

Adapting to change: Working with digital sound using open source software in a teaching and learning environment.

August 16, 2007

Abstract

This paper contributes towards knowledge and understanding of the creative use of software and hardware tools for computer music. It stems from a strategic need to continually rationalize the way money is spent within an academic department on Music IT, and an interest in the advantages of open source software for managing musical and collaborative projects. The authors discussed the most practical way of assessing the use of a completely open source software platform (Linux) and a specific distribution (Ubuntu-studio).

It was considered appropriate that the composer-author (Adrian Moore, AM) try to write something in his style, using his knowledge of software, calling upon the more broadly experienced software developer (Dave Moore, DM) who has an interest in open source software.

After a concerted effort in the studio, it is apparent that the Linux platform has much to offer, but remains limited in a number of areas of Music IT that are used extensively in academia, industry and at home. However, as a platform for teaching and learning computer music it is an ideal tool, though we conclude that some 'introduction' at the outset is required for those who are uninitiated to a Linux based system.

Keywords

Computer music, composition, music technology

Author biography

Dr Adrian Moore is Senior Lecturer in the Department of Music at The University of Sheffield. He is a composer of electroacoustic music and is the director of the University of Sheffield Sound Studios.

Dr. Dave Moore is the studio manager at The University of Sheffield. He is a computer programmer and developer. He completed his PhD in 2003 and currently teaches undergraduate and postgraduate music technology.

Authors' institutional affiliation

The University of Sheffield

Address

Department of Music, The University of Sheffield. Sheffield S10 2TN.

a.j.moore@shef.ac.uk, d.moore@shef.ac.uk

Background

To begin, it is important to set out the background of the researchers and describe the current teaching and learning environment at The University of Sheffield Sound Studios (USSS)¹. Adrian Moore is the director of USSS and is an electroacoustic music composer. Dave Moore is the USSS Manager. His PhD included the development of software tools for sound manipulation².

Traditionally, electroacoustic music involves the creative manipulation of audio using computers and tends to exist on fixed format media (CD, DVD), being played back over multiple loudspeakers. AM's creative work rests upon a tradition founded by the likes of Pierre Schaeffer, Francois Bayle, a French school of musique concrète, and his own teachers, including Simon Emmerson and Jonty Harrison. Broadly speaking, his practice involves the empirical creation and analysis of sound, proliferating materials using well known techniques, and considerable reflection in the selection and placement of materials using a multi-channel audio compositor. Creative processes driving this action-set have been well documented (Eaglestone et al, 2007). This current experiment investigates how compositional paradigms migrate between computer operating systems. There will be a number of obstacles and many similarities to current working practices. There may also be a number of new-found benefits.

Linux in the studio

In 2003 Dave Phillips reported the state of the art in terms of Linux audio development in a comprehensive survey article for the Computer Music Journal (Phillips, 2003). Amongst the significant benefits of using Linux and open source software, Phillips highlighted the ability (of the very interested and able few) to adapt and amend software to meet specific requirements, thanks to the GNU Public Licence³. In line with the development of the operating system, we see the emergence of a huge array of sound software. From a musical point of view - irrespective of platform - there is rarely a 'one software fits all' solution. Typically the composer will have a number of pieces of software at his disposal and may develop tools for specific tasks. Luckily some of this software is cross-platform and the paradigms of synth building, code interpreting and audio mixing have remained consistent.

As a research-led department, it would be beneficial to use the same set of tools for research and teaching. At USSS we increasingly deliver compositional tools as MaxMSP or Pd⁴ 'patches' explaining that students can use them on-site or run them at home. To begin with all the 'wiring' is hidden, students only see what they need. We have tended to make traditional tools such as chorus, flange, resonant filters etc., often modelling the well known GRM Tools plug-ins (another borrowing from the

¹ See <http://www.shef.ac.uk/uss>.

² Ricochet and M2, which is now called Resound (see <http://resound.sourceforge.net>) (Moore, 2003)

³ Gnu General Public Licence. [Http://www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html)

⁴ MaxMSP and Pd are graphic programming environments with much in common (including initial authors).

French Musique Concrète Tradition.⁵) These tools - developed out of the SYTER⁶ system - opened up many new creative pathways and allowed composers to get involved with sound from day one. There were two problems with using the GRM-Tools: their operation could only be explained outside of the system itself and, students could not use them at home. Both of these problems were solved by the use of dedicated teaching and research tools written for MaxMSP or Pd.

The open source advantage

Were web access, word processing and email our sole chores, Linux, Firefox, Thunderbird and OpenOffice would suffice. From an institutional perspective one might consider at this point that commercial software systems would be surplus to requirements. And while we do not wish to linger upon issues of copyright, illegal or cracked software or institutional finance, there are obvious benefits to a switch to open source software.

Cost savings

The purchase of multi-licenced software for audio composition, music typesetting, synthesis and analysis will normally form a significant part of a studio's budget. It is now all too easy for costs to spiral when one becomes tied to commercial products.

A move to Ubuntu Studio would take the studios out of costly commercial software upgrades for ever, though not necessarily out of hardware upgrades. Whilst speed and efficiency were always at the top of the agenda when someone else was paying, workarounds such as batch scripting and scheduling (meaning naturally the use of non-realtime processing) does not necessarily imply a backwards step. The added value of working with Linux is the consideration of what open source actually means especially when considering creative commons licences. Richard Stallman's⁷ 'free as in free speech, not free as in free beer' quote is surely something to use to positively encourage students from passing software illegally amongst themselves.

From a strategic point of view, any opportunity to save money purchasing multiple copies of software and buying into upgrade paths frees funds for tangible benefits that attract students and shift focus to the advantages and added value of the university infrastructure: acoustically treated studios, the most excellent monitor loudspeakers and high-quality microphones, performance opportunities and access to experienced composers and music software developers for example.

Access to software

It is interesting to note an apparent abundance of illegal audio software and the ability of students to illegally access the latest tools faster than the university can fund and support the legal versions. Furthermore, as software companies devote time and money to increasingly elaborate software protection systems, we indirectly foot the

⁵ The Groupe de Recherches Musicales, founded by Pierre Schaeffer in 1958 continues to operate from within the RTF building in Paris.

⁶ SYTER (Système temps réel).

⁷ The man at the heart of the GNU project.

bill for the piracy while suffering complex installation and maintenance as a bonus. Open source allows both students and the University access to the latest software. Access to a given tool set is clearly important to the student/researcher and open source tools are accessible by their very nature. Furthermore, it is possible for the university to distribute a specific software installation directly.

The marketing of 'Industry Standards'

The use of the phrase 'industry standard' is often presented as a reason to adopt one particular piece of software over another and is often used to market both software and hardware. We consider it far more important that the university teach its students theory and practice that is software independent than deliver specific packages. Can open source software be adopted to this end? Given the fact that there are many similarities between open source software and that used in the creative media industries and that a number of practitioners working in the industry are beginning to move towards open source solutions, we argue that it can. By disregarding industry standard marketing and focusing teaching outcomes around specific theory and practice, it becomes arbitrary what brand of software is used. It is only important that the teaching outcomes are achieved (whilst remaining tied to personal development planning with industry experience where appropriate).

Obviously, it is important for students to leave university with knowledge of commercial software but in a world where open source is increasingly popular perhaps we can also provide them with important open source skills and knowledge. It could also be argued that by promoting open source we are self perpetuating improvements to technologies that benefit the university directly and indirectly.

Case study: Electroacoustic composition in Ubuntu Studio

To prove DM's point that basic operating system installation was surprisingly easy, AM began by downloading a disc image of Ubuntu studio and made a full installation of the system on a surplus 2.5Ghz PC in under one hour. However, despite the fact that sound software in Linux has moved on significantly giving rise to a number of distributions specifically aimed at the audio market, do not consider that composition or teaching and learning in such an environment requires any less expertise than working with Windows or Macintosh systems. The authors were frequently reminded just how much experience they already possessed and how often academics require technical support. With DM being an accomplished programmer already fully conversant with Linux, it was left to AM to explore this new territory.

At the outset, AM required technical assistance both to understand the Linux operating system design history and to enable some of the functionality of the system particularly when it came to installing new software. Part of the problem was the sheer abundance of software available. The need for advice and information also forces one to forums and emails with other interested parties. Controlling ones system, building personal solutions to technical and sonic problems was extremely satisfying. However, it can become too easy to work on secondary solutions (such as developing a set of infinitely expandable and adaptable tools) and forget that (in this instance) an electroacoustic music composition was the primary goal.

AM's work within the Ubuntu studio distribution⁸ involved creating patches in Pd⁹, manipulating sounds that were initially recorded directly into the Ardour Digital Audio Workstation (DAW). Ardour converted stereo files into two mono files which then needed recombining¹⁰. Sounds were further developed, whilst investigating the variety of programs included in the distribution (for example, using Cecilia as a Csound front-end).

After booting up and logging in, AM ran the JACK¹¹ server. JACK sits between the kernel and user programs such as Ardour and Pd and allows for a multitude of MIDI and Audio connections to be made, both within and between programs. In many cases, other operating systems, if not told to free up the sound driver when a program is in the background, will not allow multiple audio programs to run at the same time. In Linux, one can run an audio send out of an Ardour track, into self designed Pd filters, and back into Ardour. That said, Ardour comes bundled with excellent LADSPA (Linux Audio Developer's Simple Plug-in) plug-in filters and VST wrappers are now available. AM's patches closely modelled the GRM Tools plug-in set. Broadly speaking, patches were additive or subtractive. Graphical environments such as Pd enabled the creation of sliders and buttons and worked in real-time. Text based programs such as Csound tended to afford the chaining of processes and more specific parameter control¹². Naturally one needs both and it is now possible to embed Csound as an object within Pd. Signal processing took place in the time domain using delays and filters, and the frequency domain using convolution and phase vocoders.

Clearly, Linux is not for the faint-hearted. Given its UNIX heritage, sound software of any real importance has a rather steep learning curve (true of Csound, Supercollider, MaxMSP and Pd). As we explored other programs, notably Lilypond for note processing, AM remembered his work in the 1990s with the DARMS syntax. DM reminded AM that open source software traditionally comes with full source code, and explained that compilation and library management is usually far easier with Linux than it would be with Windows. However, the sheer abundance of systems and dependencies resulted in a number of frustrations – particularly for AM - when trying to install rare oddities such as Common Lisp Music.

Teaching and learning

The incorporation of MaxMSP and/or Pd into undergraduate teaching and learning has become an essential part of many studio based courses (not least because some proficiency is generally seen as a requirement for postgraduate research). Where undergraduate courses are strongly studio focused, the considerable learning curve for those unfamiliar with Music IT is acceptable. In courses which are less studio centric, such as the undergraduate programme at The University of Sheffield, students that opt

⁸ A distribution comprises the Kernel - the lowest level of the operating system - other system packages that sit on top of the Kernel and user software. Commercially backed systems such as Ubuntu create open source and proprietary systems. The Ubuntu distribution is currently on US/UK trial shipping with a number of well known manufacturers.

⁹ AM's extensive use of MaxMSP in the past was therefore useful: 'patches' made in Max did not migrate to Pd but could for the most part be recreated.

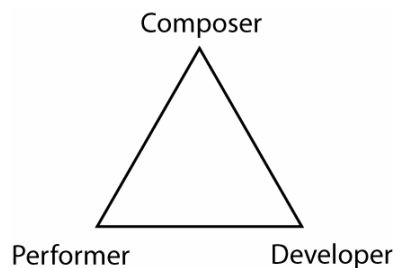
¹⁰ Solution: record directly into Pd or Audacity, a two-channel editor.

¹¹ Jack Audio Connection Kit - written by Paul Davis.

¹² There are now a number of good graphical interfaces available for the majority of what were once text-only programs.

to take a studio module are required to produce some creative output. There is not enough time to cover the basics of computer music programming. However, from a teaching and learning perspective, the ability to design specific compositional processes and add explicit user control creates a firmer bond between understanding and experience. The ability to work with software such as MaxMSP or Pd affords students the opportunity to solve problems, enables them to think logically, structure experiments, create precise documents and store information, and has application far beyond merely synthesizing or manipulating sound. At USSS we already assist postgraduate students in Music Psychology and Ethnomusicology in creating specific experiments and analytical tools.

For students, a low level understanding the operating system of a computer is less advantageous (although AM noted that there is a refreshing feeling to working with a new operating system from the ground up). AM's experience with different machines dates back some 15 years to work with Unix (Sun-Sparc), Macintosh, and PC (Windows) systems, so the Linux experience was not as daunting as it might have been (and would be to new students). And it is true, most problems are noted, or solved after careful searching on internet forums. Running alongside this investigation into Linux is the development of the Music Department's CILASS¹³ (Centre for Inquiry-based Learning in the Arts and Social Sciences) collaboratory. As well as introducing students to the sono-plastic art of musique concrète, the collaboratory affords instrumental composers the opportunity to investigate computer listening, the creation of meta-instruments and the sonic manipulation of live performers. This model (see below), once at the heart of IRCAM¹⁴, is now proving to be a successful collaborative teaching and learning paradigm at Sheffield.



The collaboratory focuses upon patches developed in MaxMSP and Pd, with wiki documentation.¹⁵

And although social networking is currently in the ascendant and its benefits to higher education relatively uncharted, USSS students are encouraged to contribute their work to the freesound¹⁶ project developed by the Music Technology Group of the Universitat Pompeu Fabra, Barcelona. In a recent Google tech talk,¹⁷ Xavier Serra of the MTG documented the potential and growth of this 'soundbook'. With some 35,000 sound bites, 377,000 registered users and 15,000 visitors per day the site is

¹³ See <http://www.shef.ac.uk/cilass>

¹⁴ Institut de Recherche et Coordination Acoustique/Musique. See www.ircam.fr

¹⁵ See <http://www.music.dept.shef.ac.uk/collaboratory-wiki>

¹⁶ See <http://freesound.iaa.upf.edu/index.php>

¹⁷ July 2007, accessed 06/08/07

being used by composers, sound artists and film producers and is of particular use to acoustic ecologists in documenting the sounds of the world. The idea of opening all ones creative work to a wider community is potentially disturbing, not least because the composition process is deeply personal. However, the consideration of sonic materials for free publication leads to careful consideration being given to process, something which composers in this area have little time for (as they tend not to weight intermediary files until many have been made and an abstract idea of the piece has formed). In this project, AM's source files were edited and uploaded to the freesound website with a number of example transformations keeping file lengths to a minimum.

As for the composition itself, this was always going to be a very impromptu assignment. Traditional (clichéd) sound sources were acquired (often with tongue firmly in cheek): a cardboard box with numerous objects rattling around inside provided noise-based gestures and textures; a plastic glass provided a tone-to-noise transformation when brutally 'played' against a concrete wall. The glass also produced a number of interesting pitched resonances. All sounds were recorded in stereo with microphones placed very close to the sources. Pd was the main tool used to transform sounds. Whilst looking somewhat similar to MaxMSP, AM found there to be a number of fundamental differences (oscillators being called *osc~* not *cycle~* for example). However, work on Pd has progressed to an extent that many of the Max objects not in Miller Puckette's original are to be found in the Pd-extended edition. The bulk of the sound processing remained basic, focusing on pitch shifting (with and without changing duration) and granulation. Long development files were then taken to Ardour for editing. Given a limited timescale, this editing is very rough around the edges and listeners can hear both original samples and transformations. AM would like to point out that the close audible relationship of the transformed sounds to their original sources (transformed sounds would normally, in his view require at least one further 'rinse') leaves the piece with little subtlety. The work (hastily entitled *linough*) is for demonstration and teaching purposes only and is not particularly representative of AM's recent compositions. Realistically, in order to create a substantial work with appropriate documentation and a substantial wiki, a six month project would seem appropriate.

However, completed transformations and accompanying Pd patches will form part of next year's student-led inquiry-based learning at level one/two and it should be noted that where the composition process is opened up with materials being prepared for sharing and process being documented in public forums (and not a personal notebook or audio diary), a composer may find they have a greater understanding of their personal compositional goals.

Conclusion

Therefore, as this project remained fairly fixed in its aspirations (to set up a system for around £500 excluding loudspeakers and microphones, that would enable sound recording, transformation and mixing stemming from a *musique concrète* tradition, useful to undergraduates, postgraduates and researcher/composers alike), having achieved what could be called partial success, there is something of an anticlimax. It could be seen as a step sideways: we are familiar with a good number of tools; our students predominantly arrive with Windows experience and we have committed a great deal of time teaching within this environment. However, there was plenty to be

gained from this experience and Linux will make a greater appearance in USSS. It will complement the USSS Windows machines which offer MaxMSP and Sibelius as well as other proprietary software. However, the question ‘is there an open source solution?’ or ‘can we make a solution?’ will now be asked before we reach into our pockets. In many respects electroacoustic composers have always appropriated (mainly popular music) tools to suit their rather peculiar needs and the creative mixing process has never really been offered more than one solution. Perhaps the Linux environment is the place where a new mixing paradigm might appear?

Finally, for AM at least, the re-exploration of older software, the re-introduction to a more ‘involved’ operating system and the discovery of some very new and exciting programs (such as CLAM – the C++ library for Audio and Music¹⁸) proved a healthy and creative ‘time-out’.

References

Brown, G. J., Eaglestone, B., Ford, N., Moore, A. (2007) Information systems and creativity: an empirical study. *Journal of Documentation*, 63: 4, pp. 443-464.

Moore, David (2003) *Real-time Sound Spatialisation: Software design and implementation*. Unpublished PhD dissertation, University of Sheffield.

Phillips, D. (2003) Computer Music and the Linux Operating System: A Report from the Front. *Computer Music Journal*, 27:4, pp 27-42.

¹⁸ See <http://clam.iua.upf.edu>