New Public-Domain Realizations of Standard Pieces for Instruments and Live Electronics *

Miller Puckette
Department of Music and Center for Research in Computing and the Arts, UCSD

Abstract

We describe here our efforts to make it easier to perform certain well-known pieces from the live electronic music repertory, without resorting to special hardware or proprietary software. We hope that these realizations of the electronic parts of the pieces will be longer-lasting than previous realizations have been, and that they will make it possible to perform the music with a much smaller investment in time and equipment than before.

1 Introduction

The last 35 years have seen the development of a significant repertory of music involving concert instruments whose sound is enhanced or transformed using live electronics. The realizations of many of these pieces have depended on specific items of hardware or software which, while chosen for their expediency at the times of the premieres of the pieces, will eventually become impossible to find, and in some cases are already becoming scarce.

We are at various stages in preparing reference realizations of four important pieces from this repertory. We intend over time to expand the collection to at least a dozen pieces for solo instrument or small ensemble and electronics.

In addition to making it much easier to perform these specific pieces, we hope to fill three other important functions. First, the realizations will serve to document the pieces in a way that will be useful to musicologists. Second, they will serve as a model showing how one might realize pieces involving real-time electronics in a less ephemeral way than is now often the practice. Finally, we hope to attain a higher level of audio quality than in previous realizations.

The new realizations are based on generic hardware running Linux. Our choice of Linux was made on the basis of the great stability of the "*nix" operating systems in time (many programs from the early 80s still compile and run today), and also because we have found Linux particularly well adapted to real-time interactive audio synthesis and processing.

We have chosen Pd [4] as our real-time environment, because it is available with source so that we will be able to recompile it at will in the future, and for its platform independence. (Another acceptable choice would have been jMax.) Pd also runs in Windows and will probably soon run in Mac OSX, for those who don't want to run Linux just yet.

We plan to archive performance materiels for the pieces, along with a suitable version of Pd and relevant documentation, on the Web in the form of text files, HTML, and ".wav" soundfiles where needed. These formats seem unlikely ever to become unreadable. We have already performed three of the pieces (essential as proof of the validity of the implementations) and plan to perform the other and make some further revisions within the next year.

2 Repertory

Each of the four pieces we have worked on so far involves its own special considerations:

2.1. Karlheinz Stockhausen, *Mantra* (1970).

*Reprinted from *Proceedings*, ICMC 2001.

This piece was originally scored for two pianos, each pianist also controlling a ring modulator. The original ring modulators apparently had very large frequency selection dials on which the pianists were called on to select frequencies varying from about 10 Hz. to 4 Khz., but also sometimes matching pitches of the piano so that accuracies better than 1 Hz. are needed. Even if the pianists were actually able to select frequencies accurately in this way (which seems doubtful) we have made what we think is a much better implementation in which MIDI sliders, still controlled by the pianists, are given appropriate ranges depending on where they are in the piece. Of course, ring modulation itself offers no great challenge to a computer other than controlling latency, which is particularly critical in this piece, and which we have held to less than 10 milliseconds. We have performed this piece twice so far in public.

Mantra also calls for crotales and a shortwave radio which one performer tunes to any frequency where Morse code can be heard. Such frequencies being rare today, we use a CD recording instead.

We performed this piece twice in 2001; the electronics were relized by Kerry Hagan with help from the author and Shahrokh Yadegari. The implementation of this piece gives rise to no copyright concerns; once we get the patch finalized and documented (hopefully before these *Proceedings* appear) we will distribute the materiels freely.

2.2. Pierre Boulez, *Dialogue de l'Ombre Double* (1985; produced at IRCAM). This piece in its published form is performed by one clarinetist accompanied by a tape of the same clarinetist. The live and tape sounds are variously processed by an artificial reverberator, a "piano reverb" (made by playing the sound through a loudspeaker underneath a piano whose sustain pedal is held down, and capturing the string resonances with a microphone), and six-channel panning in a variety of patterns.

According to folk tales around IRCAM, *Dialogue* was originally conceived as a piece for two clarinetists, one offstage who would play the sections of the piece which finally ended up as the tape part. The tape part's spatialization features very quick changes which would require score following to synchronize with a live clarinetist, and it was apparently at least partly for this reason that the offstage clarinetist was replaced by the tape. As shown in Fig. 1, the tape also sent a synchronization signal to a sequencer controlling an array of VCAs to spatialize the tape.

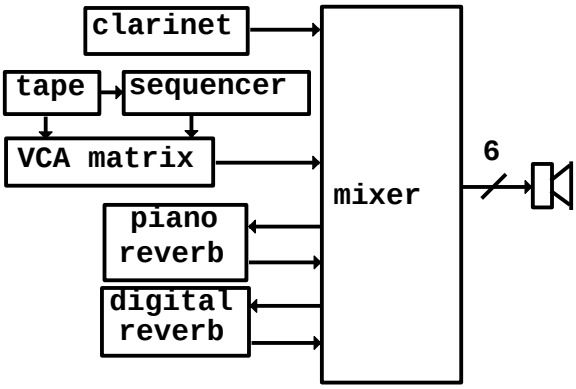

Figure 1: Traditional realization of *Dialogue*

We have made a new two-clarinet realization of this piece, using automatic score following to synchronize the spatialization with the offstage clarinet. We performed this version three times in 1996 with Patrick O'Keefe and Robert Zelickman on clarinet. At that date the piece required a fair amount of outboard gear; also, since jMax and Pd were not available, our realization used Max 0.26 on an SGI machine. This patch has since been imported into Pd and the outboard gear replaced with signal processing patches within Pd. We have not yet had the opportunity to validate the new Pd patch in a concert setting.

We have developed an interesting FFT-based patch that imitates the piano reverb effect. Although it does not replicate the effect exactly, it exhibits the same sort of tuned resonance that the piano does, and is interesting enough in its own right that we feel justified in making the patch offer this feature as an optional alternative to the piano. (The piano reverb effect is by far the trickiest aspect of producing this piece. The piano must be placed in an acoustically quiet room that nobody can enter during the performance, and the speaker and microphone(s) must be carefully placed to avoid feedthrough without sacrificing sound quality.) We feel that the new effect is

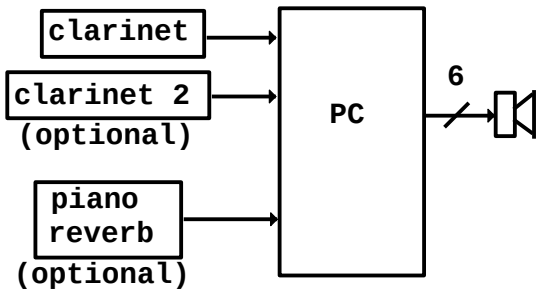

Figure 2: *Dialogue* in Pd

true enough to the intended effect that it can justifiably be used in performances; whether to use the electronic version or a real piano should be up the the discretion of the presenter.

The overall concert configuration is as shown in Fig. 2. The piece can be played with or without the second clarinetist (who can be replaced via playback from the PC's disk) and with or without a real piano. We therefore achieve much more flexibility than in the original realization, with configurations which may be somewhat, or even radically, simpler than the original.

As in the case of *Mantra* there are no copyright concerns in publishing our realization of this piece. On the other hand, score following does present a potential problem because of Roger Dannenberg's 1985 patent. Although we don't believe our published score following algorithm infringes this patent, we would like to encorporate Dannenberg's more robust algorithm [2] before publishing our realization. Since the patent expires in 2002 the most prudent course for us will be to wait until then to publish. (This same consideration comes into play in the case of *Pluton*; the piece was first realized using the score follower that has been included as part of the Max program since its first publication [3], but a truly robust version of the piece will have to wait until we can use Dannenberg's algorithm.)

2.3. Philippe Manoury, *Pluton* (1988; IRCAM). This piece is especially interesting from our standpoint since it was the first musical production ever to use Max (although another piece, by Philippe Durieux, made it to stage first). The piece is for solo piano and live electronics. In its first incarnation, the patch (on a Macintosh II) controlled a 4X which carried out the audio processing. In 1991 *Pluton* was ported to the ISPW, which permitted the audio processing and the "control" to be unified in a single Max/FTS patch. This version has been played at least once outside IRCAM, using "Max/FTS 0.26" on an SGI machine. IRCAM has dropped Max/FTS but maintains *Pluton* on jMax; so its inclusion here is meant more as a benchmark for Pd than as a means of keeping *Pluton* in the repertory, although it is always beneficial to have more than one possible implementation of such a milestone in the development of live computer music practice.

Importing the Max/FTS patch into Pd has been a straightforward task and it runs in real time on recent-vintage off-the-shelf PCs. We plan to perform *Pluton* using the Pd patch during the 2001-2002 academic year.

As compared to the previous two patches, *Pluton* gives rise to a copyright difficulty: it's by no means clear to whom the original patch belongs (IRCAM or Manoury) and the Pd patch, although different from the IRCAM one, is closely based on it. Whether we can publish this patch on the same basis as the others described here will have to be negotiated with both Manoury and IRCAM.

2.4. Kaija Saariaho, *Noanoa*(1991; IRCAM). This piece, for flute and live electronics, is published with an accompanying CDROM, which contains a Max patch and 33 soundfiles. The Max patch plays the soundfiles on cue and controls two "effects boxes" which are now obsolete. The "MIDI system dumps" for these effects boxes are included on the CDROM, but are of course useless. By listening to Camilla Hoitenga's published recording of the piece we reverse engineered the two effects involved and recreated them in Pd. We have given this piece five performances so far, with Tara O'Connor and Lisa Cella playing the flute part.

Here, since the CDROM which accompanies the published score contains the necessary soundfiles, we can simply publish the Pd patch without the soundfiles but with instructions on how to copy the files off the CDROM.

3 Conclusions

All of these pieces had previously been re-created outside their original places of realization using a variety of equipment. The intent of the work reported here is to make our realizations as easily reproducible, and as independent of proprietary technologies, as possible.

The pieces considered here all come from Europe, which has offered much stronger institutional backing to electronic music productions than has the rest of the world; as a result the pieces described here are all well documented and the necessary materiels relatively easy to find. In the future, however, as our repertory expands at a rate of perhaps one or two pieces a year, we hope to see the widest possible geographical distribution of artists represented. One important factor weighing in our favor is that the plunging cost of computer music hardware and the open software movement are decentralizing the practice of computer music.

Our future choices will also be guided by the enthusiasms of performers, and we actively seek collaborations with them. We hope eventually that both we and others will pursue this work until it can be said that there is a real repertory of performable music with live electronics.

Three of these four pieces are also part of a graduate music course offered at UCSD in which students learn the techniques underlying classic computer music pieces by realizing their own sketches. In the instance of *Pluton* we have engineered the Pd patch so that it can both be used to play the piece and to load students' own (possibly score-following-based) sequences. In this context it has also been valuable to make "study realizations" of parts or aspects of tape pieces such as Harvey's *Mortuos Plango, Vivos Voco* and Chowning's *Stria*. Our goal is to integrate the four "performance patches" described here into a larger series of "teaching patches."

Another direction of possible extension is into the field of visual music. Mark Danks's GEM [1], which extends Pd to handle graphics and image processing, is becoming popular as a way to tie graphical and audio processes together. Visual music repertory pieces might prove more difficult to realize in a PC than pure computer music pieces, but it may be that this is only a matter of waiting for PCs to catch up to the required performance levels.

References

[1] Danks, M., 2001. *GEM.* `http://www.danks.org/mark/`

[2] Dannenberg, R.B. (1984). "An On-Line Algorithm for Real-Time Accompaniment," Proceedings, International Computer Music Conference. San Francisco: Computer Music Association, pp. 193-198.

[3] Puckette, M., and Lippe, A. C., 1992. "Score Following in Practice," Proceedings, International Computer Music Conference. San Francisco: Computer Music Association, pp. 182-185.

[4] Puckette, M., 2001. *Pd.* `http://crca.ucsd.edu/msp`