The IRCAM Signal Processing WorkStation. A Composer/Performer's View.
Journal of Electroacoustic Music, Vol. 7 - Oct, 1993

Lawrence Casserley

Introduction

In 1991, with nearly 25 years' experience of Live Electroacoustic Music, and over 10 years developing experimental Real Time Digital Signal Processing Systems (Casserley, 1980, 1985, 1986), I made the decision to place an order for the new IRCAM Signal Processing Workstation. It is not the intention of this article to discuss the details of the ISPW or the MAX Environment, as these have been described elsewhere (eg, Lippe, 1991; Lindeman, 1991; Opcode, 1990; Puckette, 1988, 1991). Rather, it is intended to place the ISPW into the context of my philosophy and background as a musician, and to discuss the importance of the ISPW to my work. The article will conclude with some examples of ISPW applications and a brief discussion of the future implications of the ISPW system.

Philosophy and Background

From my earliest work in electronic music in the late sixties, the multi-media group Hydra in the seventies, Nettlefold Festival, Tube Sculpture, Electroacoustic Cabaret and Music in Colourscape in the eighties, and up to the present, my work has been dominated by three ideals: that a professional electronic musician, like a professional 'cellist, should have his own instrument; that the development of the instrument was crucial to the development of my music; and, above all, that actually performing the music in front of an audience was essential. The details of the experience that has led to and developed these ideals are beyond the scope of this article, but one or two examples will help to explain the importance of the ISPW to my work.

Solos, Commentaries and Integrations, 1969, was my earliest live electronic work, for clarinet with live treatments, percussion and tape[1]. Clarinet processing involved amplification, reverberation and ring modulation, implemented on a VCS1, forerunner of the VCS3.

Ritual Dances, 1991, with its large number of constantly- changing processes, is one of several works stretching the capabilities of commercially available equipment. It was made possible by the use of a 360 Systems 16x16 MIDI-controlled switching matrix, but was a complex and tricky system to set up, even so (Figure 1). The matrix allowed me to switch instantly from one routing, of instruments to transformations to speakers, to another, and also send program change messages to the SPX1000s.

Practical Experience of performing with the ISPW

After eighteen months experience with the ISPW I can identify three important areas in which it has affected my work.

First, the ISPW greatly simplifies setting up for concerts and rehearsals, saving valuable time and energy, which can be focused where it should be, on the music. Programmes that would, in the past, have required impossibly complex suites of equipment, can now be

planned entire lyon musical, rather than practical grounds. The processing power available is equivalent to that of several MIDI processor units, further simplifying large setups.

Second, the MAX environment allows complex patches to be built without becoming embroiled in the minutiae of programming, yet without the severe inflexibility of commercial DSP units. The composer can build viable networks quickly, modify and extend them, until the final patch is arrived at. Again, one can focus more easily on the musical results, rather than the technicalities of the implementation. The ease of modification of patches and virtually instantaneous response also encourages experimentation.

Third, the ISPW allows new possibilities for real-time transformation that have not previously been available. The examples that follow will demonstrate some of these advantages.

Example Programmes

During September, 1993 the ISPW was used in two concert programmes which sen/e to demonstrate the versatility and simplicity of the system. On 11 September, at the 1993 Nettlefold Festival, performances of *Siwrnai - Odyssey of Light* (1992) were alternated with a programme of three works performed by tubist Melvyn Poore. *Siwrnai*, a collaborative work involving Simon Desorgher, Melvyn Poore, mime artist Ian Cameron and myself, uses a patch with several separate treatments each for flute and tuba, using multiple flanger banks, transposition, modulation and frequency shifting. To make performance easier a control panel was designed (Figure 2) which, together with a MIDI fader unit, provides all that the performer needs to control the patch. The ability to design control panels to suit each application is one of the strengths of MAX as an interface for live performance.

Melvyn Poore's programme comprised three works: his classic delay piece, *123*, a realisation of Stockhausen's *Solo* and the first performance of my new work *UbAtAbU*, which will be discussed below. Poore has made not only a realisation of Solo, but a complete implementation of the delay system, together with all the controls for inputs and feedback, tuba processing and even a cue generator to keep him in sync with the system. Four very different works were performed on one piece of equipment, simply by loading a new patch for each piece.

On 28 September, at the Royal College ofMusic, London, in a concert to mark the opening of the new MMus(RCM) in Electroacoustic Performance and Composition, four works were performed. My own *UbAtAbU*, for tuba and ISPW, is my most ambitious ISPW implementation to date. There is not space in this article for a detailed description of *UbAtAbU*, but I would like to illustrate three of the processes involved which would have been impossible (or at least impractical) without the ISPW. In the opening section (Figure 3) I use a pair of three-tap delay systems with feedback to build chords from the tuba's harmonically-tuned melody. Each chord uses a different set of delay times, and these and all level parameters are controlled by a 'qlist' object in MAX. Another section uses a real-time spectrum stretcher/compressor made by cascading a hilbert transform (single sideband modulator) and a harmoniser. In order to know how much to shift the frequency, the hilbert transform is controlled by a pitch tracker. The operator's control parameter is a stretch ratio (from 0.25 to 2.2). Finally, a real-time convolution is implemented (Figure 4, with acknowledgements to Settel and Lippe, 1993) utilising

two FFTs, one fed by the tuba signal, the other by an oscillator bank replaying the harmonic structure of the opening section. The result of the convolution is resynthesised by an inverse FFT object.

Stephen Montague's *The Eyes of Ambush* is another implementation of a classic delay piece (Figure 5). Two independent delay systems, one with three taps, the other a single tap with feedback, were quickly programmed, easily modified and instantly recalled. Traditional tape delays would require six tape recorders, or it could be implemented on three SPX1000 processors. Edwin Roxburgh's *At the Still Point of the Turning World....* is one of a number of works commissioned by the West Square Ensemble for Barry Anderson's delay table system (Emmerson, 1991). It is intended that all of these works will be ported to ISPW and performed as a tribute to Barry. I hope to discuss these implementations in detail in a future article.

The final work in the programme was Melvyn Poore's *5 Movements*. This work is different as, in addition to instrument treatments, it employs a collection of MIDI sequences in 'explode' objects which are triggered by a keyboard and, in turn, drive an FM synthesiser. Ultimately, the synthesiser might be implemented within the ISPW. This patch is also interesting as it includes an 'external object', written in C. This option allows extensions to MAX to be created where an implementation would otherwise be difficult or impractical.

The ISPW and the Future

"All that we can say is that the computer makes it possible to create and explore musical structures which before could not exist.(Wishart, 1991)

The ISPW will certainly not be the last word in performance systems, nor does it address all the possibilities of live electronics, but I believe that it does represent a significant advance. In the article quoted above, Trevor Wishart talks of the transfer of computer power from the mainframe to the desktop which has revolutionised the way computers are used in music making (as in most other areas). Between the migration of mainframe synthesis and processing (eg through the Composers' Desktop Project) and the more accessible, but far more restricted, real-time capabilities of MIDI equipment there had seemed to be a great gulf fixed. Because much of my music making occurs precisely in this gulf, I spent a great deal of time and energy during the 1980s looking for ways to fill the void. The ISPW may not have achieved that, but it has created a significant bridgehead across it.

Undoubtedly, the perception, if not the actuality, of the ISPW's future received a setback with the demise of NeXT's hardware division early in 1993. IRCAM have, however, made a commitment to developing the ISPW on at least two new platforms, NeXTstep for Intel Processors and Silicon Graphics Indy, and to maintaining compatibility between versions of IRCAM MAX. A recent announcement states that work on these ports is already under way. Meanwhile, the ISPW remains the only system capable of even beginning to address the needs of a composer/performer such as myself and bringing the computer revolution of which Wishart speaks off the desktop and into the concert hall.

There are many possibilities of the ISPW that I have not been able to discuss here, for example score following and highly flexible sampling capabilities, to name but two. Iam also beginning to use MAX as an experimental system for researching other areas of live performance, in particular the inadequacy of current approaches to mixer

design. While some of my work is already pushing the limits of the ISPW's processing power (my work has always been like that), it is undoubtedly helping me to “create and explore musical structures which before could not exist”.

Repertoire

NOTE - The following list includes only those works performed by the author; many other ISPW implementations have been made at IRCAM and elsewhere. implementations are by the author unless otherwise stated.

Benjamin, G,	*Antara* (implementation Cort Lippe, IRCAM)
Casserley, L.	*Barks to Carter and Dee Mynah*
	bc (an environment for improvisation)
	The Monk's Prayer from *The Unending Rose*
	UbAtAbU
Casserley, L. and Desorgher, S.	*Sirwnai - Odyssey of Light*
Desorgher, S.	*The Incredible Clanking of the Chains and Cogs of Beelzebub*
	The Lone Ranger Rides Again
	Music of the Spheres
Lambert, J.	*Accents*
Montague, S.	*The Eyes ofAmbush*
Poore, M.	*123*
	5 Movements (implementation MP, ZKM, Karlsruhe and RCM)
	The Long and the Short
Roxburgh, E.	*At the Still Point of the Turning World....*
Stockhausen, K.	*Solo* (implementation Melvyn Poore, ZKM, Karlsruhe)

References

Casserley, L. (1980): Series Phi (Proceedings of the EMAS Computer Music Conference ‘Computer Music in Britain’)

Casserley, L. (1985): Software for an Economical Real-Time Digital Audio Signal Processor (Proceedings of the Institute of Acoustics, Vol 7 Part3.)

Casserley, L. (1986): Digital Signal Processing in Electroacoustic Music (Proceedings of the Institute of Acoustics, Vol 8 Part6.)

Emmerson, S. (1991): Live Electronic Music in Britain: three case studies (Contemporary Music Review, Vol 6 Part1.)

Lindemann, E. et al. (1991): The Architecture of the IRCAM Music Workstation Computer Music Journal 15(3).

Lippe, C. (1991): Real-time Computer music at IRCAM (Contemporary Music Review, Vol 6 Part l.)

Lippe, C. (1993): A Musical Application Using the IRCAM Signal Processing Workstation: Real-time control of granular sampling via turbulence algorithms and stochastic processes (The City University Third Science and Music Conference)

Opcode, inc. (1990): MAX (documentation for Macintosh software). (Palo Alto: Opcode, inc.)

Puckette, M. (1991): (1). FTS: A Real-time Monitor for Multiprocessor Music Synthesis (Computer Music Journal 15(3))

Puckette, M. (1991): (2). Combining Event and Signal Processing in the MAX Graphical Programming Environment (Computer Music Journal 15(3))
Puckette, M. (1988): The Patcher (Proceedings of the 1986 International Computer Music Conference. San Francisco: Computer Music Association)
Settel, Z. and Lippe, C. (1993): Real~time Timbral Transfomation: FFT-based Resynthesis (The City University Third Science and Music Conference)
Wishart, T. (1991): Computer Music and Post-Modernism - A New Musical Language? (Sonic Arts Network Journal, Vol 5 (June 1991))

[1] Incidentally, the material for the tape was created at Peter Zinoviev's pioneering computer music studio in Putney, then processed and mixed at the RCM studio.

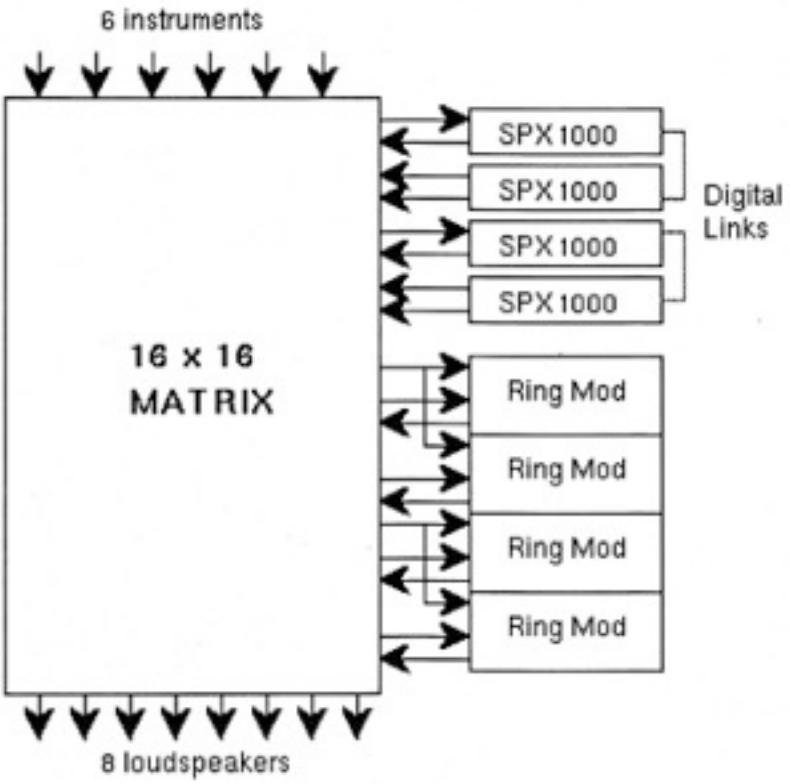

Figure 1 - Treatment matrix for *Ritual Dances*.
MIDI links have been omitted from this diagram

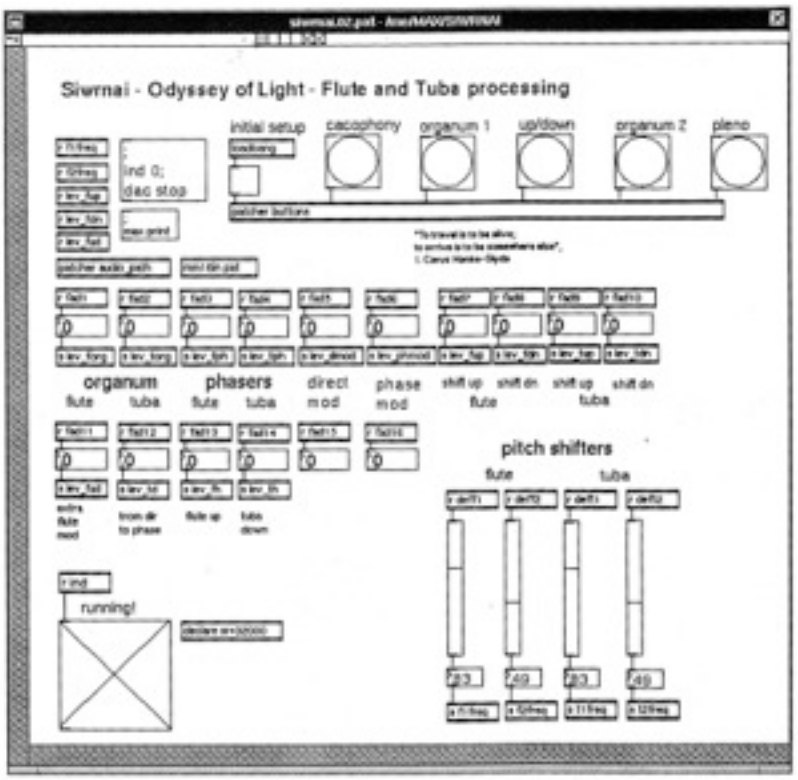

Figure 2 - Control panel for *Siwrnai - Odyssey of Light*

The large squares at the top right are push buttons (using mouse) which trigger new presets using information contained in 'patcher buttons'. The two rows of number readouts show the current state of the faders. The on-screen faders at the lower right allow tweaking of those parameters with the mouse if necessary. The actual audio connections are contained in 'patcher audio_path' (top left).

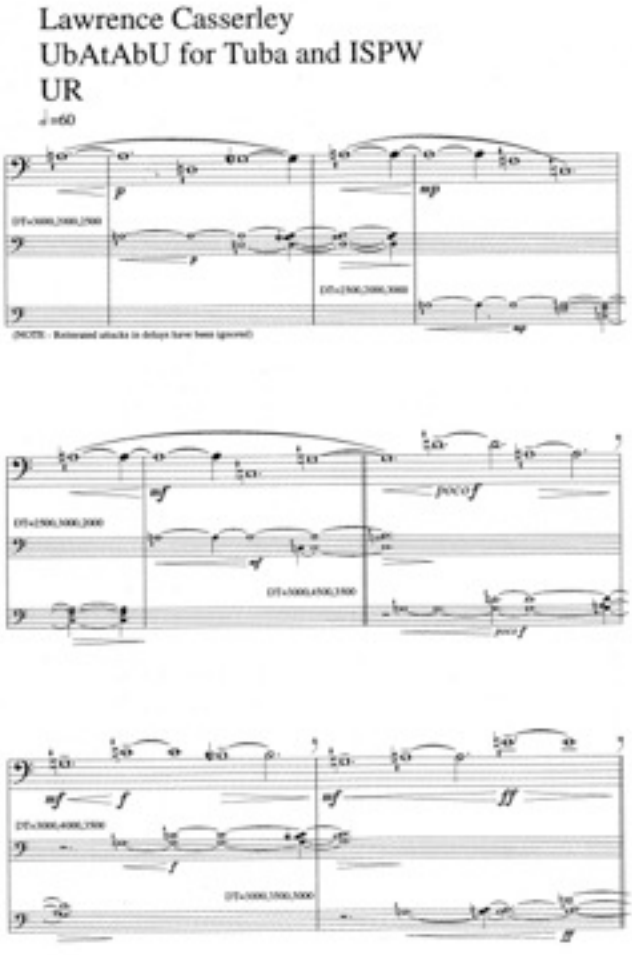

Figure 3 - Page One of *UbAtAbU*

The second stave of each system shows the playback from delay 1, and the third delay 2. The 'DT=x, y, z' notations give the new delay times for each chord.

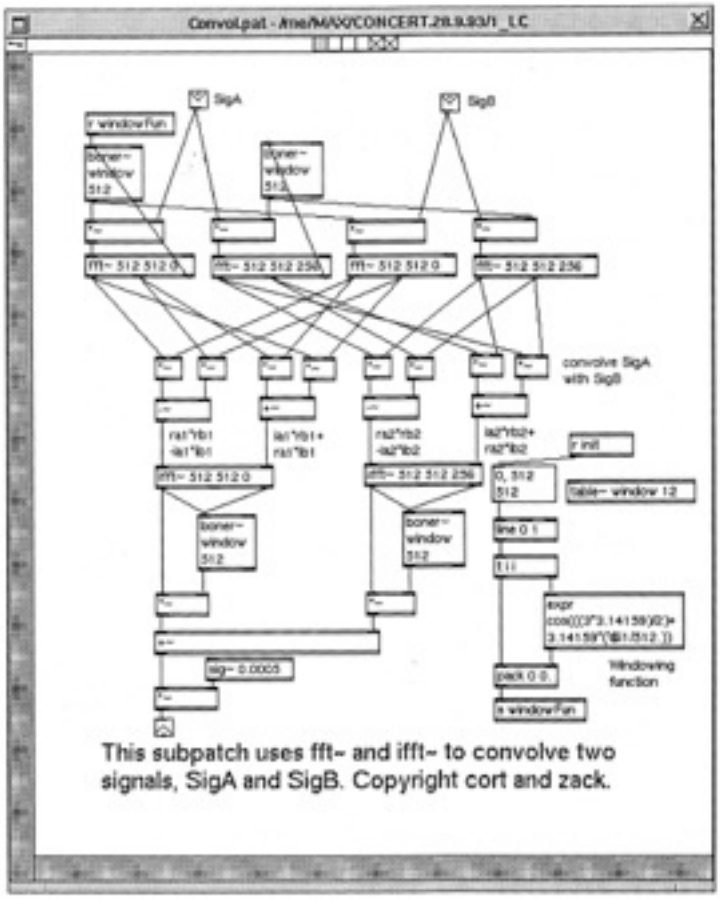

Figure 4 - Convolution subpatch (after Settel and Lippe, 1993)
The two inlets 'SigA' and 'SigB' receive the signals to be convolved, in the case of *LibAtAbU* these are respectively tuba and oscillator bank. After transformation by the 'fft~' objects the two signals are convolved and then resynthesised by the 'ifft~' objects.

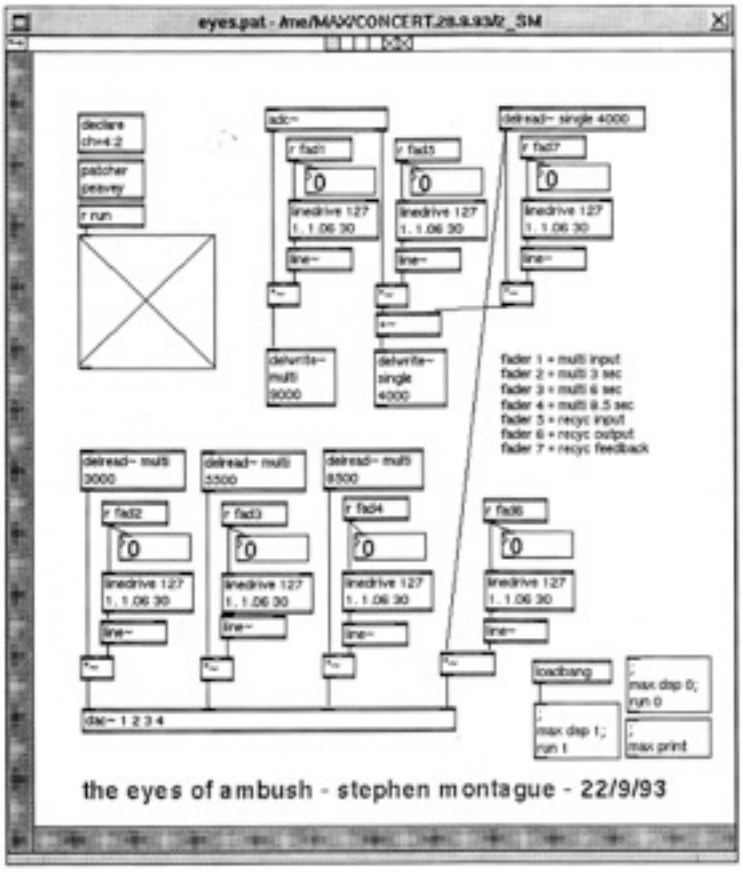

Figure 5 - Patch for *The Eyes of Ambush* by Stephen Montague
The two delays are functionally separate. Input 1 is fed to delay 'multi', which has three taps, at 3, 5.5 and 8.5 seconds fed to separate outputs. Input 2 is fed to delay 'single', a 4 second delay with feedback and separate level controls 'fad6' for output and 'fad7' for feedback'.