Journal of the Japanese Society for Sonic Arts, Vol.6 No.1 pp.6–11

Research Report

EXPANDING THE PERFORMANCE POSSIBILITIES OF REAL-TIME COMPUTER MUSIC REPERTOIRE THROUGH RE-WORKING DATED TECHNOLOGY

Richard Dudas
Center for Research in Electro-Acoustic Music and Audio Technology (CREAMA)
Hanyang University School of Music
Seoul, South Korea

ABSTRACT

While collaborating with instrumentalists on concerts of music with an interactive real-time computer component, not as a composer but as a technological co-performer, it has been regularly necessary for me to update the technology used. This need stems partially from the rapid obsolescence of both hardware and software technology: compositions that were created only a decade ago often need to be re-worked in order to be able to present them to new audiences. Even where more recent works are concerned, I have often found that many real-time compositions (even high-profile ones) frequently use hastily designed and poorly maintained Max/MSP-based software requiring some degree of renovation before they can be used reliably in rehearsal, not to mention in concert. Many of the published writings about this re-design of obsolescent technology have tended to focus on a slavish faithfulness to the original technology, regardless of any shortcomings that might be inherent in it. However, in many cases both performers and myself agree that it is necessary to redesign, rethink, and improve pieces, in the same way that a pianist may need to rethink and re-interpret music originally written for harpsichord, fortepiano, or other pre-modern pianos. Just as instrumental performers desire to interpret music to the best of their abilities in order to make that music come alive for their audience, so, too, should the technical performer endeavor to make pieces sound the best they can, both from a technical and musical standpoint, in order to entice today's increasingly technology-savvy audiences. This paper will briefly discuss several compositions as case studies, and look at some of the most useful real-time signal processing techniques in Max/MSP which have been used to improve them, so others may benefit when working on technology-updating projects, or new computer music endeavors.

1. INTRODUCTION

When an unedited print of the original 1927 cut of Fritz Lang's *Metropolis* was discovered in Buenos Aires in 2008, the film's copyright holder, Kino International, in cooperation with the F.W. Murnau Foundation – which supervised the film's restoration in 2001 from a combination of four different archival prints of varying quality – immediately set about restoring the newly found (and heavily damaged) footage and incorporating it into the existing restoration. It would have been unthinkable to present the newly discovered footage (let alone the entire film) in a severely degraded state. Even after some spectacular restoration work, the result is a film which changes noticeably in quality throughout its nearly two and a half hour length, but which presents the director's original intentions in their best possible light. As audience members we all marvel at this kind of restoration of old cinema – where digital technology is used to allow the source material to look its absolute best, and bring new life into an old film.

Part of my work as a computer musician (and also part of my work within the concert activities of CREAMA) has been to promote performences of electroacoustic repertoire, and this implies bring new life into performances of old (and sometimes not-so-old) electroacoustic and computer music, which also needs to sound its absolute best, regardless of whether it is a "classic" piece from the 1950's or a world première using the latest computing tools. Today, audiences are used to being dazzled with high tech computer-controlled sound and lighting in a variety of arenas: rock concerts, film, theater, aquariums, fireworks displays and myriad other public places and events. Similarly, performers who work with electronics are generally interested in creating a rich, memorable, captivating "show" (those with whom I have worked certainly are), and therefore the quality of the electronics in a modern computer music concert needs to fit to some degree into today's audiences' expectations. The viewpoint taken in this paper is in my role as a ***performer*** of electronic and computer music (not as a composer or software programmer). When I am "on stage" (which, for me, in reality is actually not on stage but somewhere in the hall), I want to make sure that my part of an electroacoustic music concert – usually behind the computer with a finger poised over the spacebar[1] – allows compositions, composers and performers to sound the best they can.

2. DATED TECHNOLOGY

This desire to put on a good "show" has often meant having an unsolicited side job as "data archaeologist." Some of those involved in computer music performance have

[1] Actually, I always recommend ***not*** using the spacebar for triggering musical events, but rather another, smaller, key which makes less noise, such as one of the arrow keys or the *enter* key on an extended keyboard.

written at length about the need to maintain and update software as well as locate and preserve obsolete hardware [1, 2, 3, 4]. Some of these same people, alongside others, have been working on designing comprehensive software systems and repositories to consolidate repertoire into one technology. These include, but are not limited to Miller Puckette's Pd Repertory Project [5], Jamie Bullock's Integra Live [6], and David Brooke Wetzel's Interactive Event Manager (IEM) [7]. Most of the live electronic music which I have performed has made use of the Max/MSP environment for their electronic component, so the examples shown here will use that environment without any particular third party framework.

I do not want to delve too deeply into the aesthetics and ethics of reworking old technology – that would take up an entire book – but I do need to touch upon the subject a little bit because there are a lot of different schools of thought on this subject. Although I am presenting the reworking of dated technology via my personal viewpoint, it is just one of many perspectives. Some believe in thoroughly analyzing the often obsolete technology used in the original version of a piece in order to be able to recreate it as closely as possible using modern technology. I personally believe this to be an important step, but would go one step farther: I believe that in many cases the obsolete hardware and software technology can be updated and *improved* without altering the composer's intention. Of course, in cases where imperfections in a given technology are used by composers for musical purposes, this use of the technology should be taken into consideration in any updated version.

To cite one example of this creative use of faulty technology, the inaccurate pitch detection of the early IVL Pitch to MIDI Converter from 1985 was used by composer/saxophonist Bruno Spoerri as a variation-generating feature in his improvisatory piece "Controlled Risk" [8]. Using a more precise modern pitch tracker in this scenario would undermine the musical intent of the original piece. However, on the flip side of the record (so to speak) in a scenario where one is updating a piece using the early Yamaha SPX-90 for simple pitch-shifting to create a "duo for solo instrument" effect, replacing that technology with a cleaner, more up-to-date transposition effect would probably be preferable. (Would anyone really want to add background hiss to a modern performance of Stockhausen's "Solo" that makes use of computer-based digital delay lines instead of the original analogue tape delay??!)

Of course there will always be those fine-line cases where one would have to carefully consider whether the juddery-sounding transposition effect of the SPX-90 might have been used for compositional purposes, or whether a cleaner transposition would be more effective. Certainly, modern ensembles and orchestras, using instruments whose design was greatly improved throughout the 19th and 20th centuries, are able to perform older works with a fresh interpretation and modern "sound," while remaining accurate to the composer's intentions. As such, there is no reason why modern technology couldn't be used to replace the original technology of dated electroacoustic pieces. There are no set rules, however, and those updating outdated electronic music for performance will certainly be confronted with having to make their own artistic decisions (just as performers do) based on their interpretation of the composer's intentions.

3. ORAL TRADITIONS

Interestingly, there is also an oral tradition in computer music. On the technical front, this often stems from a general lack of documentation where the operation of technology is concerned. The published scores for many pieces either do not indicate clearly indicate how to operate the hardware or software used (although some make a valiant attempt), or often discuss an earlier, outdated version of the technology. The consequence of this is that performance instructions for many pieces are verbally conveyed from generation to generation of computer musician. One example of this was my preparation for a performance of Jonathan Harvey's *4th String Quartet*, where I was fortunately able to spend several hours in a teleconference with someone from IRCAM who had performed the piece and was able to go through both the details of the Max/MSP patch and through the score – *bar by bar* – to teach me how to perform it. On the performance front, a lack of a clearly defined universal performance practice in electroacoustic music, where the operation of technology is concerned, certainly contributes to the continued need for this oral tradition.

4. CASE STUDIES

The following case studies are used to show some practical and useful examples of techiques used for the redesign or re-implementation of dated technology, as implemented in Max/MSP. Some of the techniques shown are simple, others more complex. I have tried to make a selection of real-world examples that demonstrate commonly-used effects, instead of things that may be specific to the technology for the particular piece being discussed. These are not just common improvements that can be made to older patches, but can be taken into consideration when desigining new projects as well.

4.1. Musgrave's *Narcissus*

Thea Musgrave's *Narcissus* (1987), which exists in the composer's own versions for both for flute and clarinet with digital delay, has been a widely performed work in both original versions as well as a few arrangements for other instruments by individual performers. The piece, whose electronic component was originally designed for the rare and quickly obsolete Vesta Koza DIG 411 digital delay rack unit, has been the subject of many software re-creations, using PureData, Max/MSP and other software [9, 10, 11]. The delay effect is not particularly complex from a technical standpoint, though it is musically effective for the piece. In spite of the fact that there were initial concerns that the piece would not outlive the hardware used [12], the popularity of the piece combined with the fairly straightforward technical setup has kept it in the repertoire. It is therefore not necessary to describe the original hardware and the task of porting old technology to new. The Max/MSP implementation by David Brooke Wetzel [11] is an accurate re-implementation of the features (and shortcomings) of the Vesta Koza DIG 411 delay unit, has often served as the groundwork for other software re-implementations, including the concert patch described here. Our updated concert Max/MSP patch, however, includes some important modifications, improvements

and techniques which merit discussion.

Some of the technical imperfections inherent in the original delay unit – largely due to its being amongst the first generation of digital hardware – which are also reflected in Wetzel's original patch were: audible clicks when the delay time is changed, or when using the delay's "hold" feature, slight zipper-noise when rapidly changing modulation, volume and feedback parameters, and a limited delay length. Certainly some of these limitations in the original unit informed compositional decisions, such as keeping a fixed delay length for large spans of the piece, as well as ensuring that large parameter changes occur during long rests. Among these deficiencies in the original delay unit and subsequent Max patch, the simplest update here was adding smooth gain control to all parts of the signal chain using the *line*∼ object. This is a basic update I recommend doing to all older patches, many of which may have originally avoided using an extra signal object for concerns of exceeding the processing power of older computers. This update can be done by simply adding a message box describing a short ramp connected to a *line*∼ object in-between the number box and the signal multiply, as shown in figure 1.

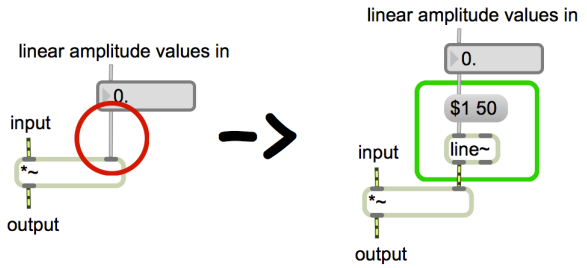


Figure 1. Smooth gain change.

Similarly, delay lines in older patches are often implemented in the simplest way possible (using a single pair of MSP *tapin*∼ and *tapout*∼ obejcts). Changes to the delay time with this simple setup often causes audible and disconcerting discontinuity in the output. Updating and improving this in order to eliminate unwanted clicks is a little more complex than simply controlling the delay time with a *line*∼ object, which would cause a quick glissando. The trick is to use a pair of delay lines in parallel and crossfade between them[2]. Only one delay line is used at any given moment, except for the brief instant when the output gains of the two delay lines are changed from off to on, and on to off, respectively. This technique, sometimes referred to as a "*smooth tapout*∼", is shown in figure 2. Note that the incoming delay time changes are speed limited to an amount of time slightly greater than the crossfade time.

Finally , the output gain of the entire delay effect has also been updated to take perceptual factors into consideration. When instrumentalists perform a crescendo or diminuendo, their timbre changes alongside the change in volume. A simple gain change does not sound very convincing in the context of this piece, so it was decided

[2] Note that this kind of crossfade is already built-in to the standard MSP *delay*∼ object. You can activate it by sending the *ramp* message followed by a crossfade time in milliseconds. Unfortunately this feature has not yet been integrated into *tapout*∼, requiring us, for the moment, to use some variation of the patch shown in figure 2.

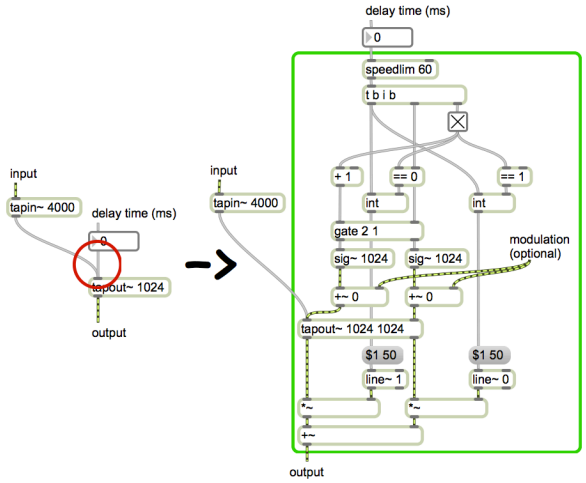


Figure 2. Smooth delay change using crossfade.

to implement a more complex volume control so the entrances and exits of the delayed instrumental sound appear to match the instrumental sound more closely – a modification of the original which amounts to a performance decision made for purely musical reasons. Technically speaking, it was achieved by applying a high shelf filter to the attenuated output signal. The amplitude of the filter's shelf is controlled by the same linear amplitude value which controls the output gain, as shown in figure 3. Note that this additionally makes a more natural volume control than simply using linear amplitude – in terms of perceived volume, it is similar to squaring the linear amplitude value, a technique often used when converting MIDI velocity values to amplitude [13].

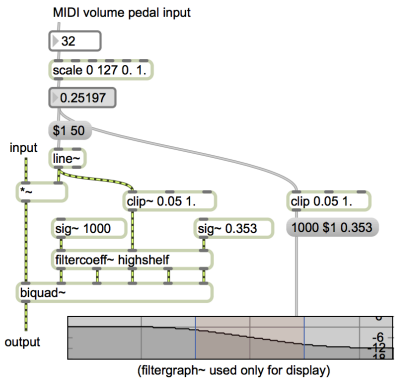


Figure 3. High shelf filter for musical volume change.

4.2. Harvey's *Advaya*

The original version of Jonathan Harvey's *Advaya* (1994) for cello, keyboard and electronics, made use of a Akai S-1000 sampler (played by the keyboardist), a Yamaha SPX-100/1000 rack-mounted effect box (only for its harmonizer effect) and two CD players for soundfile playback. For the piece's premiere at IRCAM, the soundfile playback was implemented on the Ircam Signal Processing Workstation (ISPW), even though it had been designed with commercial CD players in mind for ease of perfor-

mance outside IRCAM. By 2006, Frédéric Voisin and Denis Lorrain at IRCAM had rebuilt a Max/MSP Patch to replace the SPX and CD players. Although the Akai S-series samplers were popular and easily available throughout the 1990s, they are now becoming somewhat more difficult to obtain, and more worryingly, it is increasingly more difficult to find a functioning SCSI CD drive to connect to the Akai in order to load the samples themselves. Therefore, when our center, CREAMA, had programmed this piece for the second time in 2010, I decided it would be a good opportunity to implement the Sampler in the MSP patch, since this piece made use of only the most basic sample playback features of the Akai sampler.

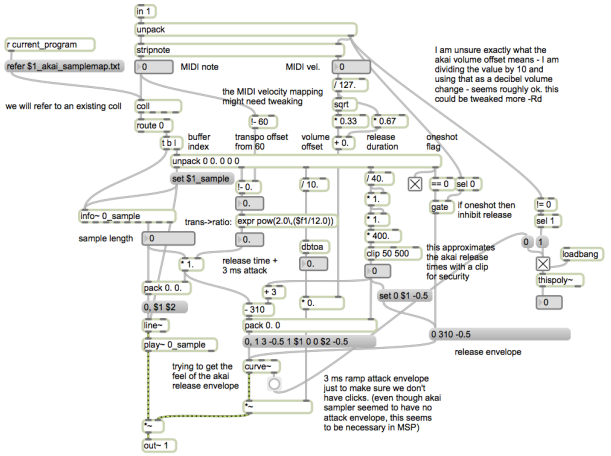


Figure 4. Imitation Akai Sampler Voice for use in *poly*∼.

The polyphonic patch to be used with the MSP *poly*∼ object, shown in figure 4, implements a basic sampler modeled on the Akai S-series samplers as used in Harvey's *Advaya*. It is not meant to be a comprehensive imitation of the Akai, but at least it might serve as a basis for emulating a similar bare-bones sampler in other pieces. The *coll* object contains a list of values taken directly from parameters in the Akai sample banks for *Advaya*:

- MIDI Note Number (*coll* index)
- MSP Buffer Number
- Number of Semitones to C3
- Akai Volume Offset
- Akai Release Time
- One-Shot Flag (play entire sample, ignore release)
- Sample Name (only for reference – not used)

Keeping the original values for these parameters as they appeared in the Akai (instead of putting them into a more Max-specific format) made it easier to cross-check them against the values displayed in the sampler in order to avoid errors while designing the patch. Furthermore, keeping them in their "original" format allows the MSP sampler to be re-designed in the future to more closely imitate the Akai, should it be necessary. This does mean that the parameters need to be manipulated slightly in Max before being used – one example of this would be the seemingly complex conversion of the *Release Duration* parameter into millisecond values that reflect the actual release time used by the Akai (just to the right of the center in the figure).

In addition to being used to create a polyphonic sampler or synth, the *poly*∼ object can also be useful for patches with multiple *sfplay*∼ objects, that are used to play longer sound files. Very often Max patches (such as the patch for *Advaya*) contain triggers for lengthy sound files, and putting their playback inside a polyphonic patch can help not only to streamline the patch but also to serve to smoothly overlap or fade-out sound files should the performer play more quickly through the series of soundfile cues than expected. A modified version of the polyphonic soundfile playback patch used for *Advaya* is shown in figure 5.

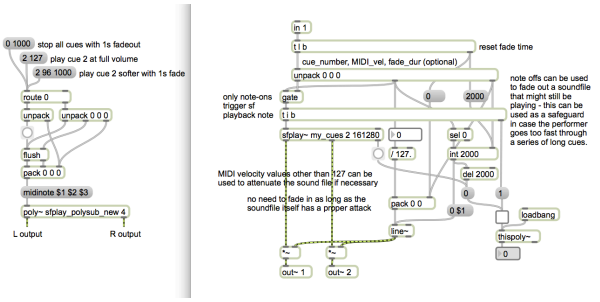


Figure 5. Using *sfplay*∼ inside *poly*∼ .

Naturally, this is just one possible implementation of such a patch; a simpler – or more complex – version could be used, depending on the context of the patch where it is needed. Notice that the note number sent to the *poly*∼ is actually a cue number defined in an *sflist*∼ object. Notice also the the *sfplay*∼ object has been given a very large buffer size (161280). This is a safety measure to make sure the object does not stop playback if the computer's processor is overloaded (larger buffers mean that the processor has to read and convert chunks of sample values from the disk to memory less frequently). The *sflist*∼ object should also be given the same buffer length argument![3]

4.3. Reich's *New York Counterpoint*

As electroacoustic musicians, very often we work with a multi-channel speaker setup in concert. In an era where many people have multi-channel home theaters, today's computer music concerts seem much richer if they go beyond having just a couple of speakers on stage. Steve Reich's *New York Counterpoint* (1985) is not always considered as an electroacoustic piece by the computer music community, however it does make use of recording technology at its core (after all, Reich was one of the early members of the San Francisco Tape Music Center). Most performances of this piece use a stereo tape recorded by the performer (or even the pre-recorded tape part available through the publisher). With 11 contrapuntal lines of music (10 and the live soloist), it is only natural that this piece has on occasion been performed by diffusing the 10 soundfiles over a multi-speaker setup [14]. Working together with clarinetist Pete Furniss and composer Alex Harker in the UK, we prepared an updated version for the Minimal festival in Glasgow (screenshot shown in figure 6). This was based on an earlier multichannel version from 2006. To spatialize the pre-recorded clarinet tracks, I used Ville Pulkki's *vbap* object suite for Max/MSP, a

[3] If *sflist*∼ and *sfplay*∼ have different buffer sizes, it could cause Max/MSP to crash!

public-domain collection of external objects which implements vector based amplitude panning [15]. This is an excellent, easy, and efficient system to use for complex multichannel panning, and has been implemented within multiple computer music software platforms, including CSound, Max/MSP and Pd. Note the interface we made using the standard Max *lcd* object – this was useful to help us work out the location of the 10 sound sources in the 8-channel circular speaker array while creating and editing the panning cues.

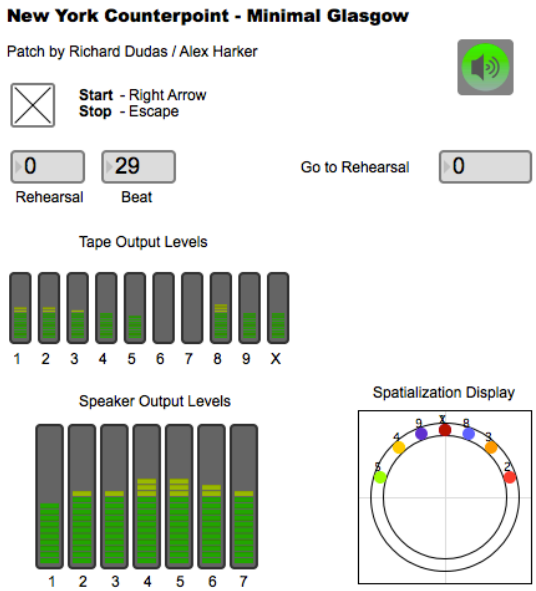


Figure 6. Multichannel panning for *New York Counterpoint*.

4.4. Saariaho's *Noa Noa*

Kaija Saariaho's *Noa Noa* (1992) is another example of a piece which is often played and which exists for multiple technologies (fixed media, ISPW, Max, Pd). I am mentioning it here not in the context of updating software, but rather the possibility of updating its sound files. Many performers I have worked with have expressed an interest in re-recording instrumental sounds used within live electronic pieces in order to be able to include their own personal "sound" in the pre-recorded component of the pieces they are performing [16]. *Noa Noa* is an ideal candidate for this, since many of the sounds in this 20 year old piece are beginning to sound somewhat dated. As such, I have transcribed the soundfile cues which are composed of unprocessed flute sounds, so performers I work with in the future will be able to personalize their performance. The first page of cues is shown in figure 7.

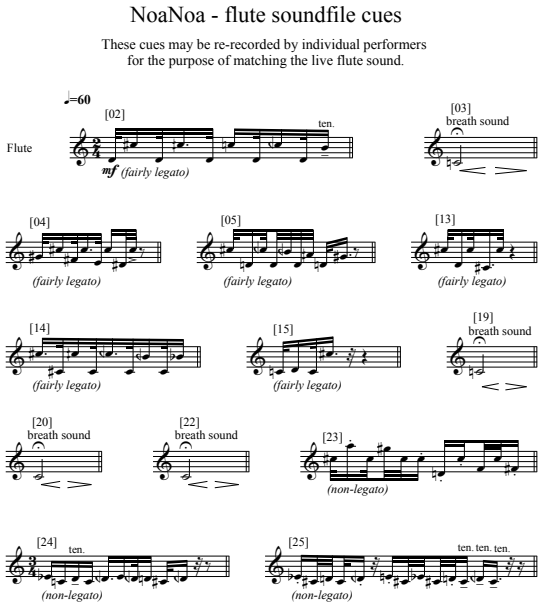


Figure 7. Transcribed cues for *Noa Noa*.

5. CONCLUSION

In updating the technology for the many pieces I have coperformed with instrumentalists, one important circumstance has come to light: much of the electroacoustic and live electronic repertoire – even that which my be well-known from recordings – is seldom performed live outside its area of origin (usually Europe or the US). Keeping this repertoire alive by creating updated, vivid performing versions of it allows these pieces to be performed more widely (at least until that technology becomes outdated), especially throughout Asia and Australia – areas which have already developed strong electronic music traditions of their own. This wider dissemination has unfortunately been hampered to some extent by the difficulty in obtaining certain pieces of older electronic music hardware: a key first step in making an initial analysis of the technology, before designing an updated software replacement. The techniques shown here will hopefully aid others not only in updating older technology, but also in designing new technology for live musical works. Even in the face of technological obsolescence, interactive electroacoustic and computer music continues to develop forward alongside the rapid development in technology itself.

6. REFERENCES

[1] Berweck, S., *It Worked Yesterday: On (Re-) Performing Electroacoustic Music*, Doctoral Thesis, University of Huddersfield, UK, 2012.

[2] Bullock, J., Coccoli, L., "Modernising Musical Works Involving Yamaha DX-based synthesis: a case study " , *Organised Sound*, Vol. 11, Issue 3, Cambridge University Press, Cambridge, England, UK, pp221-227, 2006.

[3] Polfreman, R., Sheppard, D., Dearden, I.,"Time to Re-Wire? Problems and strategies for the mainte-

nance of live electronics", *Organised Sound*, Vol. 11, Issue 3, Cambridge University Press, Cambridge, England, UK, pp229-242, 2006.

[4] Wetzel, D. B., "A Model for the Conservation of Interactive Electroacoustic Repertoire: Analysis, Reconstruction, and Performance in the Face of Technological Obsolescence", *Organised Sound*, Vol. 11, Issue 3, Cambridge University Press, Cambridge, England, UK, pp273-284, 2006.

[5] Puckette, M., "New Public Domain Realizations of Standard Pieces for Instruments and Live Electronics", *Proceedings of the International Computer Music Conference*, Havana, Cuba, pp377-380, 2001.

[6] Bullock, J., Frisk, H., "Libintegra: a System for Software-Independent Multimedia Module Description and Storage", *Proceedings of the International Computer Music Conference*, Copenhagen, Denmark, pp43-46, 2007.

[7] Wetzel D. B., "The Interactive Event Manager (IEM): A Performer's Approach to Interactive System Design", *Proceedings of the International Computer Music Conference*, Montreal, Quebec, Canada, pp517-520, 2009.

[8] Spoerri, B., "Brouillard – déebrouillé (vom Improvisieren mit der Maschine)", *Audio-Design Symposium*, Elektronisches Studio Basel, Switzerland, 1997. Online article: http://www.computerjazz.ch/Pages/Doc4.html.

[9] Bliton, N., "Thea Musgrave ' s Narcissus", 2011, http://nateblitonmusic.com/?p=162, accessed 2014.7.24.

[10] Rudi, J., Bullock, J.,"The Integra Project", *Proceedings of the Electroacoustic Music Studies Conference*, New York, 2011.

[11] Wetzel, D. B., "Analysis and Reconstruction of Interactive Electroacoustic Works for Obsolete Technology: Thea Musgrave ' s Narcissus", *Proceedings of the International Computer Music Conference*, Miami, FL, USA, 2004.

[12] Dick, R., Untitled Music Review, *Notes*, Second Series, Vol. 46 No. 1, Music Library Association, Middleton, WI, USA, pp236-237, 1989.

[13] Dannenberg, R. B., "The Interpretation of MIDI Velocity", *Proceedings of the International Computer Music Conference*, New Orleans, LA, USA, pp193-196, 2006.

[14] Pras, A., Féron, F-X., Demers, K., "The Pre-Production Process of New York Counterpoint for Clarinet and Tape Written by Steve Reich", *Proceedings of the International Computer Music Conference*, Montreal, Quebec, Canada, 2009.

[15] Pulkki, V., "Compensating displacement of amplitude-panned virtual sources", *Audio Engineering Society 22th Int. Conf. on Virtual, Synthetic and Entertainment Audio*, Espoo, Finland, pp186-195, 2002.

[16] Furniss, P., Dudas, R., "Transcription, Adaptation and Maintenance in Live Electronic Performance with Acoustic Instruments ", *Proceedings of the International Computer Music Conference*, Athens, Greece, 2014. (forthcoming)

7. AUTHOR'S PROFILE

Richard Dudas

Holding degrees in Music Composition from The Peabody Conservatory of Music of the Johns Hopkins University and from The University of California, Berkeley, Richard Dudas also studied at the Franz Liszt Academy of Music in Budapest, Hungary and the National Regional Conservatory of Nice, France. In addition to composing music for acoustic instruments, he has been actively involved with music technology since the late 1980s. As a computer musician, he has taught courses at IRCAM, and developed musical tools for Cycling '74. Since 2007 he has been teaching music composition and computer music at Hanyang University in Seoul, Korea.