Musical Performance Using the IRCAM Workstation

Cort Lippe and Miller Puckette
IRCAM, 31 Rue St. Merri, 75004 Paris, France

At ICMC-1990, IRCAM presented the IRCAM Musical Workstation then under development. Since then, we have begun work on musical tools and the realization of musical sketches. Miller Puckette has written a version of MAX for the Workstation which includes signal processing objects, as well as the standard objects found in the Macintosh version of MAX (including MIDI objects). The MAX environment for the Workstation is quickly turning into a powerful realtime tool for signal processing and synthesis.

The IRCAM Musical Workstation [1] is built around a dual Intel i860 card developed at IRCAM, and commercialized by ARIEL, which plugs into a NeXT machine cube. The Workstation attempts to break with the paradigm whereby a DSP is controlled by a separate host computer. The i860 is a general-purpose RISC processor - in other words general enough and yet fast enough to do control and synthesis at the same time.

MAX [2], as developed by Miller Puckette, extended by David Zicarelli, and commercialized by Opcode, is a graphical programming environment for the Macintosh with MIDI handles. MAX now runs on the IRCAM Musical Workstation with an additional library of signal processing objects. Presently, Puckette has written about 30 signal processing objects. This should increase to 50 or 60 in the future and includes objects for filtering, sampling, pitch tracking, delay lines, FFTs, etc. MAX runs in the FTS [3] real-time monitor program under the CPOS operating system [4].

With the Workstation version of MAX, the flexibility with which one can create control patches in the original version of MAX is carried over into the domain of signal processing. Creating an oscillator object and a DAC object, hooking them together, and

controlling the oscillator's frequency and amplitude via a MIDI keyboard in realtime becomes an easy task. Some of our early experiences have consisted of porting compositons made at IRCAM using the 4X to the Workstation. This has meant porting the 4X code, the control code from the 68000 host machine which controls the 4X, and finally MAX programs which run on a Macintosh and communicate with the 4X host via MIDI. Reducing all this code from a network of three machines to a single machine and a single environment on the Workstation has been quite successful. One module that was coded across the three machines which allows for independence of pitch and time in playback of sampled sounds was reduced to a rather simple MAX patch upon being transferred to the Workstation.

A typical 4X piece making use of a variety of modules for signal processing and sound synthesis might contain the following: harmonizers, delay lines, frequency shifters, samplers (with recording and playback in realtime), one or two synthesis algorithms such as additive synthesis or *Phase Aligned Formant Synthesis* (developed by Puckette) , filtering, reverberation, spatialisation, and possibly an FFT (for the analysis of incoming signals in order to extract control information for additive synthesis). Finally, a crossbar enabling all signals to be sent from any module to any other is normally included to maximize the number of possible signal paths. A configuration of this complexity for the composition *Pluton* [5] by Philippe Manoury (originally a 50 minute piano/4X piece) has been ported to the Workstation.

The signal crossbar between modules in the MAX patch for *Pluton* appears in Figure 1. The outlets of the patcher boxes on the top row are the signal outlets of the named signal processing modules. Notice that the outlets from each module go into the inlets of all the patcher boxes below to insure that all signals are sent from the output of every module to the input of every module. The frequency shifter module which includes its crossbar as well as the frequency shifting algorithm are shown in Figure 2.

One of the signal processing objects recently developed in MAX on the Workstation is a pitch tracking algorithm. Tests with a clarinet and a Zeta violin have given positive

results. For a clarinet sketch produced by Lippe, the pitch detection algorithm analyzes the incoming clarinet signal and outputs MIDI results which are sent to a score follower using the Explode object [6] which in turn triggers and controls the electronic events in the score. The pitch follower also outputs continuous control information analogous to pitch bend in the MIDI domain. This, coupled with envelope following for articulation detection allows for a fair amount of continuous control information coming from the clarinet.

Future developments in MAX include the implementation of a vocoder, realtime direct-to-disk soundfile playback, as well as interfaces developed in ANIMAL [7], an interface prototyping environment designed to control MAX patches.

REFERENCES

[1] Eric Lindeman *et al,* "The IRCAM Musical Workstation: Hardware Overview and Signal Processing Features", Proceedings, ICMC 1990 (Glasgow).

[2] Miller Puckette, "The Patcher", Proceedings, ICMC 1988 (Cologne).

[3] Miller Puckette, "A Real-time Monitor for Multiprocessor Music Synthesis", To appear in the Computer Music Journal 15(3)

[4] Eric Viara, "A Real-time Operating System for Computer Music", Proceedings, ICMC 1990 (Glasgow).

[5] Cort Lippe, "A Technical Discription of *Pluton*: by Philippe Manoury" IRCAM Annual Report 1988, IRCAM, Paris

[6] Miller Puckette, "EXPLODE: A User Interface for Sequencing and Score Following", Proceedings, ICMC 1990 (Glasgow).

[7] Eric Lindeman, "ANIMAL: A Rapid Prototyping Environment for Computer Music Systems", Proceedings, ICMC 1990 (Glasgow).

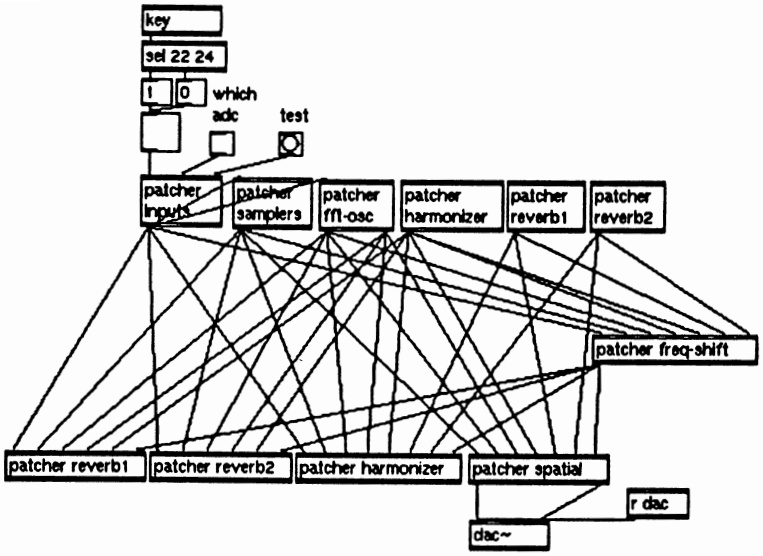

Figure 1. Signal processing crossbar in MAX.

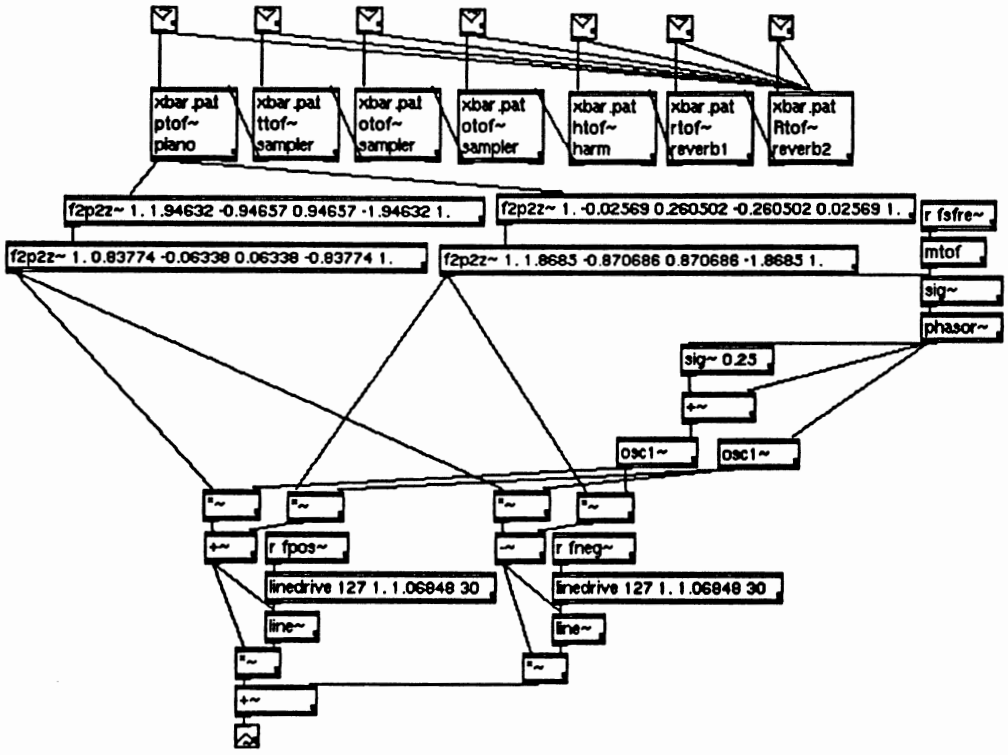

Figure 2. Frequency shifter and its crossbar.