CHAPTER 6:
ANALYSIS OF TECHNOLOGY COMPONENTS: BRUCE PENNYCOOK'S *PRAESCIO IV*, FOR EXTENDED CLARINET AND INTERACTIVE SYSTEM (1990)

In 1991, while researching the emerging field of real-time interactive computer music systems, I ran across Bruce Pennycook's description of his *Praescio* series of compositions in the *Computer Music Journal*. The concept of flexible, performer-controlled electronics, as outlined by Pennycook, was revelatory to me, and was a formative factor in my decision to pursue advanced study in computer music. At the Peabody Computer Music Studios, my digital music programming instructor and mentor was Ichiro Fujinaga, a former member of Pennycook's team at McGill University that had developed the interactive software system for the *Praescio* series. The performance aesthetic of this system, including expressive control of electronics, and flexibility determined by the musical sensibilities of a live performer rather than a fixed tape, were guiding principles in my computer music training at Peabody. *Praescio IV* is a work that was groundbreaking in its day, but is not playable in its original form due to technological obsolescence. It is an excellent counterpoint, musically and technologically, to the other works considered in this study.

6.1 HISTORICAL BACKGROUND

Bruce Pennycook's *Praescio IV* is part of a series of compositions written between 1989 and 1993 at McGill University using the custom-designed "MIDI-Live"

interactive computer music system.[60] *Praescio IV* was written in 1990 for clarinetist Jean-Guy Boisvert and premiered at the 1990 Clarinet Fest International in Quebec City. Boisvert performed *Praescio IV* subsequently at the 1991 International Computer Music Conference at McGill University. In 1995, the composer prepared an updated version of the software for Boisvert's Canadian concert tour. Boisvert has recorded *Praescio IV* on a compact disc titled *Zodiak/Zodiaque*.[61]

The MIDI-Live interactive system for *Praescio IV* consisted of software that captured live input from the performer via MIDI controllers and pitch-tracking hardware and produced electronic sound via MIDI-controlled synthesizers. The pitch tracking was originally accomplished using the IVL 4000 Pitchrider pitch-to-MIDI interface. Additional control of the system was accomplished using a custom-built harness that added MIDI control keys to the clarinet. A sound module (Emu Proteus I) generated the synthesized accompaniment in response to MIDI output from the system software, based on pre-recorded MIDI sequences and on direct control by the pitch tracking system (effectively turning the clarinet into a MIDI controller).

Pennycook's 1995 update of the Praescio IV software was developed in Max for the Apple Macintosh computer (then distributed by Opcode Systems). However, this version no longer runs on the current generation of Macintosh computers and another update is now necessary. The IVL 4000 Pitch Rider is also no longer available. Therefore the pitch-tracking functions required for *Praescio IV* must be accomplished by

---

[60] Bruce Pennycook, "Machine Songs II: The *Praescio* Series – Composition-Driven Interactive Software," *Computer Music Journal* 15, no. 3 (1991), 16 – 26.

[61] Jean-Guy Boisvert, *Zodiak/Zodiaque,* SNE Records #586, 1997.

other means. The Proteus sound module also may prove difficult to obtain for many performers and alternate means of tone generation may be necessary for performance of this piece. In fact, the composer recommends using a more up-to-date sound set using synthesizers chosen by the performer.

As of spring 2004, Pennycook has created a new update of the performance software using Max/MSP. Programmer Dale Stammen has updated an essential Max object (*playSMF*) used in the *Praescio IV* software to run on the current generation of Macintosh computers. I have now added a pitch-tracking component to this software that emulates the features of the IVL 4000 Pitchrider used in the original incarnation of this piece. While this current version was developed to run on late-model Apple Power-Macintosh computers (G3 – G5), the recent release of Max/MSP for Microsoft Windows XP means that the current software is theoretically cross-platform compatible.

Now that there is a functional version of the interactive system software, performances of *Praescio IV* are possible at least for the present and very near future. As demonstrated by the history of technological change over the relatively short lifetime of this work, it is likely that new versions of the performance software will have to be constructed again within a few years.

6.2 MUSICAL ROLE OF TECHNOLOGY

The interactive system used in *Praescio IV* functions as a separate instrument that accompanies and contrasts with the solo clarinet. All electronic sounds in *Praescio IV* are generated by a synthesizer, rather than through transformations of the clarinet sound.

The distinction between the acoustic and electronic instruments is quite clear. Pennycook describes the relationship of the clarinet to the synthesized sound in his program notes:

> The work explores the various relationships among the sonic resources - clarinet alone, synthetic sounds, and "colorized" clarinet where pitches from the clarinet are tracked by the system and enhanced with synthetic sounds. The interplay of these three resources is used to articulate the formal organization of the work.[62]

The *Praescio* series was conceived as an attempt to achieve meaningful interaction between live performers and electronic musical instruments and processes. Pennycook's aim was primarily "the removal of prerecorded tape from electroacoustic performance" by creating flexible computer-controlled MIDI systems.[63]

The interactive system produces sound both in direct response to actions of the performer and according to pre-programmed automated processes. Pacing is completely under the control of the performer who is therefore free to interpret the score in a flexible manner, rather than having to play to a fixed timeline.

A single player can perform *Praescio IV* without additional technical assistance. Although the interactive system plays an accompanying role, *Praescio IV* is essentially a solo work.

6.3 ANALYSIS OF TECHNOLOGY COMPONENTS

The technology requirements for *Praescio IV* are extensive, although current trends in computer music technology make it increasingly possible to reduce the overall

---

[62] Bruce Pennycook, *Praescio IV for extended clarinet and interactive system*. program note (Penntech Records, 1990)

[63] Pennycook, "Machine Songs II," 25

size of the performance setup by incorporating more functions into the system software. However the system is implemented, certain elements must be present, whether in physical or virtual form. The complete system consists of a sound input and output hardware, loudspeakers, microphones, input devices, a synthesizer, score-related data files, and interactive MIDI processing software.

6.3.1 Sound Reinforcement

The sound system requirements for *Praescio IV* are fairly standard. Two-channel playback from the synthesizer is recommended, though the composer gives no special instructions for loudspeaker placement and no sound spatialization effects are employed. The clarinet may be amplified slightly to balance with the electronics, but this is not an absolute requirement. A microphone is required to provide input to the interactive system from the clarinet for purposes of pitch tracking. For this, a contact microphone is recommended in order to isolate the clarinet sound as much as possible. Separate standard microphones might be used for amplification of the clarinet.

6.3.2 Control Interface

The interactive system for *Praescio IV* relies on live input signals from the performer to control playback of pre-recorded sequences, to initiate changes to various system parameters, and for direct expressive control over volume and sustain parameters of the synthesizer output. Four separate devices are used to provide these signals to the system: an event trigger, a pitch tracker, a sustain pedal, and a volume controller.

Event Trigger. A simple trigger signal is used to advance sequentially through a list of system events. This technique is not unlike using a remote control to advance through a slide show. Since the performer is responsible for triggering event changes, the device used should be convenient to operate from stage while playing. Events to be initiated by the trigger interface are notated in the score by the T symbol, as shown in figure 6.1, below:

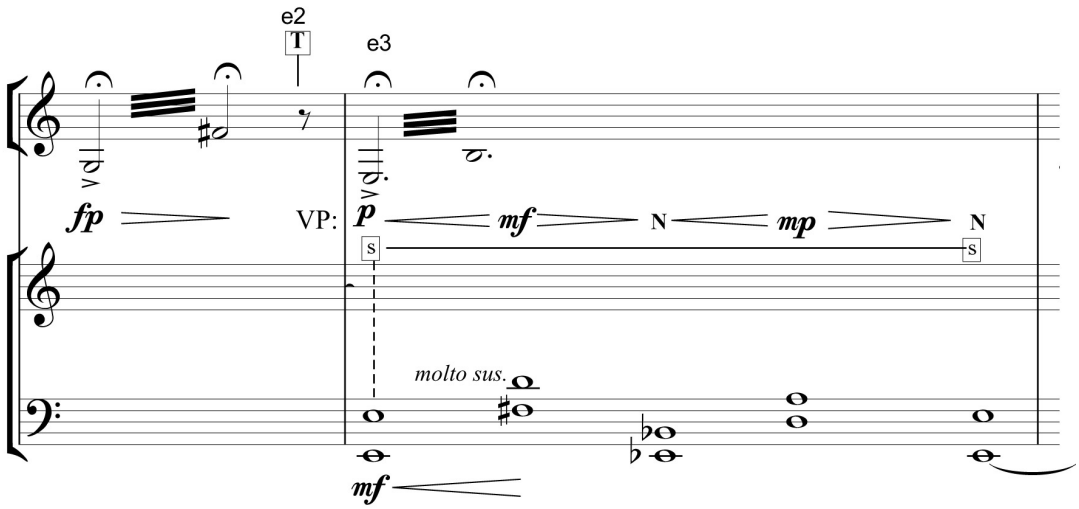

Figure 6.1. Score example: event 2 triggered by footswitch control

The 1990 version of *Praescio IV* used a custom-designed harness with extra keys mounted to the clarinet, designed by Pennycook and Eric Johnstone at McGill University.[64] Only one harness was ever built, so this is not a reasonable option for most clarinetists. While the clarinet-mounted key switches may have been ergonomically more convenient for the performer, in terms of system interaction they were no different

[64] Pennycook, "Machine Songs II," 24

from a simple MIDI footswitch. Adapting to available devices would in no way affect the functionality of the system.

Pitch Tracking. Pitch data derived from real-time analysis of the live clarinet signal is used for two purposes in *Praescio IV*. Pitch information is used to advance system events in the same way as the trigger, described above. For these events, the system waits for a particular pitch to be played before initiating the next event. Event 3 (shown in figure 6.1 above) is cued after event 2 is triggered by the foot pedal, but is not executed until the clarinetist plays the low E.

At certain event numbers the system follows the clarinet pitch and matches it with synthesized notes and parallel harmonies. The composer refers to this as "colorization" of the clarinet sound through synthesizer doubling. Attack volume (MIDI velocity) of the synthesized sound is derived from an analysis of the clarinet's attack amplitude for each note detected. These events are termed "THRU" events by the composer because MIDI data from the pitch tracker is passed directly "thru" to the synthesizer.[65]

Therefore, *Praescio IV* requires a pitch-tracking system that reliably tracks the clarinet and accurately identifies, in real time, the notes being played. The pitch-tracker must also measure attack amplitude and assign a numerical value. In MIDI terms, this would be a note velocity value between 0 and 127. The system should provide a way for

[65] The use of the word "THRU" by Pennycook is a reference to the MIDI specification, which defines IN, OUT, and THRU ports for MIDI devices. THRU ports pass data received at the MIDI IN without alteration or processing. The "THRU" events in *Praescio IV* involve processing and alteration of the pitch tracker data, and therefore Pennycook's use of the term "Thru" is inspired by, but not exactly in compliance with, the MIDI definition of THRU.

calibrating velocity values to the minimum and maximum attack dynamics produced by the individual clarinetist using his or her own microphone and audio input system, based on a pre-concert sound check.

Sustain Pedal. A footswitch or other device is required for turning on and off the sustain state of the synthesizer. Similar to the sustain pedal of a piano, MIDI sustain holds any notes that are initiated by Note-On messages to the synthesizer until sustain is released. The sustain pedal is indicated in the score with the symbol [s]. Like the event trigger described above, the 1990 version of *Praescio IV* used the custom harness, with an added fourth-finger key acting as a sustain pedal. Pennycook now recommends a simple MIDI footswitch.

Volume Controller. Several points in the score require direct dynamic control of the synthesizer output volume. All versions of the *Praescio IV* software to date have required a standard MIDI pedal to send volume messages to the synthesizer. Whether or not a MIDI pedal is used, the volume controller should smoothly control volume fades between minimum and maximum amplitudes for only the synthesizer channel to which it is assigned.

6.3.3 Synthesizer

Praescio IV requires a multi-timbral polyphonic synthesizer capable of playing back pre-programmed MIDI sequences (a list of synthesizer control data messages with associated playback timings) on at least 16 separate channels simultaneously. The aforementioned capabilities are currently standard for most professional-grade commercial synthesizers compatible with the MIDI specification. Alternatively, synthesis functions could be integrated into the interactive system software (using software-based synthesizers or samplers), obviating the need for an external sound module and streamlining the equipment needed on stage.

The E-Mu Proteus-I was a popular sample-based sound module found in many recording, production, and project studios. A list of Proteus programs with MIDI channel assignments, is included with the current version of the performance software.

Rather than replicating the sounds used in the original version, the composer leaves synthesizer voicing up to the performer as long as basic sound types are used for the appropriate channels (sustained, percussive, etc.). Table 6.1 (below) gives Pennycook's general suggestions for voicing each channel of the synthesizer. This guideline is based on the Proteus I sound set that is listed in a sub-patcher of the current Max/MSP software. Pennycook recommends a richer sound set than what was possible with the original Proteus I, and suggested Boisvert's recording as an audible guide.[66] In any case, synthesizer voice programming is the performer's prerogative in this piece.

---

[66] Bruce Pennycook, email to the author, March 17, 2004.

Table 6.1. Proteus I sound set for *Praescio IV*

Channel	Voice type	General description
1	Pad	"Spacey" FM-type sustained sound
2	Piano	Concert grand piano sampled in stereo
3	Pad	"Hard-edged" pad
4	Pad	"Stringy" pad, big
5	Bell	"Tingly," bell-like with sustain
6	Pad	"Hard-edged" pad, like #3, but more edge
7	Piano	Concert grand piano sampled in stereo
8	Brass	"Cheesy" synthesized trombones
9	String	Big flanged strings
10	Piano/pad	Piano with synthesizer sustain tails (long decay)
11	Percussion	Vibes
12	Pad	"Airy," "spacey" pad – diffuse, with slow attack and decay
13	Brass	French horn like with light flange
14	String	Big flanged strings (as #9 above)
15	Guitar	A "stratocaster" guitar sound
16	Pad	FM-type sustain used for the clarinet THRU sustains

6.3.4 Prepared Data

The interactive system for *Praescio IV* operates according to a set of data files that are specific to the piece, analogous to a traditional score. An "event list" stores a series of cues for system actions and parameter changes tied to specific transition points in the score. In addition, a collection of pre-recorded MIDI sequences (stored as Standard MIDI Files) drive synthesizer playback according to parameters stored in the event list.

Event List. For each of the seventy-seven event points in the score, the event list specifies a trigger condition (either a signal from the footswitch or a specific note played by the clarinetist) and a series of parameters for either controlling SMF playback ("Play"

events) or for defining colorization/harmonization settings ("THRU" events). An excerpt from the event list is given in figure 6.2, below. The complete event list for *Praescio IV* can be found in appendix B (with kind permission from the composer).

Table 6.2. *Praescio IV* event list sample

Ev.	Trigger:		Play Event Parameters:						THRU Event Parameters:				
#	T	P	Chan	Track	Trans	Harm	Vel	Temp	H1	H2	H3	H4	Vel
1	X		-	-	-	-	-	-	-	-	-	-	
2	X		-	-	-	-	-	-	0	-	-	-	100
3		50	5	3a	-2	-	100	100	"	-	-	-	"
			4	3b	-2	-	"	"	"	-	-	-	"
			5	3c	-2	-	"	"	"	-	-	-	"
4		64	4	3	-2	-	"	"	-	-	-	-	-
5		59	6	5a	-4	-	"	"	-	-	-	-	-
			1	5b	-2	-	"	"	-	-	-	-	-

MIDI Sequences. Like the event list, a series of short, pre-recorded sequences of notes and synthesizer control data serve as electronic "score excerpts" for the synthesizer. Unlike the event list, the sequences are extremely detailed, specifying precisely which notes to play, and how and when to play them.

The prerecorded sequences are stored as multi-track standard MIDI files (SMF, type 1 files).[67] Individual sequences for each event are stored as separate tracks within the file. Counting each track as a separate sequence (which is indeed how they are used), there are a total of 73 short prerecorded MIDI sequences (although a few tracks are in fact duplicates of others).

[67] A brief overview of MIDI is given in chapter 2

The first MIDI sequence used in *Praescio IV*, sequence 3a, is extremely short, consisting of a series of five two-note chords. The sequence data contained in this file is shown in table 6.3.

Table 6.3. Sequence 3a in MIDI event list format

Time[68]	Event	Duration	End
1|1|000	Note 52 (E2), velocity 126	7|160	2|4|160
	Note 40 (E1), velocity 126	7|160	2|4|160
2|3|320	Note 50 (D2), velocity 126	9|080	4|4|400
	Note 42 (F#1), velocity 126	9|080	4|4|400
4|4|000	Note 46 (Bb1), velocity 126	10|477	7|2|477
	Note 27 (Eb0), velocity 126	10|477	7|2|477
7|1|477	Note 45 (A1), velocity 126	8|340	9|2|337
	Note 38 (D1), velocity 126	8|340	9|2|337
9|1|437	Note 40 (E1), velocity 126	14|320	12|4|277
	Note 28 (E0), velocity 126	14|320	12|4|277

This same sequence of notes, rendered in standard musical notation, is shown as it appears in the score in figure 6.2.

[68] Time in this MIDI sequence is given in a "measure | beat | tick" format used by most commercial MIDI sequencing programs. In this case, there are 480 ticks per quarter note beat, and four beats to a measure. Timings are therefore relative to the tempo, defined in the file header, but subject to variation on playback. Corresponding Note Off messages are given as "End" times, and durations are calculated as "beat | tick"

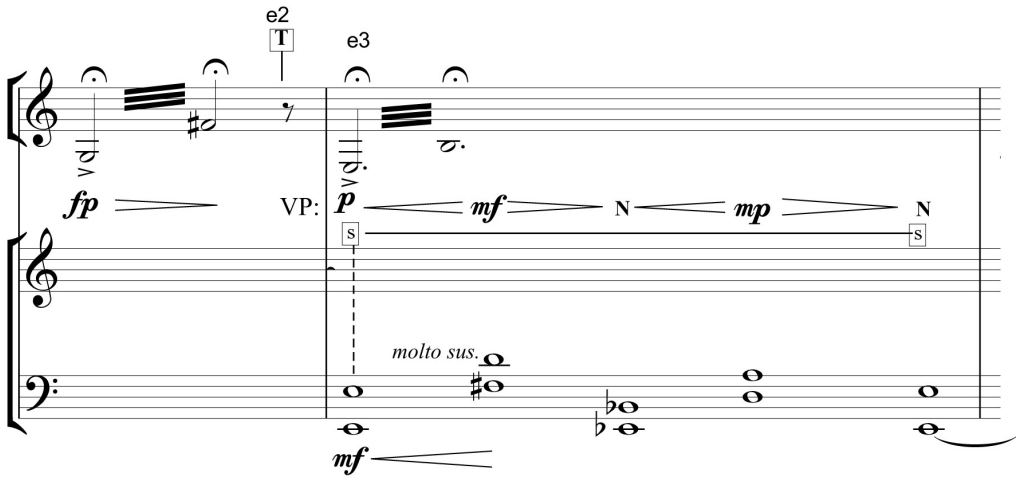

Figure 6.2. Score example: sequence 3a as shown in the score

Because MIDI is an internationally accepted standard protocol for synthesizer control now in use for more than 20 years, storage of sequence data in this format is reasonably secure, even if future realizations of *Praescio IV* are not MIDI-based. Problems with digital file storage and retrieval can be easily avoided by formatting (and possibly printing) data files in standard plain-text format, similar to the format given in Figure 4. Due to the sheer quantity of data, and the composer's copyright, the complete set of MIDI sequences required for *Praescio IV* will not be included here in its entirety.

6.3.5 Event Processing

Score events in *Praescio IV* are processed in three distinct stages: input processing, "play" event processing, and "thru" event processing. Input processing controls the advancement through the event list sequence in response to MIDI input received from the foot switch trigger and the pitch tracker. The play event processor

controls the playback of SMF sequences for each event. The thru event processor controls the synthesizer colorization of the clarinet (passing tracked pitches as MIDI note messages "thru" to the synthesizer).

Input Processing. For each event in the list, a trigger condition must be met before the event parameters are sent to the play or thru event processing stages. Trigger conditions are either a signal from the footswitch (i.e., MIDI controller 64, value 127), or a specific note number received from the pitch tracker. The input processor must look ahead to the upcoming event in the list and set the appropriate trigger condition. When the appropriate trigger signal is received, the list of parameters stored for that event are sent out to the play and thru event processors.

Play Event Processing. The Play event processor interprets a list of playback parameters for each event in order to execute playback of MIDI data stored in the SMF sequences. The event list specifies a track name, a MIDI playback channel (1-16), transposition level (+/- *n* semitones), harmonization (+/- *n* semitones), velocity scaling (% of stored values), and tempo (% of stored tempo). Several sequences may be played back simultaneously, and the same sequence may be played in multiple simultaneous instances at various transposition levels, dynamic levels, and tempi.

Several sequences are used in more than one event, although playback parameters may be changed. For example, events 35 – 38 all use the same sequence data ("tr38," or

track 38—a rapid single-note sextuplet ostinato, with a gradual decrescendo). Changes to transposition, velocity, and tempo variables alter playback, as shown in figure 6.4:

Table 6.4. Event list excerpt: events 35-38

Event	Action
35	play tr38 chan 8
36	play tr38 chan 8 trans 8 vel 0.6
37	play tr38 chan 8 trans -3 vel 0.8 tempo 1.1
38	play tr38 chan 8 trans -2 vel 0.5 tempo 0.9

Event 35 plays back the MIDI sequence "tr38" on synthesizer channel 8 with no alterations. Event 36 plays the same sequence a minor 6th higher (+8 semitones) with all velocity values scaled by 60% (x 0.6). Event 37 plays the sequence a minor 3rd below the original (-3 semitones), with velocities scaled to 80% and tempo slowed by scaling durations to 110%. Event 38 transposes the same sequence two semitones below the original, cuts velocities by half (0.5), and shortens durations to 90% of their original values.

This synthesizer output is only roughly notated in the score in order to give a simplified version of the electronic accompaniment as a convenience to the performer. Although standard musical notation is not as precise as the MIDI sequence data itself, the rendering of this passage in the score provides an easily readable reference of the synthesizer activity, as shown in figure 6.3:

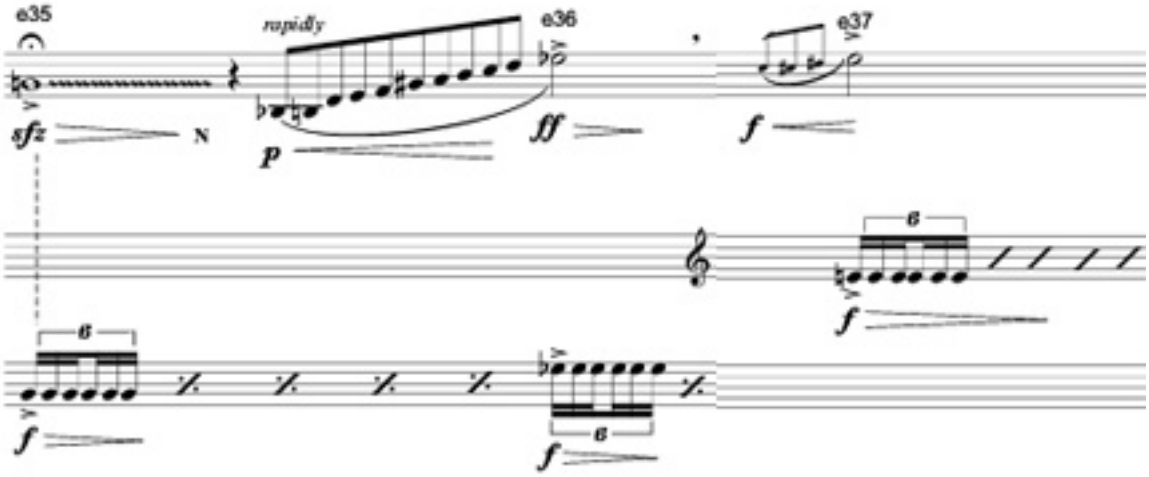

Figure 6.3. Score example: events 35 - 37 as notated in the score

Play events are formatted in Pennycook's most recent performance software for the custom-built Max external object playSMF. PlaySMF was created in 1993 by Pennycook, Dale Stammen, and Basil Hilborn for *Praescio VI* (for flute and interactive system) and was used in the 1995 version of *Praescio IV* performed on tour by Boisvert.[69] The current software uses an updated and recompiled version of playSMF (as a Max4/MSP2 external object for Macintosh PPC computers) completed by Dale Stammen in February 2004.

THRU Event Processing. Score Events that apply synthetic "colorization" (doubling or harmonization) to the clarinet are designated as "thru events" because they pass the clarinet pitch data "thru" to the synthesizer on MIDI channel 16, playing the synthesizer as if by keyboard control. Because note data sent to the synthesizer from the pitch

[69] Pennycook, Old Problems New Solutions

tracker is mediated by the interactive software and controlled by parameters in the event list, pitch data can be altered or manipulated before it reaches the output. The thru event processor manages harmonizations by matching each incoming note with up to four pitches at variable transposition intervals, expressed in semitones (or more accurately, as MIDI note number offsets). The harmonizations are based on a list of variable settings controlled by the event list. Figure 6.4 shows the synthesizer simply doubling the clarinet at event 12 (sequence 11 continues playing):

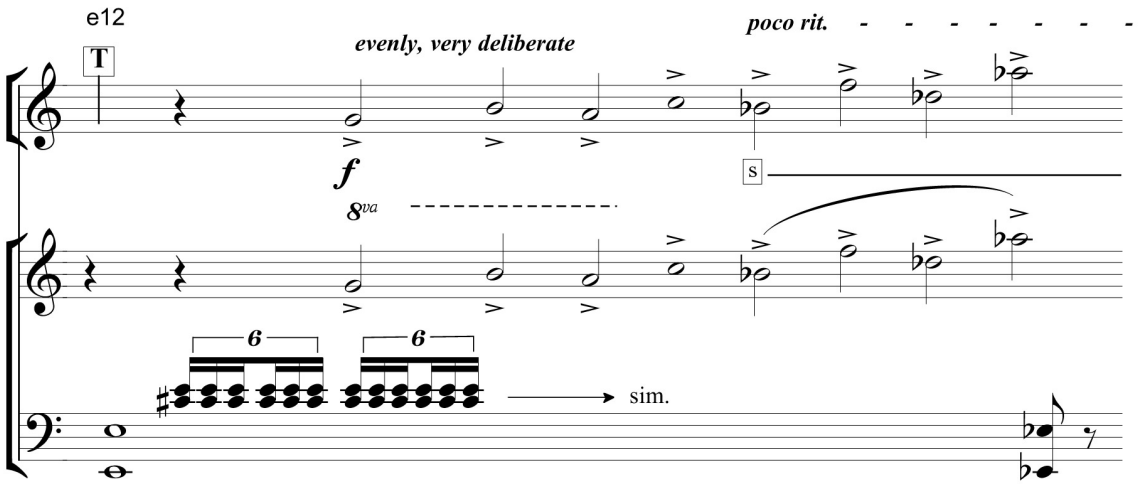

Figure 6.4. Score example: parallel tracking of the clarinet by the synthesizer (event 12)

Event 29 is more complex. The clarinet G3 is harmonized by notes at intervals of 0 (G3), 1 (Ab3), -7 (C3), and –18 (C#2):

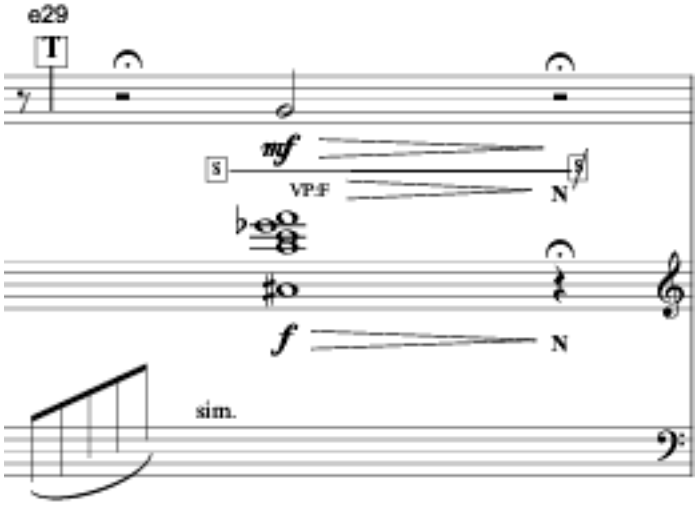

Figure 6.5. Score example: chordal harmonization of clarinet pitch

6.4 SUMMARY

On close examination, the interactive system used in Praescio IV can be broken into two main parts: a general-purpose MIDI data processing and synthesizer control system, and a set of data files (MIDI files and specially formatted event lists) that are specific to the piece and act as an electronic "score." The entire system is controlled by real-time inputs from the performer, in the form of footswitch signals and pitch data collected by a pitch-tracking module. The main role of the system software is to interpret incoming control signals, parse commands from the event list, playback recorded MIDI sequences on cue, manage the colorization/harmonization of the live clarinet, and route MIDI messages to the appropriate synthesizer channels. A diagram of the complete processing system is shown in figure 6.6.

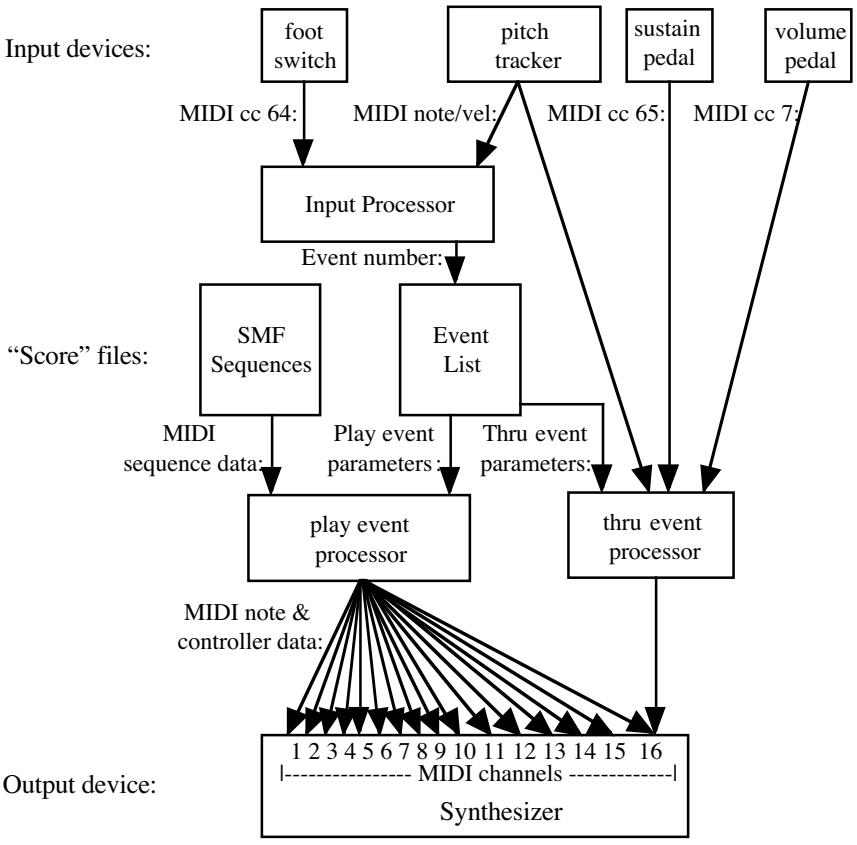

Figure 6.6. Diagram of the *Praescio IV* interactive MIDI system

Implementation of *Praescio IV* for newer technology or for competing computer platforms could be a fairly straightforward process, so long as the technology used is capable of the input and MIDI file processing outlined above. However, any realization of this work will require a complete set of Pennycook's SMF sequences in addition to the clarinetist's score. These items are currently distributed in electronic format with the performance software upon request to the composer.