Australasian Digital Instrument Building

Andrew R. Brown

Queensland University of Technology &
The Australasian Centre for Interaction Design
a.brown@qut.edu.au

Abstract

The Australasian computer music community has both a history and significant current activity in digital instrument building. In this paper I report on digital instrument builders in Australia and New Zealand, I provide an overview of software and hardware development up to this date, and I identify themes and trends that characterise Australasian digital instrument building.

1. Introduction

The breadth of digital instrument making activity and the level of innovation in Australasia is impressive. In this paper I will describe that breadth, highlight the innovations. I discuss many of the pressing issues that confront digital instrument makers.

There has been limited documentation of the activities of digital instrument builders in Australia or New Zealand. Notable exceptions include Caroline Wilkins survey of some Australian builders (Wilkins 1997), a chapter in Michael Atherton's book on Australian instrument makers (Atherton 1990), and a limited overview of activities in New Zealand by Michael Norris and John Young(Norris & Young, 2001). I hope that by bringing the activity together in this paper the extent and importance of this area of digital instrument making will become evident and that it may stimulate further study and documentation.

2. The Scope of Digital Instruments

Because the computer, by its nature, requires setting up and customising, many musical activities with a computer might be considered instrument making. In order to contain this survey it was necessary to set boundaries about what was to be included. These boundaries were not clear divisions, but relied upon the following heuristics. Systems should be based on computer hardware or software to be included. Systems that were interactive were included, those that did not respond to human gesture (e.g. installations) were generally excluded—these systems are often documented by the Australian Sound Design Project (Bandt 2003). Systems that involved digital audio output were prioritised over instruments where digital systems controlled acoustic sound sources.

It is inevitable, despite extensive research, that some instrument makers and instruments that fit within this scope will have been overlooked and I look forward to publishing updated overviews as further information is brought to my attention.

3. History

The first use of the computer as a musical instrument in Australia was in 1951. Even though it was not interactive, Geoff Hill and Thomas Cherry's imaginative use of the SCIRAC computer to play melodies was an auspicious start to Australasian digital music making (Doornbush 2001). Today's digital instrument builders have also benefited from those who experimented with electronic instruments in the mid to late twentieth century such as Percy Granger, Ian Fredericks, Martin Wesley-Smith, Don Banks and Douglas Lilburn.

Australia can also boast the first commercial digital sampler, the *Fairlight CMI* (Computer Musical Instrument) developed by Peter Vogel and Kim Ryrie and released in 1979. The Fairlight CMI also featured one of the earliest real-time digital sequencers and waveform editors. Its development was inspired by, and built upon, Anthony Furse's work during the early 1970s on the *Qasar* digital synthesizer. Unfortunately, only prototypes of this leading work were realised (Atherton 1990).

Contemporary digital instrument makers work within the context of this history but by and large, with little systematic leveraging of the accumulated experience through teaching, mentoring or even publication. Nevertheless, the amount of activity has been significant and outstanding work continues to emerge. I will now survey some of that activity under the broad categories of public instruments, personal instruments and development toolkits.

4. Public Instruments

This section includes instrument builders who designed their instruments for others to use. Many of the instruments featured here may have been played predominantly by their creators, but their inclusion in the public categorisation highlights an attempt by the builders to design instruments with some generalisation or breadth of musical adaptation such that they might be useful beyond their own compositions. Makers in this section usually display a willingness to distribute their instruments or invite others perform on them. The instrument builders are presented in alphabetical order.

Rodney Berry has created a number of interactive systems based upon theories of Artificial Life, often in collaboration with Alan Dorin. These systems include *Feeping Creatures*, *Gakki-mon Planet*, and *Listening Sky*. More recently, he has produced *Augmented Groove* a controller interface with a DJ record-scratching metaphor (Berry 2003).

Andriy Biletskyy wrote *Doctor Webern*. This software features extensive rule-based thematic variation of a specified motif, and the ability of arrange the resulting musical fragments in a temporal score which can be played as a MIDI file. Biletskyy's software was a joitn winner of the Bourges Fourth International Music Software competition. (Bilestkyy 2000).

Andrew Brown has developed a number of software instruments including *NAIM* (Brown 2001), a probabilistic cycle-based system, the *Online Music Tools* (Brown 2002), simple web-based music composition applets, and *Jammin'* (Brown 2003), a generative music application for collaborative networked improvisation.

Paul Cohen is creating *MooZk*, a program that creates a composition as the user draws on a graphics tablet. The pieces can have a generative music background or simply respond to drawn gestures in real-time (Monro 2002).

Paul Doornbusch has been involved in developing a suite of objects for the Max electronic music environment that allow the use of multi-dimensional arrays, cellular automata, limited and controlled randomness, data routing. These are available with the Max software distribution and from Ircam.

Alan Dorin developed a program called *LiquidPrism* (Dorin 2000) that produces cyclical patterns whose structure is controlled by cellular automata. Each face of a visualised cube produces its own pattern. The user controls cellular automata activation levels with mouse clicks and movements.

Anthony Furse developed Australia's first analogue and digital synthesizer, the *Qasar I* and *II* in the early 1970s. These were contemporary with the Mini Moog (Atherton 1990).

Rainer Linz has developed a range of interactive sound installations and instruments since the 1980s that utilise specialised software and hardware components. These include the ASP1 and 2 (Analog Synthesiser Player) developed with Alistair Riddell that involved digital control of analogue electronics. Recently, Linz has developed a series of interactive Java applets (Linz 2003).

Peter McIlwain has developed a number of polished instruments in the Max/MSP environment. These include the *MatrixSynth*, a simple but flexibly modulated subtractive synthesizer based on the VCS3, *Additive Gendèr*, an additive synthesis instrument played with the mouse like a harp, *Fractured Delay*, a signal distribution system that delays and reroutes a monophonic signal via eight separate outlets, and *Cubist Audio*, a system based on Cubist painting techniques where multiple views of the same object are concurrently presented (McIlwain 2002).

Iain Mott has created a number of installation works, often with the assistance of Marc Rszewski and Jim Sosnin. These include *The Talking Chair*, where the user controls the 3D sound space with a wand interface, *Sound Mapping*, where users wheel a cart equipped with audio system, movement sensors and GPS locator, around an outdoor environment, and *Summond Voices*, an interactive intercom system (Mott 2003).

Michael Norris has developed some interesting software tools, including plug-ins for the *SoundMaker* (Norris 1997) and the *SoundScript* application using Apple's *QuickTime* libraries to process audio samples offline or in real-time (Norris 2003).

Tim Opie has developed the *Poseidon*, an MIDI controller and real-time granular synthesis software combination. The instrument utilises Fraietta's CV-MIDI controller and the jMusic software toolkit (Opie 2003).

Garth Paine has developed a series of interactive installations, what he calls responsive environments, that involve sensing whole body movements in space. These include MAP1, MAP2, REEDS and Gestation that rely on real-time manipulation of sound materials and synthesis based upon sensed movements in space, often developed with SuperCollider (Paine 2003).

Robin Petterd has developed the *Responsive Hand Media Objects*. These self-contained sonic disks contain all the hardware and software in an integrated package with a focus on sound triggered by tactile interaction. Petterd's work practice is largely in the visual arts, and the hand objects reflect this in their attractive visual appearance and subtle lighting effects (Petterd 2003).

Peter Vogel and Kim Ryrie developed the *Fairlight CMI* in the 1970s, as mentioned in section 2 above. The CMI was a pioneering workstation with sampling, wave editing, and sequencing features. Fairlight went on to produce several versions of the CMI, as well as the *Voice Tracker* pitch-to-MIDI interface and a number of video editing devices.

Ian Whalley is developing generative interactive software systems. These include ENACS (Emotion Based Non-Linear Automated Composition System), a real-time automated composition system for non-linear visual narratives and WECS (Interactive Web based Evolutionary Composition System), a web-based interactive composition system based on evolutionary principals.

Robin Whittle has developed hardware instruments that are both digital and electronic. Notable amongst these is his *Devil Fish* modification for the Roland TB-303 bass machine. Whittle has also developed an extensive number of Csound unit generators (Whittle 2003).

Rene Wooller is developing the LEMu, software for live electronic music (Wooller, Coleman & Brown 2001, Wooller 2003). His software generates music in various genres of electronica. Musical parameters are controlled during performance via mapped MIDI controllers and/or with the mouse and keyboard.

David Worrall developed *Steamer*, a real-time software environment (Worrall 1990), and a number of domes for spatialised performance of music. The domes, or geodesic tents, range in size and portability and allow for the suspension of muli-speaker systems that surround the audience within (Worrall 1989, 2003).

5. Personal Instruments

This section focuses on instrument designers who create instrument for their own use, and on instruments that may be idiosyncratic or specialised in application. It is in this section that it is most difficult to conduct a thorough survey. Included are builders whose work is noteworthy in some way and ready to hand. It clearly does not include every person who has every created a Max or AudioMulch patch for live performance.

Warren Burt has an extensive history of assembling instruments for his music making. He has especially focused on using freely available software, microtonal systems, and portable electronic and digital hardware. In recent years he has made extensive use of, and contributions toward the *SoftStep* application(Dunn 1999).

Damian Castaldi has produced a number of interactive installations including *Microphlower* and *Sight Explicit Oscillations*. His work focuses on real-time signal processing of sampled and acoustic sound sources, under partial user control (Castaldi 2003).

Roger Dean and Greg White have developed numerous instruments, often based around patches built in the Max/MSP environment, for use in live performance with their ensemble australYSIS.

Geoff Hill and Thomas Cherry developed software in the 1950s to play popular tunes of the day on SCIRAC, Australia’s first computer, at the CSIRO in Melbourne. This work was achieved using programs entered into punch cards, and if there was any interactive element it was at a very slow pace indeed (Doornbusch 2001).

Stuart Favilla has developed a number of instruments with a focus on expressivity of synthesized sound performance. Most notable is his *Light Harp* which has undergone several revisions. It features a controller with lasers and light sensors that trace strings through space which are ‘plucked’ (cut) by the player’s hand. There are also sliders and dials that adjust the synthesis attributes. The software environment is written in Max/MSP.

Donna Hewitt and Ian Stevenson have developed *Emic*, a microphone-stand interface controller that sends gestural controller data that can be mapped by software such as AudioMulch or Algorithmic Composer

Tim Kreger has developed a number of synthesis and generative music tools, often with a link to computer visualisations. These include audio filters based on cellular automata. Kreger’s work is predominantly in the Max/MSP and SuperCollider environments.

Alistair Riddell has produced a series of computer-controlled pianos (Riddel 1990, Hopkins 1991). He has also developed a number of software and microelectronics devices including the *Analog Synthesizer Player*, with Rainer Linz, that built on the ideas of Percy Granger’s Free Music machine.

Greg Schiemer has worked since the mid 1970s to develop a series of interactive systems focused on hardware with embedded software control (Schiemer 1999, Burt 1999). Notable amongst his instruments is the *Tupperware Gamelan*, and his instrument building systems including the *MIDI Tool Box* and the *Electronic Movement-to-Sound Interface*. He is currently working on a distributed software instrument using wireless networks.

Benn Woods has developed a data glove and the *InBox*, a CV-MIDI interface, which he uses in performances controlling patches written in Max/MSP.

Jeremy Yuille has developed a number of software instruments for personal use in performance, predominantly using the Max/MSP environment. Notably, he has in recent years used a game-pad controller as the interface to control the sounds coming from his laptop positioned on a music stand in front of him.

6. Toolkits

This section focuses on software libraries or hardware toolkits designed to support digital instrument making.

Ross Bencina develops the well-known *AudioMulch* software (Bencina 1998, 2003). AudioMulch is an visual programming environment for constructing audio processing patches that can be controlled in real-time by mouse, keyboard or MIDI input. It can support up to 16 channels of real-time audio input and output.

Angelo Fraietta built a CV-MIDI controller which has developed into the *Smart Controller* which is a low cost, but fully featured, controller interface (Fraietta 2001). In partnership with this hardware he has written *Algorithmic Composer*, a visual programming environment that can generate and control MIDI data. The hardware and software work in an integrated way, but can also be used separately (Fraietta 2003).

Andrew Sorensen and Andrew Brown have developed the jMusic software toolkit. It is a library of Java classes for music composition, audio signal processing and digital instrument building (Sorensen and Brown 2000, 2003).

7. Issues and Discussion

Analysis of this instrument making activity reveals a number of trends and variations between makers, instrument design and usage. In this section I will discuss these issues with a view to highlighting important considerations for the digital instrument making community.

Performability is obviously at the forefront of the mind for all instrument designers, and the range of performance interfaces in digital instruments can vary from elaborate to trivial. This variety often corresponds with a variety of gesture to sound-event mappings. Instruments with an elaborate interface often feature one-to-one gesture to sound mappings, while those with less complex interfaces often have the computer produce many sound events (a sequence or series of note for example) for each gesture. Additionally, some instruments include generative components where the instrument can act with a considerable degree of autonomy. Another performance issue is sound projection. Multi-speaker arrays are increasingly common place and decisions about how the sound will be distributed in space can have a significant impact on the musical affect of the performance. More elaborate speaker systems often have an impact on the portability and mobility of the performer. Is seems that just as the computer system itself is reduced to a lap top, that the sound system is expanding to 5.1 or even 8 speaker systems. Related to the issue performance, is the question concerning the level of material preparation prior to performance. Many digital systems enable such a highly specified construction of material (composition) that virtual or literally no interaction is required during presentation. However, for most performers this defeats the purpose of real-time control of manipulation that they enjoy. A system that can provide a range of preparation and improvisation requires is likely to achieve the widest usage amongst computer musicians

Digital instrument designers are concerned with issues of compositional process and musicology. In particular the digital representation of sound and musical structure. The chosen representation can have a significant bearing of the musical potential and capability of instruments. In particular, designers need to allow for a variety of levels of data abstraction and representation, so that the performer can make either micro or macro level changes to the musical content and structure as they desire. These considerations are particularly apparent in digital instruments. Many allow the performer to manipulate data not only at the traditional note (sound event) level, but also at meta-level of event groupings or at the micro-level of individual parameter or sample.

For many musicians, conceiving of themselves as instrument builders is, at first, strange. Many musicians start as performers, and then develop an interest in composition, and may pursue analytical tasks described as musicology. These categories of musicianship are reinforced in musical training institutions, where instrument building is rarely on the curriculum. The computer musician would be familiar with blurring the traditional boundaries between performer and composer, and so it is not surprising that many come to include instrument design and construction in their skill sets. The between public and personal instrument makers in this paper reflects a potential difference in self conception. A part of accepting oneself as an instrument builder is a consideration how others might utilise your tools, and to take the trouble to go the extra steps required to produce robust instrument usable by others.

Some form of instrument building is a part of most computer musicians lives, even if only for personal use. Therefore, the skills of construction become part of the required body of knowledge in this field. Environments such as *AudioMulch* and *Algorithmic Composer* make this a not-too-arduous task, whereas the dedication to build using a computer programming language, as required by the *jMusic* library for example, takes a bit more commitment. Building your own library of functions from scratch, as many of the builders listed above have done, requires considerable commitment to development of programming skills as an aspect of their musicianship.

Digital instrument building has a number of social and communal dimensions that I'd like to explore next. Building a community in Australiasia around digital instrument development has some way to go I believe. Organisations, such as ACMA, and festivals, such as REV, provide a basis for further work. Also, new digital instruments are well accepted by those in the growing electronica community and their festivals provide good exposure and feedback for digital instrument makers. The Australia Council has been a long-time supporter of digital instrument development, they even provided funds for the *Qasar* synth development in the 1980s and the recent Sounding Out grant round recipients included a number of digital instrument projects—including the development of a Digital Instrument Web Site that the author was involved with (Brown 2003a). However, the support from the music board of the Australia Council for digital instrument development has been patchy at best and greater and more consistent sources of financial support should be sought.

There is a serious gender imbalance amongst the ranks of digital instrument developers (at least if this survey is anything to go by) and every effort should be made to ensure that there are not systemic barriers to interested woman working in this area.

A final social issue to consider is the documentation and preservation of the digital instrument building heritage. It is an important cultural heritage, which can say a significant amount about the expressiveness and inventiveness of the Australian and New Zealand people. But, like all digital systems, the ability to maintain documents and working systems is compounded by the rapid technological change in digital media. Hopefully efforts like this paper, the Australian Music centre Archives, and the Australian Sound Design Project will assist in this effort.

8. Conclusion

There is clearly a rich history and significant ongoing activity in digital instrument building in Australia and New Zealand. In this paper, I have outlined some of that activity over the past several decades. It is certain that I have left out some contributions, perhaps even significant ones, for that I apologise in advance. But, even so, it is clear that there is considerable productivity and innovation in digital instrument building in our region.

As well as documenting some of the activity, I have drawn out from that data significant issues that arise in relation to performability, musical representation, personal relationship with technology, and the social or cultural context.

There is good reason to be optimistic about the future of digital instrument building in Australasia, given the performance to date. With greater support and organization within the community itself, there could be quite an exciting future indeed.

References

Atherton, Michael. 1990. Australian Made - Australian Played : Handcrafted Musical Instruments From Didjeridu To Synthesiser. Kensington: New South Wales University Press.

Bandt, Ros. 2003. *Australian Sound Design Project*. http://www.sounddesign.unimelb.edu.au/

Bencina, Ross. 1998. “Oasis Rose, the composition: Real-time DSP with AudioMulch.” *Proceedings of the Australasian Computer Music Conference*. Canberra: ACMA.

Bencina, Ross. 2003. *AudioMulch web site*. http://www.audiomulch.com/

Berry, Rodney. 2003. Rodney’s web site. http://www.mic.atr.co.jp/~rodney/

Biletskyy, Andriy. 2000. “Doctor Webern: A visual environment for computer-assisted composition based on linear thematism.” *Computer Music Journal*, 24:3, pp. 34-37.

Brown, Andrew R. 2001. NAIM web site. http://jmusic.ci.qut.edu.au/applications.html

Brown, Andrew R. 2002. Online Music Tools. http://musictools.ci.qut.edu.au/

Brown, Andrew R. 2003. Jammin’ web site. http://www.explodingart.com.au/jammin

Brown, Andrew R. 2003a. Digital Instrument Building site. http://digitalinstruments.ci.qut.edu.au/

Burt, Warren. 1999. “Greg Schiemer
An Email Interview with Warren Burt.” *Chroma*, Vol. 25, pp. 2-6.

Castaldi, Damian. 2003. Damian’s web site. http://www.sodacake.com/

Doornbusch, Paul. 2001. “The Music of CSIRAC.” Waveform 2001; A digital Musics Conference. Sydney: Australasian Computer Music Association, pp.25-42.

Dunn, John. 1999. SoftStep software. http://geneticmusic.com

Fraietta, Angelo. 2001. “Smart Budgeting for a Smart Controller.” *Waveform 2001: A digital musics conference*. Sydney: Australasian Computer Music Association, pp. 43-47.

Fraietta, Angelo. 2003. Angelo’s web site. http://www.users.bigpond.com/angelo_f/

Hopkins, Bart. Berstein, Alec. Riddell, Alistair. 1991. “Computer Control for Acoustic Instruments.” *Experimental Musical Instruments*. Vol. 7(1), pp. 8-11.

Linz, Rainer. 2003. Rainer’s web site. http://www.rainerlinz.net/

Mcilwain, Peter. 2002. “A Survey of Software Designs from the Sonic art Group” *Australasian Computer Music Conference 2002*. http://www.iii.rmit.edu.au/sonology/ACMC2002

Monro, Gordon. 2002. *REV Festival Review*. http://www.gordonmonro.com/misc/rev2002.html

Mott, Iain. 2003. Iain’s web site. http://www.reverberant.com/

Norris, Michael. 1997. “SoundMaker plug-ins” *Computer Music Journal*. Vol. 21 (1): 43-46.

Norris, Michael & Young, John. 2001. “Half-heard Sounds of the summer Air: Electroacoustic music in

Wellington and the South Island of New Zealand." *Organised Sound*. Vol. 6(1):21-28.
Norris, Michael. 2003. Michael's software page. http://mnorris.wellington.net.nz/software.html
Opie, Tim. 2003. Tim's granular synthesis site. http://www.granularsynthesis.live.com.au/
Paine, Garth. 2003. Garth's web site. http://www.activatedspace.com.au/
Petterd, Robin. 2003. Robin's media objects page. http://www.otheredge.com.au/objects/
Riddell, Alistair. 1990. "A Meta-Action for the grand piano." *Proceedings of the International Computer Music Conference*, pp. 395-397.
Schiemer, Greg. 1999. "Improvising Machines: 'Spectral Dance' and 'Token Objects'." *Leonardo Music Journal*. Vol. 9, p. 107-114.
Sorensen, Andrew. & Brown, Andrew R. 2000. "Introducing jMusic." *Interfaces: Australasian Computer Music Conference*. Brisbane: Australasian Computer Music Association.. pp. 68-76.
Sorensen, Andrew & Brown, Andrew R. 2003. The jMusic web site. http://jMusic.ci.qut.edu.au/
Whittle, Robin. 2003. Robin's web site. http://www.firstpr.com.au/
Wilkins, Caroline. 1997. "Innovative Power: The Australian tradition of composing music automata." *Musik Texte-Zeitschrift-fur-Neue-Musik*. Vol. 71:55-57.
Wooller, Rene. Coleman, Nick, Brown, Andrew, R. 2001. "LEMu: Live Electronic Music Generator." *Waveform 2001: A digital musics conference*. Sydney: Australasian Computer Music Association, pp. 93-95.
Wooller, Rene. 2003. LEMu web site. http://www.lemu.com.au
Worrall, David. 1990. "Streamer: A real-time performance and composition software environment." *Australian New Music Conference*, Brisbane.
Worral, David. 1989. "A Music and Image Composition System for a Portable Multichannel Performance Space : a technical overview." *Chroma* (Newsletter of the Australian Computer Music Association) Vol 1(3), p. 3-6.
Worrall, David. 2003. David's dome web site. http://www.avatar.com.au/worrall/domes/