BACK TO THE SEA: A SOFTWARE REALIZATION OF DEXTER MORRILL'S *SEA SONGS*

Robert Hamilton
Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music
Stanford University
rob@ccrma.stanford.edu

ABSTRACT

This paper describes the technical and aesthetic challenges faced in the recent software-based recreation of composer Dexter Morrill's work *Sea Songs* for soprano voice, computer-generated tape and Radio-Baton controlled hardware effects-processor. Through careful analysis of the composer's own notes as well as through extensive testing of Morrill's original Digitech TSR-24 stereo-effects processor, a flexible and extensible software emulation of the piece was created as a Max/MSP application.

1. BACKGROUND

In 1995, Dexter Morrill composed *Sea Songs*, a four-part work for soprano, computer-generated tape and real-time effects processing. Premiered by soprano Maureen Chowning in October of 1995 at Colgate College, the four songs respectively titled *I. e lo soleills plovi*, *II. The Sea Becomes Me*, *III. The Slow Pacific Swell* and *IV. Or Followed the Water*, set text based on poems by Ezra Pound, Agha Shahid Ali and Yvor Winters[1] over a pre-recorded accompaniment generated using an E-mu Morpheus synthesizer and Paul Lansky's Cmix. Morrill's use of Max Mathews' Radio-Baton as the real-time controller for the Digitech TSR-24 hardware stereo effects-processor broke from earlier uses of the Baton as a synthesizer controller and allowed the soprano to incorporate fluid gestural control of real-time processing into her live performance.

While *Sea Songs* was performed numerous times throughout the 1990s and subsequently recorded on Morrill's Centaur Records release entitled *Music for Stanford* [5], the piece's exposure to a larger community of performers was likely limited by its reliance on the availability of a Digitech TSR-24 effects processor. As time has passed and the TSR-24 became first more difficult then nearly impossible to find, the piece has faced an uncertain future due primarily to its reliance on this specific hardware device.

The project to recreate *Sea Songs* began in the summer of 2006 with a plan to feature a software-based version of the work at CCRMA in the Spring of 2007 as part of a series of festivities honoring Max Mathews' 80th birthday. With a new wireless version of the Radio-Baton already in use at CCRMA, it was suggested by Max Mathews that one work which should be included in such a celebration was *Sea Songs*, as Morrill's composition represented "a unique and powerful way of being expressive with the Radio-Baton" [4]. The goal

Figure 1: Maureen Chowning performing *Sea Songs* at the 2007 MaxFest concert.

was to create a modernized and extensible version of the piece in a popular and well-supported interactive computer-music language running on a laptop computer, thus making the piece accessible to a new generation of performers and audiences alike. With the support of Mathews, soprano Maureen Chowning and Dexter Morrill himself, system design and development on the new *Sea Songs* began in earnest in the Fall of 2006 and culminated with Chowning's performance of the newly recreated work on April 29, 2007 in concert at the Computer History Museum as part of CCRMA's MaxFest celebration.

2. EXISTING REPERTORY INITIATIVES

While the *Sea Songs* recreation was tasked as an immediate solution to a specific performance need, there exist a number of ongoing initiatives dedicated to the larger goal of archiving and recreating seminal electroacoustic works. Featuring interactive versions of works such as Karlheinz Stockhausen's *Mantra* and Phillipe Manoury's *Jupiter*, the PD Repertory Project [6] headed by Miller Puckette seeks to create open source and public-domain versions of classic electroacoustic compositions using Puckette's Pure Data (PD) software. Similarly, the Integra Project [3] - currently being pursued by a consortium of research centers and performance ensembles - defines as one of its primary goals the "migration of existing repertoire using obsolete technology" to their own new open-sourced software environment. And the Mustica Initiative, developed as part of the more sweeping InterPARES 2 Project [2] brings together a number of international research centers working towards the development of strategies and software tools capable of archiving and recreating the complex informational systems for interactive musical works.

[1] Morrill makes use of text from Pound's *Canto IV* and *Canto XX,* Shahid Ali's *Notes on the Sea's Existence*, and Winters' *The Slow Pacific Swell*.

3. PROJECT OVERVIEW

For the original version of *Sea Songs*, Morrill programmed eight individual patches on the TSR-24 combining multiple instances of digital delays, pitch-shifters, auto-panners, filters and reverb processes into interactive effect-chains with specific parameters linked to the X, Y and Z axes of one baton. Certain patches were enabled for each of the four songs and could be selected in real-time by moving the second baton into specified trigger-regions on the baton antenna. In this manner, the singer herself could control both the level of processing present in the stereo mix and various effect parameters for the currently selected processing patch.

3.1. The Radio-Baton

The Mathews Radio-Baton [1] is a versatile musical controller which tracks two low-frequency radio-wave emitting baton-controllers across a flat antenna array and outputs baton-specific three-dimensional position data over a MIDI connection. The latest incarnation of the baton features wireless batons, five buttons, a four-antenna board and eight 12-bit AD converters. Signals transmitted from each baton to the antenna board are converted into MIDI polyphonic after-touch messages. These messages, decoded in software and used to track baton positions in coordinate space, can be mapped as control signals onto any number of musical systems.

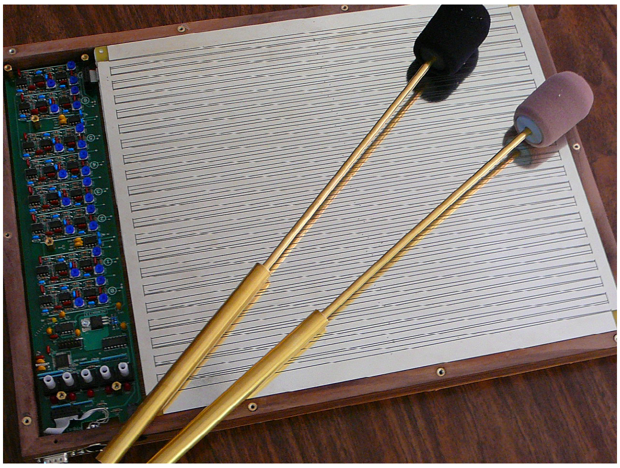

Figure 2: The Mathews Radio-Baton

3.2. The Digitech TSR-24

Introduced in 1993, the Digitech TSR-24 stereo effects processor featured 18-bit A/D and D/A converters, limited sampling capabilities, four-channel output, MIDI control and a customizable algorithm creation and patching system for building complex signal processing workflows. The TSR-24 could be augmented with additional memory and processing cards to increase both the number of effects concurrently available in a single patch as well as the depth and richness of the effects themselves. Real-time MIDI control of individual effect parameters meant the mappings of data from continuous controllers such as the Radio-Baton could be applied in a virtually unlimited number of combinations, making the device ideally suited for real-time dynamically controllable effects processing.

Figure 3: Digitech TSR-24 Stereo Effects Processor

3.3. Max/MSP Realization

In an effort to retain the characteristic sounds and behaviors used in the original *Sea Songs*, a modular approach was taken in building the Max/MSP realization. By re-constructing the original TSR-24 patches as chained DSP modules linked to a Max/MSP matrix~ object, individual effects patches could be tested, modified and replaced without compromising other patches or the over-all stability of the system. While the vast majority of Max/MSP patch-code for data-processing and system control was built specifically for this project, code for each of the DSP modules used in the system was freely-available code supplied by the Max/MSP user community itself.

By chaining modules together as well as routing each module's individual output level through independent gain stages, the relative strengths of each effect could be carefully manipulated. In this manner, each effect could easily be fine-tuned to match the original output from the TSR-24 as closely as possible.

While this project could have been realized in a number of musical software environments, Max/MSP 4.6 was chosen due in part to the software's wide adoption among contemporary electroacoustic composers and performers as well as to the software's large existing user-community. As many of the patches used in the TSR-24 made use of relatively straightforward and well-understood processing methods such as delay, reverb and pitch-shifting, the use of Max/MSP also allowed for the leveraging of existing DSP-code and patches from the community at large. The option of coding the project in PD was strongly considered, as the open-source nature of the software and possible future alignment with the PD Repertory project were compelling arguments for the use of PD. However with a limited development time-frame and a pre-existing signal-routing framework culled from other projects, the decision was made to use Max/MSP with use of the free Max/MSP runtime environment for performance and distribution.

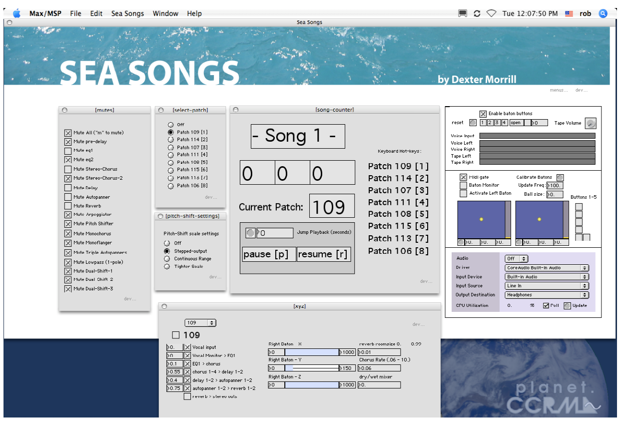

Figure 4: The Max/MSP version of *Sea Songs* features graphic interfaces for controlling and displaying both real-time audio and midi data-streams as well as temporal cues for performance and controls for rehearsal.

4. SYSTEM DESIGN AND DEVELOPMENT

To faithfully recreate not only the subtle timbral and temporal processings employed by Morrill in his extensive programming of the Digitech TSR-24 but also the gestural mappings needed to impart equally subtle control to the performer, it was necessary to address the issue based on the study and analysis of a number of primary sources. Morrill's notes on his methods of composition and hardware programming, his own personal recollections communicated via phone and email as well as in person, and perhaps most importantly, access to Morrill's own Digitech TSR-24 provided the keys to modelling this complex system. From the performer's side, Maureen Chowning's personal scores, complete with annotated baton positions and clear depictions of the desired control gestures proved invaluable for the successful recreation of the work. Additionally, Max Mathews' own insights towards the inner workings of the Radio-Baton, as well as his notes, writings and remembrances about the technical and aesthetic concerns pertaining to *Sea Songs* proved extraordinarily helpful.

4.1. Baton Input

To be analyzed and mapped in software, Radio-Baton MIDI output must be converted from its form as polyphonic after-touch messages into a more useful set of X, Y and Z coordinates. While initial testing and development of the system has made use of a Max/MSP-based conversion patch to decode the polyphonic after-touch messages – kindly made available by Ingrid Linn – work has begun on a simple Max/MSP external coded in C fulfilling the same function. Additionally, in anticipation of the planned addition of USB output to the Radio-Baton hardware, work is underway on the development of an Open Sound Control address-space spec for baton coordinate and button output.

4.2. DSP modules

As the DSP processing utilized in the piece originally was designed as a chain of processing modules with flexible input and output paths, a similar paradigm was employed in the Max/MSP realization. For reverb processing, Olaf Matthes' freeverb~ implementation of the Schroder-Moorer reverb model was used, both for baton-controlled reverb effects as well as for an independent global reverb for room sweetening. All other effects modules were built from basic Max/MSP objects.

With careful use of DSP-chain muting on dormant processing modules, it was possible to run full development recreations of Morrill's most intensive patches - some making use of up to six simultaneous pitch-shifting modules, as well as a number of chorus, feedback delays and reverbs – on an Apple 1.25 GHz PPC Powerbook running OS X. For live performance, a newer Intel-based MacBook Pro was used, providing sufficient processor speed to allow much lower Max/MSP I/O Vector and Signal Vector sizes and correspondingly lower latencies.

4.3. Data Scaling Controls

One aspect of the original processing workflow which was clearly in need of updating was the manner in which ranges of control data were scaled and mapped to individual parameters of DSP modules. Rather than set one hard-coded mapping, it was most useful to instead create simple range-interfaces for each axis of movement on the right baton. In this manner, a performer could customize the size and range of their physical gestures across the baton antenna's field of motion.

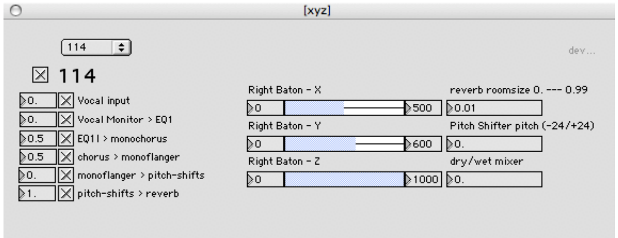

Figure 5: Input data scaling controls allow users to customize the range mappings for each effect along each axis of baton motion (right sliders) as well as each effect's individual gain levels (left number boxes).

For instance, if the Y-axis of the baton is mapped to a possible 4-octave pitch-shifting variation, given a full 1-1 mapping of physical Y range to pitch, it becomes very difficult to perform gradual or precise variations of pitch; a small motion would simply create too great a pitch-varying response. By making use of a graphical range-bar as seen in Figure 5, the performer can restrict the range of mapped output to any subset of the full four-octave range, allowing for flexible customized use of the full baton range of motion.

4.4. Performance Modes and Display

From song to song, Morrill's compositional style varied in that clearly defined program changes are notated at specific instances in three songs (songs 1, 3 and 4), while Morrill allows the performer a more improvisatory choice of processing as well as sung rhythm in song 2. To allow for varying interpretations of each of the four songs, and to aid the performer in rehearsal situations, dual modes of performance were created which would allow the selection of individual software patches to be made either interactively, using the left baton, or automatically by timed triggers synched to the pre-recorded tape track.

In the first mode, as the left baton is moved into different quadrants of the baton antenna, a control message is sent to load pre-set configurations of DSP modules. To aid the performer in the selection of patches, a simple visual feedback display was used to highlight the selected portion of the baton on a computer-monitor (see Figure 6). Whereas the original performance practice gave no feedback at all to the performer when selecting a new patch, this method allows the performer the reassurance of clearly viewing which patch was selected as well as the other possible patch options at any given time in the piece.

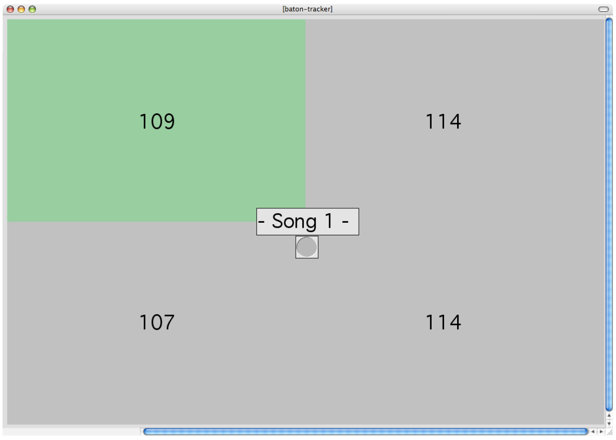

Figure 6: Visual Patch-selection display

Musically, the introduction of multiple performance methods for individual songs in the greater work appears to work extremely well. In rhythmically strong songs such as songs 1 and 3, where patch changes occur clearly aligned with audibly defined events, the ability to synchronize automatic patch changes with the tape accompaniment allows a relatively inexperienced performer to sing with only one baton, making the learning-process of suitable interactive gestures significantly easier. Subsequently, as the performer becomes more comfortable with the baton, the traditional two-baton performance mode can be used. In the case of song 2, where the tape-accompaniment is more ambient and without clear temporal or timbral cues, and where the intent of the composer is to allow greater performer improvisation with the selected effects, it is most appropriate to use the more flexible two-baton performance mode.

5. DISCUSSION

The recreation of classic interactive electroacoustic works in the musical repertory is an important task which should be increasingly addressed as countless important works in the field face an unplanned obsolescence due to their reliance on aging or discontinued hardware and software systems. In the case of *Sea Songs*, the fact that the composer, principal performer and the inventor of the hardware controller were easily accessible during the course of the recreation made what could have been an extremely difficult task relatively straight-forward and without incident.

One substantial benefit of having contact with the work's composer and original performer was the ability to draw clear distinctions between original programming choices made for creative gain and those made as a compromise in dealing with the technology of the time. While the original intention of the project was to model the processing and control systems of the TSR-24 as closely as possible, discussions with the composer opened up the possibility of using advances in DSP and software to improve upon some of the original effects used in *Sea Songs*. For example, the sound of the TSR-24's "gigaverb" reverb effect used for a number of patches was not in and of itself a necessary part of the work; indeed the composer was happy to utilize any more realistic and flexible reverb algorithm available. Similarly, with the specific details of pitch-shifting available on the TSR-24, Morrill's goal was less the use of a specific pitch-shifting algorithm and more the cleanest and most natural manner in which to shift pitch in real-time. In this manner it is clear that not all compositional decisions made using less flexible musical control and generation systems were not made based on their aesthetic merits, and that perhaps such processes can be modified and improved without altering the fundamental experience of the piece itself.

6. ACKNOWLEDGEMENTS

The author would like to thank Max Mathews, Dexter Morrill and Maureen Chowning for their invaluable assistance and insights which made this project possible.

7. REFERENCES

[1] Boulanger, R., Mathews, M. "The 1997 Mathews Radio-Baton & Improvisation Modes", *Proceedings of the International Computer Music Conference*, Thessaloniki, Greece, 1997.

[2] Duranti, L. "Preserving Authentic Electronic Art Over the Long-term: the InterPARES 2 Project", In Proceedings of the AIC Annual Meeting, EMG Session, June 2004.

[3] Integra Project, http://integralive.org/.

[4] Mathews, M., Personal e-mail communication, May, 2007.

[5] Morrill, D. *Music for Stanford*, Centaur Records, CRC2732, 2004.

[6] Puckette, M. 2001. New Public-Domain Realizations of Standard Pieces for Instruments and Live Electronics. *Proceedings of the International Computer Music Conference*, San Francisco, 2001. pp. 377-380.