something that nature does not produce." Barbaud confronts objections that automation removes the "humanity" from composition by turning the complaint against itself:

> Music is generally called "human" when it considers temporary or inherent tendencies of the mind, of part or all of a composer's personality. Such music is based on feeling and since it turns its back, in a sense, on pure knowledge, it might rather be called

> "inhuman", for it celebrates what we have in common with all the animals rather than with what is individual to man: his reason. Algorithmic music is thus "inhuman" only in one sense of the word, it is "human" in as much as it is the product of rational beings.

Perhaps the best-known composer to employ computers has been Iannis Xenakis. Already well-known in the late 1950's for his introduction of statistical distributions into such manually composed works as Pithoprakta (1956) and Achorripsis (1957), Xenakis imposed such distributions upon random procedures with a "stochastic music program" developed in 1962. This program, which we shall subsequently refer to as the "ST program", requires only the most general information to generate a complete musical score: length of composition, mean duration of sections, instrumentation, and so on. In his book, Formalized Music (1971), Xenakis explains his attraction to computers by citing motivations similar to Hiller's, Brun's, and Barbaud's. The following imagery elaborates on Xenakis's own sentiments:

> With the aid of electronic computers the composer becomes a sort of pilot: he presses the buttons, introduces coordinates, and supervises the controls of

a cosmic vessel sailing in the space of sound, across sonic constellations and galaxies that he could formerly glimpse only as a distant dream. Now he can explore them at his ease, seated in an armchair.

A less well-known but equally significant figure has been Gottfried Michael Koenig (b. 1926). Koenig worked with Karlheinz Stockhausen at the Cologne Electronic Music Studio before taking a position at the Institute of Sonology in Utrecht, the Netherlands. Koenig's work in automated composition began with a composing program entitled PROJECT1 (1964). PROJECT1 is responsible for several of Koenig's compositions, including <u>Project 1, Version 1</u> for 14 instruments (1965). Though PROJECT1 is serially oriented, Koenig indicates in his 1970 description of the program that it was not so much the motivic aspects of serial practice which attracted him but rather the saturation of values used to control the various musical attributes. For example, Koenig's use of 12-tone rows insures that the twelve chromatic degrees will adhere strictly to a uniform distribution, but motivic relationships occur only accidentally because PROJECT1 shuffles the twelve degrees randomly with each new statement. A more elaborate program called PROJECT2 (1969) includes selection procedures which allow some values to occur more frequently than others, but which still maintain very tight distributions, and the notion of "statistical frames" discussed in Chapter 6 is based upon this approach. Though Koenig developed PROJECT2 for pedagogic use, he has used it himself to create an ambitious Uebung fur Klavier ("Study for Piano", 1970). Closely associated with Koenig at Utrecht has been Otto Laske.

1.1.3 "Interactive" Utilities in the U.S. and Canada: 1972-1978

During the mid-1970's, automated composition was well-nigh eclipsed by the more popularly appealing (and much more lucrative) field of digital sound synthesis. Typical composing programs developed during these years are utilities oriented strictly toward synthetic realizations: Mathews's GROOVE (1970; most extensively used around 1974 by Emmanuel Ghent and Laurie Spiegel), Leland Smith's SCORE (1972; also developed for music printing), Truax's POD (first version, 1973), Brun's SAWDUST (1976), and Buxton's SCED (1978). With the exception of SCORE, these utilities have been motivated largely by a desire to facilitate rapid interaction between composer and sound; their originality lies less in the compositional procedures employed (most of which are derived from earlier efforts) than in their economic means for issuing directives and evaluating the results.

In particular, a real-time environment such as Max Mathews's GROOVE system allows a composer/performer to deal at a reflex level with sounds and timings. The player is enabled to bypass a certain amount of conscious reasoning; in payment, he becomes limited to a restricted class of compositional procedures. Real-time procedures must be simple enough so that the delay (due to execution) between touching a keyboard or adjusting a knob and hearing the musical result is imperceptible.

Two students of Koenig's, Barry Truax and William Buxton, have organized computer music facilities in Canada: Truax in Vancouver, Buxton in Toronto. In addition to developing real-time performance systems, both Truax and Buxton have developed non-real-time environments which allow a user to get by with little or no programming. These environments enable users to monitor results very closely as a composition takes shape. They minimize delays in non-real-time interaction by providing a menu of precompiled automata and by incorporating facilities for rapidly evaluating products, specifically, real-time digital synthesis augmented visually by graphic display.

1.1.4 New Approaches to Form: 1976-1983

More recent practitioners have brought about a resurgence of interest in automated decision-making along with a new concern for matters of large-scale compositional form. Included among the 'new generation' of practitioners are Thomas DeLio, Clarence Barlow, Curtis Roads, Steven Holtzman, Kevin Jones, and the author. DeLio's Serenade for piano (1976) employs a large variety of determinate and/or statistical procedures in order to create a multisectional form with an equally large variety of material. Barlow's composition Cogluotobusisletmesi ("Bus Journey to Parametron", 1978) also for piano ("two-or-more hands") uses computers not only to select musical details, but also to generate lists of resources and to tackle ambitious formal problems.

Curtis Roads's monograph, A Systems Approach to Composition (1976) describes a composing program called PROCESS/ING which dramatically generalizes the Markov processes employed earlier by Hiller and Xenakis. In Road's program, selections of attributes describing a complex musical event (e.g., densities and mean frequencies for a cloud of sonic "grains") are conditioned not only by the past history of the attribute under consideration, but also by the histories of all other attributes.

In a subsequent monograph, Composing Grammars (1978), Roads forsakes this 'left-to-right' approach for a 'top-down' approach inspired by linguistic models proposed by Noam Chomsky. These models enable a composer to describe elaborate musical forms very economically by providing a general 'archetype' of the form and by further listing a set of "productions" for deriving details from generalities. Significant extensions directly inspired by Road's ideas include the "Generative definitional language for music" described by Holtzman (1980) and the notion of "stochastic web grammars" proposed by Jones (1981). The author's composition Crystals for string orchestra (1980) results from similar efforts undertaken in response to an independent source of inspiration -- the Gestalt principles advocated by James Tenney.

The effectiveness of composing programs has been radically improved through the incorporation in recent years of recursive programming techniques. Such techniques make it possible to implement compositional automata which are capable of propagating "copies" of themselves, so that automata spawn sub-automata, these in turn spawn sub-sub-automata, and so on to arbitrary levels of complexity. Recursive capabilities are necessary features of the "productions" described above, which must be able to act upon their own results.

An especially potent approach which also relies on recursive programming techniques has drawn from the discipline of computer science known as "artificial intelligence" in order to implement decision-making programs which actively discriminate between multiple options. Such programs divide into two classes, comparative searches and constrained searches. Comparative searches systematically evaluate every possible configuration of decisions in order to discover the best possible solution to a problem. Constrained searches improve dramatically on the generate-and-test strategy of Hiller and Isaacson's Illiac Suite 1) by organizing options into schedules so that the most desirable options are considered first and 2) by incorporating the ability to backtrack and revise one or more earlier decisions, should every option in a schedule prove unacceptable.

Though a fully recursive search is described by Stanley Gill as early as 1968, the approach has mostly been ignored until very recently. Exceptions include a search with non-rigorous backtracking implemented by Larry Polansky to compose his Four-Voice Canons (1975) along with more rigorous searches employed by Kemal Ebcioglu to model compositional processes in sixteenth-century counterpoint (1980) and to simulate J.S. Bach's procedures for harmonizing chorales. Recursive searches have been heavily exploited by the author to compose works such as Protocol (1981) and Gradient (1982), both for piano, along with Undulant for seven instruments (1983).

1.2 MUSICAL NOMENCLATURE

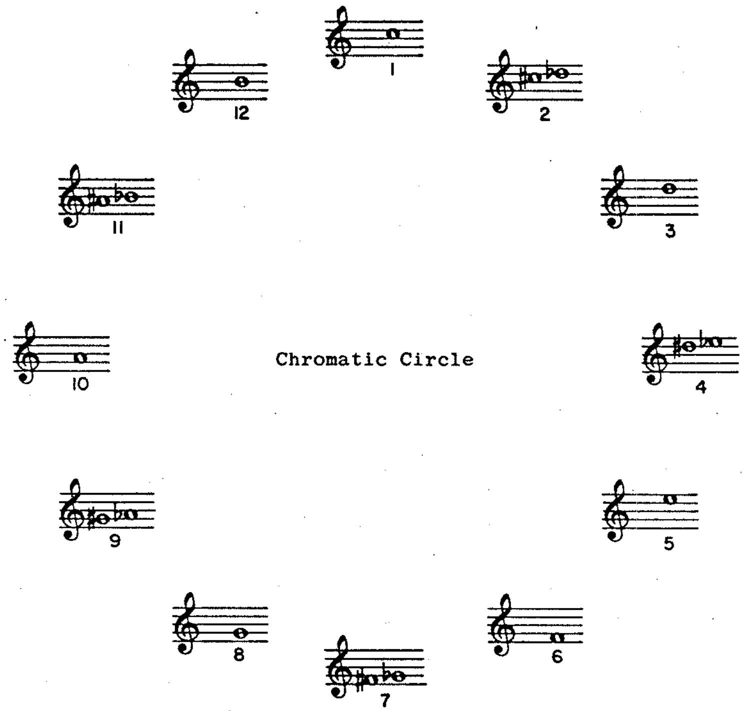
Terminology in technical literature such as the present book can easily lead to highly abstruse language, in turn prompting reactions such as the following by Stravinsky:

> The computer prose of musical analysis is probably a necessary tool for the new music, though it makes me feel a graybeard, left leagues behind. Nor can I see the utility of such locutions as 'pitch priorities', 'simultaneities' (yclept 'chords' in my day), and 'dyads'... . My vocabulary consists entirely of hangover expressions from defunct musical traditions, and virtually all of my words require quotation marks or the qualifier 'so-called', though these are mere warning flags for word-inflation.

The author ~~highly~~ strongly concurs with Stravinsky's complaint, and this book adopts the practice of using traditional musical terminology whenever it seems to apply. It will therefore be useful to lay out here at the beginning how such traditional vocabulary translates into today's analytic jargon.

When two pitches in different registers have the same letter name and accidental (that is, if their frequency ratio closely approaches a power of two) then these pitches exhibit the property known among musical analysts as "octave equivalence". Such relationships are referred to by analysts and also by the author (when necessary) as identities. We shall use the word degree to denote qualities of pitch which function independently of register. This word typically indicates a position either in a chord or in a scale; for example, "the fifth degree of the major scale" indicates the degree which lies a perfect fifth above the tonic. Though it is most common to enumerate the degrees of a scale in ascending steps, the property of "octave equivalence" suggests that a scale should properly be regarded as a repeating sequence which "wraps around" upon itself. It also often proves fruitful to consider alternatives to stepwise representation.

Figure 1-1: Chromatic Resources of Twelve-tone Equal Temperament.


Chromatic Circle
1
2
3
4
5
6
7
8
9
10
11
12

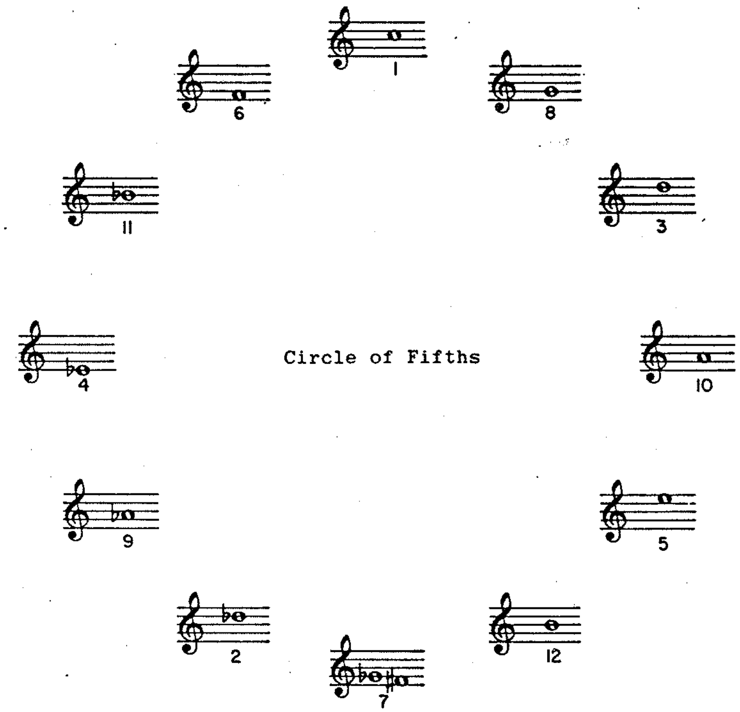

Circle of Fifths
1
8
3
10
5
12
7
2
9
4
11
6

Figure 1-1

In particular, the "chromatic circle" and the "circle of fifths", both depicted in Figure 1-1, illustrate the two most common ways of representing the standard temperament of twelve equally spaced pitches to the octave. Much of the music described in this book employs standard temperament; in describing this music, the author strongly prefers the traditional term "chromatic degree" over the American analytic term "pitch class". We shall also adopt the European convention (c.f. Ligeti, 1958) of enumerating positions on the chromatic circle from 1 (C) to 12 (B) (note 1).

Figure 1-2: Classification of intervals.

Though this book employs the time-honored classification of intervals illustrated in Figure 1-2, we hasten to qualify that the distinction between "consonances" and "dissonances" and between "perfect" and "imperfect" consonances no longer carry the normative connotations associated with these terms when they were first introduced during medieval times. One should also recognize that in certain harmonic contexts a perfect fourth, or a major or minor sixth against the bass projects an unstable feeling which has led these intervals to be treated similarly to dissonances in such contexts.

Among psychologists it has been the practice to characterize

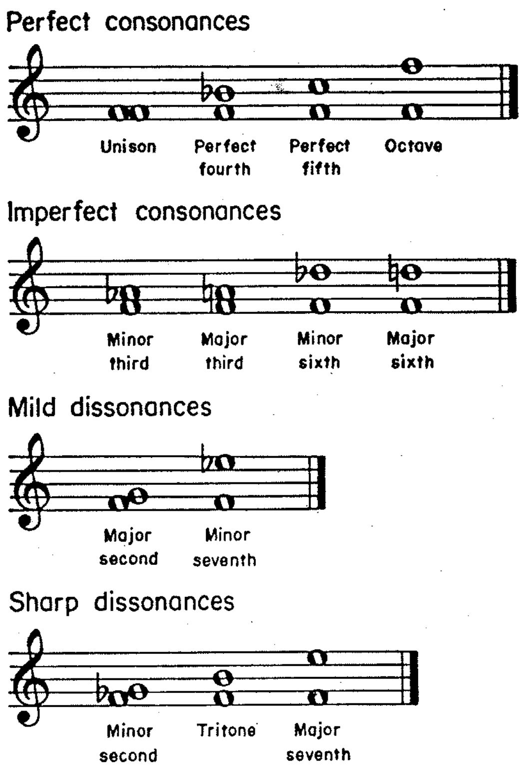

Perfect consonances
Unison
Perfect fourth
Perfect fifth
Octave
Imperfect consonances
Minor third
Major third
Minor sixth
Major sixth
Mild dissonances
Major second
Minor seventh
Sharp dissonances
Minor second
Tritone
Major seventh

Figure 1-2

perceptions as "dissonant" or "consonant" depending upon whether or not they conflict with a subject's prior experience. Lately it has become fashionable among musical analysts to adopt this practice, much to the confusion of all concerned: if one should encounter a sustained, root-position major triad in the midst of listening to Schoenberg's Erwartung, so the notion goes, one would perceive such a triad as a dissonance. The author admits that such an occurance would be startling, but prefers the older term, "contrast".

When discussing rhythm, this book shall denote the beginning time of a note (or blocked chord) as the attack and the ending time as the release. We shall also distinguish between durations and periods as follows: a duration denotes the segment of time joining the attack and release of note (that is, the segment over which a note physically sounds), while a period denotes a segment of time joining attacks of two consecutive notes (note 2). Finally, the term rhythmic unit will be occasionally necessary to signify segments of time occupied by either a note or a rest, but not both.

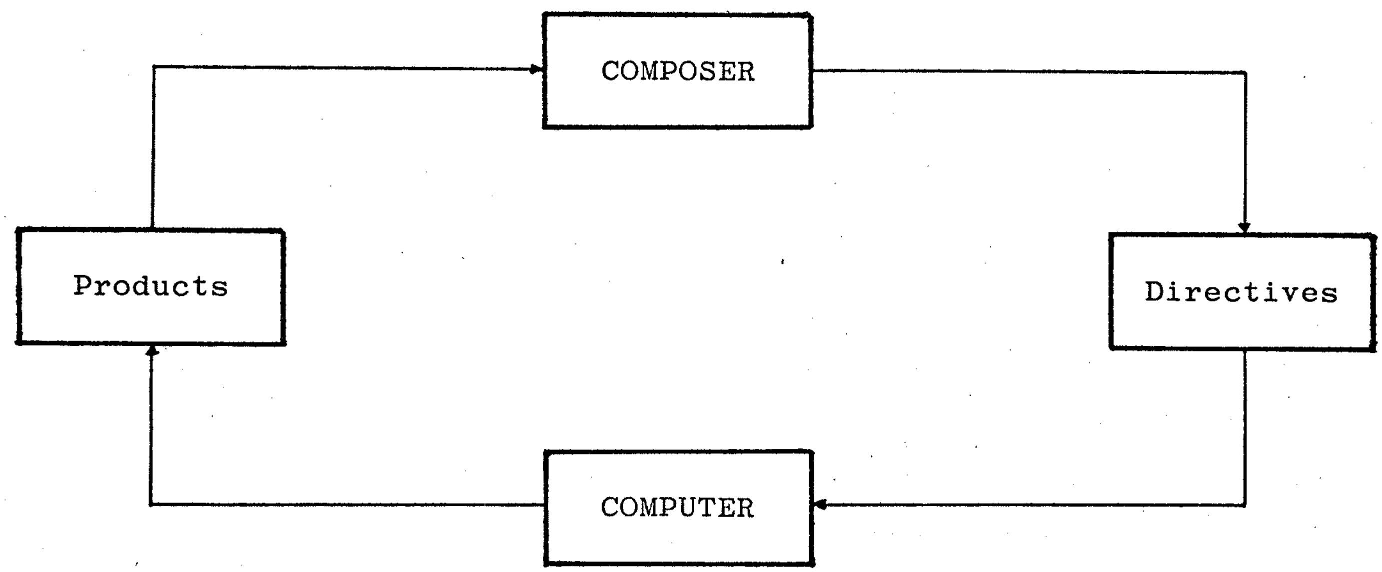


Figure 1-3

1.3 PRODUCTION OF COMPUTER-COMPOSED SCORES

Figure 1-3 illustrates the terminology adopted by this book to describe the process of automated composition at its broadest level of generality. We represent this process as an interactive loop with two phases: the composer supplies musical inspirations the form of directives understandable to a computer (that is, input); after some period of computation, the computer responds to these directives by supplying products (output) to the composer. The composer may exploit these products in several ways. As one option, the composer may accept these products as the final musical result. Alternatively, the composer may employ products as intermediate results by directing the computer to elaborate upon these products in further cycles through the loop. A third alternative is to discard the products and to initiate new, more refined directives.

We should recognize that the act of composition often divides into many stages of production. Individual stages might include such tasks as generating material, describing forms, determining chordal progressions, working out counterpoint, orchestrating. Clearly, the nature of each stage will vary with the composer and with the specific composition. A stage of production may itself divide into substages, sub-substages, and so on. For example, producing an entire composition might entail composing some number of sections; producing a section might entail composing some number of segments; a segment might consist of some number of phrases, phrases of motives, motives of fragments, and so on until ultimately every note in the composition has been produced.

Any person or machine attempting to conduct such tasks as are described above is continually required to make <u>decisions</u>. The simplest decisions are acts of <u>selecting</u> attributes for various sorts of entities; for examples: selecting a key for a movement, a local tonic for a phrase, a pitch for a note. More elaborate decisions may often be reduced to acts of selection by employing suitably general definitions of "entities" and "options". Consider an act of <u>organization</u>, for example, deriving a twelve-tone row by taking the twelve chromatic degrees and arranging these degrees in some order. This act may be subsumed under "selection" if one takes the row itself as an "entity" and each of the (479 million) possible arrangements as an option. Options for high-level decisions might themselves constitute entire <u>lists</u> of options for lower-level decisions. For example, one might orchestrate an ABACABA movement by selecting woodwinds and strings (eliminating brass, percussion, electric guitar, and so on), then in turn by employing only strings for the B sections, only woodwinds for the C section, and both groups for the A sections. Within the C section one could orchestrate some passages in double reeds, some in flutes and clarinets, and so on. Extending the notion of lists, it is even possible to select distributions of options for lower-level decisions, simply by appending a relative weight to each entry in a list.

Much of this book will be concerned with decision-making strategies. A common feature of most strategies is that they recognize constraints on the field of options and attempt to provide a result which is "acceptable" in the sense that it adheres to these constraints. More sophisticated strategies not only judge options as "acceptable" or "unacceptable", they also analyze the specific circumstances of a decision in order to favor the more desirable of the acceptable choices. We call a decision conditional, or contextually sensitive, if its options (or their relative likelihoods) vary in this manner with the circumstances. Feedback occurs when each in a series of decisions is actively conditioned by the results of its predecessors.

1.4 OVERVIEW OF SUBSEQUENT CHAPTERS

This book consists of fourteen chapters, four of which may be considered 'auxiliary' and ten of which are 'primary'. The 'auxiliary' chapters include the present introduction along with chapter 2, which describes the programming conventions employed in this book, chapter 10 and chapter 13, which develop certain advanced data structures necessary for more sophisticated applications.

Each of the 'primary' chapters investigates a technique of automated composition using roughly the following format:

1. Description of what the technique is, what its motivations are, and how it works,

2. Explanation of how ~~the specific instances~~ useful aspects of the technique may be implemented as closed subroutines,

3. Analysis of computer-generated scores by composers who have employed the technique, and

4. One or more demonstrations, usually drawing upon compositional insights developed in items 1 and 3. These demonstrations combine the procedures implemented in item 2 with those presented in earlier chapters in order to construct full-fledged composing programs.

Chapters 3 through 7 deal primarily with decision-making procedures: literal selection (3), direct random selection (4), statistical frames (5), Markov chaining (6), and cumulative feedback (7). Chapter 8 introduces techniques of prescribing determinate compositional 'profiles' using linear and exponential graphs. Chapter 9 explains how a ubiquitous computer programming technique, "sorting", can be exploited both directly, to organize musical material, and more subtly, as a way of 'scheduling' certain decision-making processes. Chapter 11 shows how hierarchic grammars can generate complex but highly integrated musical structures. Finally, chapters 12 and 14 apply strategies drawn from "artificial intelligence" which are capable of "searching" through extremely large collections of potential solutions in order to determine which of these solutions is most appropriate.

The sequence of chapters is arranged to present techniques in increasing order of sophistication; each chapter also draws heavily upon insights developed during earlier chapters. For these reasons, the recursive searches described in the culminating chapter, 14, cannot be properly appreciated without the knowledge of linked data structures provided in chapter 13 and the knowledge of recursive procedures provided by chapter 10, without understanding of the sorting mechanism described in chapter 9 and without fluency in the methods of conditional decision-making presented in chapters 6 and 7; these last-named methods in turn can be well appreciated only with the insights about statistical distributions and of randomness developed in chapter 4. In short, no chapter is self-contained. The most beneficial way of reading this book for the first time is to start at the beginning and work through to the end.

The original demonstrations appearing periodically through the book progress gradually from very simple models to elaborate, multi-stage processes. For ease of understanding, all have been encoded in a consistent language (FORTRAN '77) using similar names for similarly functioning variables. Because the demonstrations have been freshly composed for this book, it has been possible for the author to trace the genesis of each work very closely from his initial directives through each stage of production to the final musical product (note 3). In order to show how the techniques can be adapted to varied styles, the demonstrations adopt an eclectic approach: some demonstrations are diatonic, others chromatic, some emphasize consonant relationships, others emphasize dissonances. The procedures are fully capable of polyphony even though the products are scored in these instances for a monophonic instrument, the clarinet; indeed several of the later demonstrations employ arpeggiation and perceptual 'channeling' to produce "implied polyphony" of very convincing effect.

1.5 NOTES

1. Much American analytic literature numbers the twelve "pitch classes" from 0 (C) to 11 (B).

2. Equivalents for the term "period" include "time-point interval" and "entry delay".

3. The composing programs reproduced in this book have been substantially edited down from their actual running versions, which provided quasi-graphic displays allowing the author to monitor decisions in progress. Such displays are indispensible tools for developing and debugging programs; however, including the requisite code here would have swelled the listings to at least twice their present size. It should be acknowledged that

these programs were by no means conceived immediately in their present form; rather, they evolved as the end result of many trials occuring over a year's time.

1.6 RECOMMENDED READING

Brun, Herbert, 1970. "From musical ideas to computers and back", chapter 2 of The Computer and Music, Harry Lincoln, editor (Ithaca, New York: Cornell University Press).

Hiller, Lejaren, and Leonard Isaacson 1959. Experimental Music (New York: McGraw-Hill). Reprinted 1979 (Westport, Connecticut: Greenwood Press).

Hiller, Lejaren, 1970. "Music composed with computers -- a historical survey", chapter 4 of The Computer and Music, Harry Lincoln, editor (Ithaca, New York: Cornell University Press).