elaborate, but understandable complaints. As Mya Tannenbaum states later in the book: "The meetings with Stockhausen always took place where he was working: Florence, Paris, Milan, Rome. The recurring theme of these conversations was his persistent protest against the multitude of difficulties that had to be overcome in defence of his music."

More interesting from a musical standpoint is his discussion of the Stockhausen "formula." In each of Stockhausen's work is some kind of formal generating idea—unique to each piece—such as a series of pitches, or a relationship among different series of parameters. Stockhausen likens this notion to a genetic code for each piece.

The book concludes with a chronology of the life and works of Stockhausen from 1928 to 1987, and an index. Add another volume to the legend of Karlheinz Stockhausen.

Reviewed by Tona Green
Westport, Connecticut USA

John Strawn, editor: The Proceedings of the AES Fifth International Conference: Music and Digital Technology
Audio Engineering Society, 60 East 42nd St., New York, New York 10165 USA, 1987. 248 pages, softcover, $35.00

In his Preface to this volume, John Strawn states the purpose of the AES Fifth International Conference: "to summarize the state of the art of digital music-making, and to point directions to the future." To judge by these papers, that goal was accomplished admirably. In this volume, representatives from all areas of the digital music world—both the commercial and the academic/research sectors—cover a rich variety of topics. The interest/boredom ratio is exceedingly high, and the reader is left hoping that this conference will become an annual event.

I will comment upon the papers in the order in which they were presented.

Dominic Milano: Digital Music-Making: The Past, Present, and Future. The paper's title is more comprehensive than its content; the author's view of the state of the art seems to be limited to commercial, MIDI-related products. Milano's jaded asides on the state of quality assurance in the music industry are especially enjoyable:

> Sequencers, particularly software sequencers, are the worst culprits when it comes to getting what you paid for about three updates and six months later.

Writing in May 1987, Milano seems to have predicted the history (to date: February 1988; story still incomplete) of Mark of the Unicorn's Performer Version 2.x, right down to the number of updates and months!

David Rossum: Digital Music Instrument Design: The Art of Compromise. Rossum, of E-mu Systems, discusses cost/performance tradeoffs in the design of commercial instruments. The commercial developer's area of research, he says, is "that of determining what they can get away with." He urges engineers to adopt an experimental attitude, through which they can arrive at a musically useful and commercially successful design—even though the canons of digital audio purists may be violated. Design choices made in the development of E-mu's Drumulator and SP-12 drum machines are used as illustrations.

Rossum's pragmatic attitude is refreshing, especially if one is accustomed to reading learned academic papers about systems which are "not yet fully implemented," or which are accessible only to three professors and their cronies. One wishes one saw more such candid articles by real-world instrument and software designers.

J. William Mauchly and Albert J. Charpentier: Practical Considerations in the Design of Music Systems Using VLSI. Mauchly and Charpentier, of ENSONIQ Corporation, discuss features they consider desirable in a single-chip synthesis engine in general, and go on to describe the Ensoniq Digital Oscillator Chip (DOC).

They start with a comparison of the relative merits of variable-sampling-rate and fixed-rate (with interpolation) wavetable synthesis techniques. The latter are favored on the grounds that fixed-rate oscillators are less hardware-intensive and that their outputs can be more readily routed to further processing devices in the digital domain.

The final section of the paper describes the architecture of the DOC. In light of the authors' conclusions above, it is interesting that the DOC employs a *non*interpolating fixed-rate oscillator; techniques used to minimize the distortion resulting from the absence of interpolation circuitry are covered. The authors conclude by predicting that fixed-rate techniques will be more widely used in the future.

Steven M. Misek and Dana C. Massie: Multitasking Operating System Design for Electronic Music. This article sketches ARTOS (Audio Real Time Operating System), the system that runs on the various processors which, networked together, make up the AudioFrame workstation. In many ways, ARTOS resembles other

real-time system kernels such as VRTX. It is a nonpreemptive multitasking system in which tasks are selected for execution on a prioritized basis. Facilities exist for interrupt-to-task, intertask, and interprocessor message passing, for dynamic task creation, memory management, and other traditional system functions. ARTOS also includes extensions appropriate to an audio workstation, such as routines for sample memory management, voice allocation, MIDI processing, and vibrato. This paper is rather sketchy; four-and-a-half pages simply do not suffice to cover a system of such complexity. It would be good to see more extensive coverage of ARTOS and the AudioFrame, perhaps in the pages of this journal.

F. Richard Moore: What is a Computer Music Workstation? Moore first defines computer workstations in general; he then discusses the demands made upon a system in real-time digital audio processing applications. Current general purpose computers, Moore contends, cannot accomplish the necessary processing quickly enough to produce high-bandwidth audio in real time. Most readers would find this obviously correct, but Moore's discussion of alternative approaches to digital synthesis makes many nonobvious points. One alternative—lowered fidelity via lower sampling rates—would at first seem undesirable, but Moore points out that it has its place in a workstation for the efficient production of musical "proof copies." The alternative of non-real-time processing also has its advantages; as the author observes, editing (perhaps the most widespread current application of workstations) is inherently non-real-time. The third alternative is special-purpose hardware; Moore contrasts software and hardware implementations of a simple oscillator to illustrate the advantages of the latter. Laudably, the author includes actual source and object code to make his point.

Moore classifies possible audio workstations according to their audio quality, degree of user programmability, and support for various modes of operation: real-time, non-real-time, interactive, and noninteractive. In his view a complete workstation must support all of these modes of operation, must be fully programmable by the user (including a program development environment), and must produce high-quality audio output. The paper's final section is an overview of the CARL workstation, which consists of two interconnected subsystems supporting both real-time operation and general purpose computing. Moore concludes on an optimistic note: such full-featured workstations will, he believes, be "within the means of seriously interested individuals in the near future." Let us hope so.

Bruce Pennycook, Jeffrey Kulick, and Dave Dove: Music Workstations: Real-Time Subsystem Design. The authors, as their title implies, are less interested in operations outside of real-time than is Moore. They focus upon the human-interface requirements of a workstation that "must be capable of the type of immediacy that analog gear provides." Timing constraints for controllers and for disk-based recorder/player systems are examined in some detail. The parallel-processing architecture of the Image and Audio Systems MINITAUR-SPS is then discussed in the context of such requirements. This system, comprising multiple signal processing boards employing the TMS32020 DSP chip, is said to yield up to 128 time-multiplexed audio channels at a 50 KHz sample rate. As described by the authors, it appears to be powerful; but in their own words it is "somewhat less than user-friendly." This seems an understatement, since they also mention that the user must undergo the "discomfort of writing 32020 assembler code."

Paul Lansky: Linear Prediction: The Hard, but Interesting Way to Do Things. Where it not for Lansky's paper, one might almost conclude from these *Proceedings* that software-based non-real-time synthesis had disappeared from the face of the earth. The bulk of Lansky's presentation is a tutorial on signal analysis/resynthesis by means of linear prediction. Lansky feels that the technique is a powerful means of independently manipulating the timbre, pitch, and duration of sampled signals, but he does not hesitate to point out its drawbacks: it is computationally intensive and often less than successful when used with non-speech signals. This article is an excellent introduction to the subject.

Christopher Yavelow: Composition or Improvisation? Only the Computer Knows! Yavelow deals with a relatively new trend in computer music-making, often termed "interactive composition." After a discussion of the nature of composition and improvisation as traditionally defined, Yavelow gives his view of interactive composition: it is a "hybrid creative process" that combines aspects of both composition and improvisation. It descends from algorithmic composition, in that a program makes decisions about the variation of musical input supplied by the user or a programmer; but these decisions are made in real time and are influenced by real time interaction with the user. Interactive composition is currently embodied in a number of popular programs, and

most of this paper describes three of them: M, Jam Factory, and Music Mouse. As these are widely known, I will not summarize the discussion here.

Yavelow has written an engaging description of an interesting trend, but he makes a number of questionable assertions. Do programs like these really "offer the benefits of knowledge equivalent to years of advanced musical training" to the untutored user? It is true that almost any musican can sit down for a few minutes with M and generate something musically palatable. But such a program seems to me to be a sophisticated tool, akin in its complexity to a musical instrument. Most musical instruments require prolonged and intensive study and practice before virtuoso results are achieved; I see no reason why "intelligent instruments" should be any different.

And is the "eventual replacement of looping by probabalistic variations" really to be welcomed? It is difficult to believe that the creators of station breaks, jingles, and so forth will handle this resource any more imaginatively than they now handle looping MIDI sequencers. As long as 15 years ago, Roger Powell was producing interesting loop variations quite similar to the output of Jam Factory using simple analog sequencers and sample/hold units. But—since technology is more widespread than imagination—the loopers kept right on iterating.

Finally, I do not share the author's enthusiasm for phenomena such as "clip music" and "style templates" which—he claims—will become widespread in the near future. Does the world really need a Barry Manilow "style template" in the hands of hack producers everywhere? Indeed, Yavelow fails to consider some of the more mind-boggling implications of such practices. Thanks to "style templates," composers (notorious seekers after immortality) will be able to go on composing after they are dead—without the intervention of psychics. One supposes that deceased critics, by means of their own "templates," will be able to continue to abuse these composers. If one extends the concept to nonmusical areas, one can foresee "templates" that will allow politicians to lie both in their graves and to their constituents. (The reader may remember the 1950s science fiction thriller, *They Saved Hitler's Brain. . . .*)

Craig Anderton: The MIDI Protocol. In his usual succinct fashion, Anderton covers the basics of the binary numbering system and of serial data transmission; this is followed by an exposition of the format and significance of each message type employed in the basic MIDI specification. He does not go into such extensions of the specification as MIDI Time Code (MTC) or the Sample Dump Standard. This article covers material that has received much attention elsewhere—for example, in Anderton's own excellent book, *MIDI for Musicians.* Nevertheless it will serve as an adequate introduction to the subject for the nontechnical reader who wishes to understand MIDI at the lowest byte-by-byte level.

Perry Leopold: MIDI by Modem: The Future Is Now. This paper is little more than a commercial. Nevertheless, as a frequent PAN user, I can attest that its description of the network is accurate and that PAN is a valuable source of practical technical information, and a generally entertaining place to pass some spare time.

Kimball P. Stickney: Computer Tools for Engraving-Quality Music Notation. Stickney first develops criteria for high-quality music printing systems. Of primary importance are the quality of iconic (fixed-shape, or "fontable") and algorithmic (defined by sets of instructions rather than lines, dots, or curves) objects; accurate connection of objects; and flexibility with respect to object placement and page layout. Next, he considers aspects of the design of music notation sofware—including the relative merits of systems that emphasize graphic tools versus those based on intelligent musical rules, and the importance of a good user interface. The final portion of the paper describes the design objectives of Notation Research's High Score program: "a system with an integrated implementation of quality fonts, well-designed algorithmic objects, sophisticated connectivity capability, and flexible layout and placement features."

Evan Brooks: Software in the Studio. Digidesign's Brooks begins by summarizing present-day applications of PC software in the recording industry. What follows is more interesting: a sort of consumer guide to the pitfalls of software acquisition. The reader is warned about vaporware, planned obsolescence, bugs, poor support and service, inadequate documentation, disk crashes, copy protection, and other familiar nightmares. This discussion is particularly welcome in that it comes from a leading software developer who practices what he preaches; Digidesign has not, in my experience, committed any of the listed software sins, other than copy protection.

Brooks concludes with a discussion of the problem of integrating studio functions—in particular, problems of synchronizing devices that support incompatible time-code/cueing schemes. Brooks objects to the "bandaid box" approach to such problems, in which hardware is used to convert SMPTE code to MIDI

sync, because such boxes are expensive, nonstandard, and require the entry of "tempo maps" to control the rate at which MIDI clocks are emitted. Brooks offers MIDI Time Code (MTC) as a solution. Regrettably, he does not discuss MTC in much detail, so it is somewhat difficult to evaluate his claims in its favor. It does seem clear, however, that MTC is a less comprehensive solution to the problem of integrating studio functions than the digital audio workstations discussed in other conference papers. MTC is, however, less expensive than such workstations, and it does not make obsolete currently existing devices.

Adrian Freed: Recording, Mixing and Signal Processing on a Personal Computer. Freed describes MacMix, a program that has been licensed to Integrated Media Systems to run with the Dyaxis digital audio processor. MacMix integrates the functions of a hard disk digital audio recorder with editing, mixing, and some signal processing capabilities.

Freed's paper, though brief, is one of the more exciting papers in this collection, since it announces the existence of a versatile, software-upgradable, and nearly affordable digital audio work environment hosted on a popular personal computer.

Pat Downes: Motion Sensing in Music and Dance Performance. Downes discusses techniques for sensing, transducing, and processing the gestures of freely moving performers with a view toward generating musically expressive MIDI output. The performer—or portions of the anatomy of a performer—is modeled as an object moving freely relative to a reference frame. This object's velocity and acceleration can be measured by appropriate transducers, such as gyroscopes and accelerometers. Transducer principles are discussed, as are problems (timing resolution, aliasing artifacts) involved in sampling their output. Once the transducer has been digitized, the result is mapped onto discrete or continuous MIDI messages of various types, and transmitted to musical devices. The Airdrums, a commercial motion sensing device/MIDI controller, is briefly described. This device senses the rotational acceleration of two hand-held tubes and emits MIDI control signals. Doubtless its operation could have been the subject of another equally interesting paper.

Robert A. Moog: Position and Force Sensors and Their Applications to Keyboards and Related Control Devices. Moog describes three keyboard controllers that allow performers to apply continuously variable control to one or more musical parameters per key. The Kurzweil MIDIBOARD supports polyphonic afterpressure. It also has the interesting capability of retriggering a note as a function of the player's afterpressure variations. The Key Concepts Notebender allows the player to move the key in both the up-and-down and back-and-forth dimensions. The Big Briar multiply-touch-sensitive keyboard lets the player control the key on left-right, front-back, and up-down axes, providing independent control of the three parameters.

Some discussion of what sonic parameters might be appropriately manipulated with such controllers, and of the ergonomic and performance difficulties that might be experienced by keyboardists playing the devices, would have made this paper more interesting.

David Wessel, Pierre Lavoie, Lee Boynton, and Yann Orlarey: MIDI-Lisp: A Lisp-based Programming Environment for MIDI on the Macintosh. An interesting development in personal computer music has been the introduction of programming tools for the MIDI-oriented programmer/musician. Those available in the Macintosh world, for example, include subroutine libraries such as Altech Systems' MidiPascal and MidiBasic, the inclusion of MIDI support in language implementations such as Coral Software's Object Logo, and interactive programming environments such as Frog Peak Music's HMSL. Wessel and others describe ACT Informatique's MIDI-Lisp, a programming environment running under Le_Lisp on the Macintosh. Moreover, they describe it in terms comprehensible to non-Lispers, which is no mean feat.

Lisp was chosen as the basis for a music environment because its facilities for symbol manipulation make it appropriate for working with musical structures. Lisp systems are highly interactive, providing fast feedback to composers, and—say the authors—the language is easy to learn. One drawback to using Lisp for real-time applications is the well-known garbage collection (reclamation of dynamic data structures) problem, which can halt all running processes for a time. The authors feel that both Le_Lisp and MIDI-Lisp have been implemented in such a way that this problem can be avoided.

MIDI-Lisp adds several specialized features to Lisp. A MIDI driver handles data streams through both Macintosh serial ports (2 * 16 channels), with configurable buffer sizes. The driver time-stamps incoming data, sequences outgoing data, and can send or receive MIDI sync. The driver also provides functions for outputting MIDI data in several convenient formats, and inverse functions for MIDI input. At a higher level, a module for sorting events in time order is provided; such events

include both MIDI events and "Lisp events," that is, time-stamped dispatching of Lisp processes. MIDI-Lisp also provides support for graphic control panels that can be used as input to musical processes.

The final section of the paper describes a real-time application written in PREFORM, another Le_Lisp extension that employs the MIDI-Lisp drivers.

MIDI-Lisp has been the subject of rumor and speculation for well over a year. One hopes that it will soon see general release; if it lives up to its description, that will be an exciting Lisp event.

Brian L. Schmidt: Natural-Language Interfaces and Their Application to Music Systems. Schmidt's article reviews developments in natural-language systems research (mainly in areas such as robotics and database query) and describes CIMS (Conversational Interface for Music Systems), an English-like command "front end" to a score processor/MIDI sequencer. CIMS can control score manipulation, MIDI recording and playback, and studio automation functions.

At present, CIMS is limited to performance of what Schmidt terms "well-defined" tasks—that is, those that involve precise modifications to the state of the studio configuration. The command "play the bass" describes such a task, since it requires only the activation of a playback channel which has previously been associated with the term "bass." The interface does not handle more complex tasks requiring knowledge of the musical context or subjective judgments, such as "make the bass sound fuller." Within its restricted domain, CIMS appears to be a powerful tool for the studio user whose hands happen to be free for command typing. As Schmidt points out, a voice-recognition system for command input would be a considerable enhancement.

Bill Buxton: Masters and Slaves Versus Democracy: MIDI and Local Area Networks. Buxton describes several well-known problems with MIDI: its low speed, low bandwidth, and its unidirectional nature, which often requires repatching of devices or the use of switch and merge "boxes." He rejects one possible solution—abandoning the standard in favor of a high-speed bus—and offers another: a proposal to interconnect MIDI devices and computers on a local area network (LAN) in which each MIDI device is connected to a "MIDI server." Such a server is a node on the network and consists of a single-chip transceiver (LAN controller) for communication with the network, a single-chip MIDI controller for communication with a MIDI device, and—between these two—a microprocessor whose software can perform MIDI switching, merging, filtering, and other functions while translating between the LAN and MIDI protocols.

This architecture has the advantage that diverse components—the above-mentioned "boxes," and even conventional MIDI interfaces—are replaced by a single uniform server component for all devices; additionally, any device on the network can talk to any other device, and all MIDI devices become resources available to multiple users at workstations or computers on the network.

Buxton's proposal is attractive, especially for those who work in large MIDI configurations. The MIDI industry would be well-advised to look closely at standardizing on some LAN configuration and protocol.

Ken C. Pohlmann: Technical Overview of the CD-I Format. Pohlmann introduces the *Compact Disc-Interactive* (CD-I) format. CD-I is an extension of CD-ROM, which allows for the storage of mass quantities of audio, video, text, and graphics information together with executable code allowing for presentation of the information to and interaction with the user. The potential of this medium is well-described by one of Pohlmann's hypothetical examples:

> A single CD-I disk could contain a biography of a composer, providing textual information, pictures, recorded examples of his music, played while the score is displayed, and a complete catalog of his musical works, as well as available recordings. . . .

Pohlmann describes the audio, video, and text formats available under the standard, as well as the standard operating system (CD-RTOS, a descendent of OS-9 running on 68000-family processors) and the hardware and interfaces employed in a CD-I system. Possible applications in education, training, entertainment, information retrieval and analysis, and other fields are also briefly discussed.

Stan Cornyn: CD-I and the Media. Pohlmann's presentation is immediately followed by Cornyn's evangelical sermon on the wonders of CD-I. The most important of these wonders is, evidently, that it will be highly lucrative for the entertainment industry. This is no doubt true; but one wonders why it needed to be said at such length during a supposedly technical conference.

Reviewed by John Duesenberry
Brookline, Massachusetts USA