University of Huddersfield

A thesis submitted to the University of Huddersfield in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Notational Approaches for Composing and Directing a Non-Homogeneous Laptop Orchestra

Author:
Scott Hewitt

Supervisors:
Prof. Pierre Alexandre Tremblay
Prof. Michael Clarke

February 3, 2014

Revision Information

2014/02/03 00:55:09 +0000

04c3150

04c31500057b43187d182983e73570cb31c3f4a7

Copyright Statement

Acknowledgements

I would like to thank my supervision team of Professor Pierre Alexandre Tremblay and Professor Michael Clarke for their insight and guidance. I am grateful for the willingness of the performers of HELO and HELOpg. I would also like to thank Julian Brooks, Sam Birkhead, Samuel Freeman, Alex Harker, Adam Jansch and Scott McLaughlin.

I would like to dedicated this work to Susanna and my parents, as without their encouragement and patience I would never have finished.

Typographical Conventions

Compositions included as part of the submission are indicated in **bold**.

Code included as part of the submission are indicated in *italic*.

Urls are written as `http://phd.scotthewitt.co.uk/` throughout the text.

Contents

List of Figures

Abstract

Within this composition commentary, I seek to outline my practice for composing for laptop ensembles, as well as the notational approaches I have developed to facilitate composition, direction, rehearsal and ultimately performance within an intentionally non-homogeneous laptop ensemble.

Illustrating the requirement to move beyond the current typical 'application as score and meta-instrument' paradigm, I outline my own notational approach for laptop ensemble writing and the features it offers to the operation of laptop ensembles. As a consequence of the notational approach I seek to outline the performative coding role of the player and acknowledge the compositional role it extends to the performer.

These theoretical considerations are considered within the practical operation of the Huddersfield Experimental Laptop Orchestra (HELO) and its sibling HELOpg. As a consequence of these experiments I present my preference for graph and text based notations for directing the laptop performer.

Chapter 1

Instrument Concerns

1.1 Introduction

This commentary accompanies the compositions, their respective recordings and the code listed within the appendixes. Through the discussion of my compositional activity I will explore possible methods of notating performance of a laptop instrument with a particular focus on facilitating ensemble practice. After examining the features of the laptop instrument, I will discuss my compositional activities and the notational experiments they facilitated before finally highlighting the methods I have found most satisfactory. The purpose of my compositional activity is the engagement and facilitation of performance, ideally within the chamber and social setting, with only a limited interest in the audience's experience. As a consequence of my research work I present the methods of composition I have found satisfactory for composing and directing a non-homogeneous laptop orchestra.

1.2 Purpose Of Notation Within My Work

In my personal practice, the purpose of notation is to facilitate the participation of players in delivering a performance. Secondary to this concern is the desire to communicate compositional edicts. Finally, while the scores exist undeniably as objects, the primacy of the work is the sonic outcome and the experience of player collaboration and performance.

1.3 The Laptop, Device / Tool / Instrument

With the reduction in cost of ownership and increased availability, the laptop computer has grown to a near ubiquitous nature within business, education and the home. Its original role as a business productivity tool of limited creative functionality has been expanded to offer functionality ideal for work in a variety of creative fields. While the capabilities of mobile computing platforms have increased, the corresponding device penetration now places a powerful, potentially musical, creative tool in an unprecedented number of places.

With the emergence of smartphone and tablet computers in recent years, a distinction needs to be drawn between the fully featured computer and these more limited devices. For this work, I choose the distinction to be based on the capability to compile code on the device for execution on the device. As such, the current generation of tablets and phones is largely excluded but the form factor is not permanently so. However, I would agree with Woolley (2012) that the use of the desktop computers does fit within this definition. For readers interested in mobile phone based practice Oh et al. (2010) offers an introduction.

As a tool, the laptop has seen rapid adoption within administration roles associated with

musical endeavour. The ability to document, record and then communicate results on a significantly reduced time frame enhances productivity and simplifies administration and collaboration. Within more direct musical roles, the computer has seen use as a non realtime compositional aid and a powerful facilitator of notation creation and distribution. Applications such as Sybil (Clarke et al., 2004) have established the value of computer based learning environments for music.

While the use of the laptop computer may have initially been considered as supportive to musical performance, the sight of a performer now sitting only with a laptop has become increasingly common. The flexibility and portability of the laptop as suggested in Collins (2003), to play through composed generative works, and the obvious ease of tape piece playback, outweighs previous technologies. While these are significant uses, this tool use is not of primary interest. Rather, this work considers the use of laptop as a performance instrument, its integration into ensemble practice, and primarily notation required to facilitate this endeavour.

While previous computer music performers may have seemed hesitant to identify the computer as a musical instrument (Wessel & Wright, 2002) it is interesting that external observers identify early laptop users as musicians (Ratliff, 2000). Trueman (2007) states that, “making music with laptops and performing with them is by now commonplace and seemingly here to stay”. He does, however, qualify this assertion with the observation that, “the laptop is often not thought of as an instrument even when being used to create music live”.

Performers such as Casserley (2007) have chosen to use the laptop as an instrument, augmenting its original design through the use of custom software and additional human interfaces. While the New Interfaces for Musical Expression (NIME) conference provides a useful focus for this work, due to practical considerations such as availability, cost and portability (Fiebrink et al., 2007) and the value of experience (Tremblay et al., 2007) I

choose to avoid the use of 'exotic' extended interface tools.

1.3.1 Instrument Features

Most instruments can be differentiated through mechanisms of sound creation, and further subdivided by the consequence of physical attributes; for instance, within the string family the viola is a physically larger instrument than the violin. Emmerson (1998) writes, "we expect a type of behaviour from an instrument that relates to its size, shape, and known performance practice".

Electronic instruments, with their disconnection and/or augmentation of the physical sound generation, provide a challenge to expectations and this style of classification. We could argue that the laptop instrument is included within the electronic instrument family in terms of method of generation of sound, all sounds essentially leaving the digital computer domain as a change of electrical voltage over time. However, this style of classification is unhelpful as it fails to consider some of the most interesting and key characteristics of the laptop instrument.

The definition of an instrument offered by Schaeffer (1966) is, "any device that allows us to obtain a varied collection of sound objects or varied sound objects keeping at heart the permanence of a cause", which offers inclusion to electronic instruments, though perhaps is problematic when considering the variable interface of laptop instruments.

1.3.1.1 The Variable Interface

The lack of static relationship between the interface and the sound produced is a significant feature of the laptop instrument (Schloss, 2003). As early as 1991, Puckette (1991) highlights this as a concern of computer music performance,

> There must be a direct and comprehensible relationship between the controls

> we use and the sounds we hear. (This would not be a bad thing from the audiences point of view either.) A performer who pushes a button to start a sequence is not showing us how the music was really made; all we learn about the music is what our ears can tell us.

Toolkits such as SMELT (Fiebrink et al., 2007) strive to facilitate the cementing of such relationships, however the laptop instrument is inherently open in this regard. While the consequence of the physical keyboard interaction may not be standardised, their physical location is largely static and, consequently, muscle memory can be established. Due to the subversive re-appropriation of the ASCII interface for musical instrument use (by performers such as Eric Lyon (2006)), this muscle memory is likely present in non-typical performers and perhaps held with significantly virtuosic prowess[1].

The scope of the interface is succulently summarised in Henke (2007),"the minimum difference between pianissimo and a wall of noise? One pixel, 0.03mm". This perhaps attests more to the power of interface design rather than posing inherent problems with the available interfaces.

Another new feature of the blurry laptop instrument interface is explored through the practice of livecoding where, while the performer physical interface is static the interface to the software instrument is constantly rewired and developed as required. In Collins (2003) it is suggested that, "live rewiring allows the diversion of control and generation to whatever pathway is desired". Fiebrink et al. (2010) states that "the choice of computer instrument mapping strategy or algorithm presents important compositional interaction implications", a position with which I agree entirely; indeed, a significant portion of a composition for laptop may in fact occur within a carefully biased interface design.

[1]It should be noted that such is the current popularity of the ASCII keyboard interface within society that familiarity and associated muscle memory is increasingly common.

1.3.1.2 Infinite Sonority

Croft (2007) writes that, "limits of an instrument are essential to its being perceived as an instrument" suggesting that we are required to establish the boundaries of the laptop as a musical instrument too. Croft continues, "on an instrument, almost all sounds are impossible, and of those that are possible, some are more difficult to produce than others, and this difficulty is patent in the act of performance". However, the laptop perhaps stands unique in the offering of infinite sonority, unhindered by a fixed physical method of sound creation; it offers the capability to create any sound. All sounds are equally difficult/easy to create, their creation in fact mediated through the variable interface discussed in 1.3.1.1.

Trueman (2007) relates these features of variable interface and infinite sonority while commenting on the performance spectacle issues created, "most laptop music is larger-than-life; the laptopist typically generates enormous amounts of sound, with little or no effort or continuing attention".

This infinite sonority is one of the more exciting compositional features of the laptop and a key point of interaction between composer and performer within my compositions.

1.3.1.3 Flexibility of Role

Furthermore this infinite sonority also introduces significant scope in terms of instrument flexibility and likely role. In offering the ability to play individual parts, or entire orchestrations, the laptop instrument lacks an immediate definition of its role and quantification of virtuosic performance. While other instruments, such as the organ, could be considered to offer similar functionality, they do so to a lesser degree.

Collins (2003) suggests, “four techniques for laptop music performance software design that have proved immensely powerful: presets, previewing, autopilot, and live coding”. These techniques alluding to the perception of roles, some particular to the laptop instrument.

Armstrong (2006) identifies possible laptop instrument uses, modes of performance, describing them as Digital Musical Instruments and Extended Acoustic Instruments. However beyond the primary role of sound source, the laptop instrument also offers notation control and sound projection possibilities, all contained behind a single, unified physical interface.

It is perhaps these features combined that leads to Trueman (2007) suggesting that, “...the laptop, and laptop music, is without tradition and without much of a performance practice per se”.

This lack of an established dominant laptop instrument practice (caused by the instrument flexability discussed above), further complicates the quest for a notational method. This is not to suggest that the flexibility of the laptop should be removed. Rather, that a likely per-formative method, matched with a notational approach would provide a base from which less typical practices could be notated.

1.3.1.4 Accessible Instrument

One of the interesting consequences of the laptop instrument flexibility and variable interface is that the instrument offers the opportunity for facilitating various participant roles, matched against a given performer and their capabilities. This flexibility can be used to create a customisable learning curve, tailor-made to a player’s experience and established skill, potentially offering quicker progression from a beginner state to an active and fruitful inclusion within an ensemble. The laptop instrument facilitates less

able players, while also aiding more expressive playing by the more capable[2].

Chadabe (2000) attested to this possibility in reference to computer music:

> The challenge for computer music composers in the near future will be to use their elite knowledge and skill to create situations in which members of the public without that knowledge and skill can participate meaningfully in a musical process.

Paradiso (1998) goes further, highlighting how a variable interface offers the opportunity for virtuosic performers to work alongside less skilled musicians.

> This merger has two basic frontiers; at one end, there are interfaces for virtuoso performers, who practice and become adept at the details of manipulating subtle nuances of sound from a particular instrument. At the other end, the power of the computer can be exploited to map basic gesture into complex sound generation, allowing even non musicians to conduct, initiate and to some extent control a dense musical stream.

Wessel & Wright (2002) observes that the laptop instruments offers, "low entry with no ceiling on virtuosity". While this is an exciting instrument feature, one which offers reward for practice and accessibility, it does however pose difficulties. I find myself in agreement with Ruviaro (2010) that, with over simplification of interface, "the actual laptop begins to look as an accessory, a mere processor of instructions; it appears less, or not at all, as the instrument". Certainly the laptop instrument interface and performative role can become varied and in being so change the nature of the laptop's use.

I consider the accessibility of the laptop instrument as offering the re-democratisation of music making. For instance in purely practical terms, ensembles such as L2Ork (Bukvic

[2]These issues of accessibility are also applicable to players with physical or mental disability, as well as the frequently under-rehearsed professional performer.

et al., 2010) actively seek to support accessibility and participation through reduced cost, by using open source software.

Further more, when considering that the laptop device is likely to be present in the home, offering silent, authentic rehearsal through the use of headphones, the laptop instrument perhaps offers a near ideal set of characteristics for a first instrument, further increasing accessibility. Consequently, perhaps the laptop instrument offers a real democratisation of music making and performance.

1.3.1.5 Composition Specific Instrument

Having identified these key instrument features of variable interface, infinite sonority, flexibility of role and accessibility, it is important to consider the consequence of these, in relation to compositional and performance activity. Fundamental to this, is the ability to program the laptop; altering its functionality, interface and sonority as required.

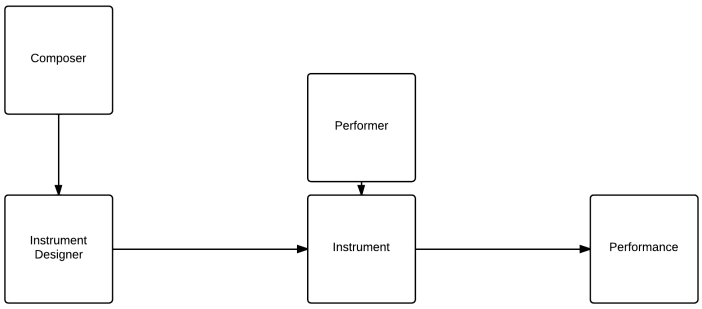

Figure 1.1: Composition structure with composer creating an instrument.

Due to its variable interface and its instrumental role flexibility, a common compositional method is the creation of a composition specific software instrument for the physical laptop instrument (work flow illustrated in figure 1.1). In Smallwood et al. (2008) the early PLorK compositions are detailed through the software written to re-

alise them. This methodology was also used by the Worldscape orchestra, as documented by Harker et al. (2008). This method offers the opportunity to establish a composition specific instrument, with the benefits of establishing near complete compositional control; however, it does encumber the composer with issues of software obsolescence, and an ongoing maintenance obligation to enable future performances, while also addressing security concerns as identified in Hewitt & Harker (2012) [3].

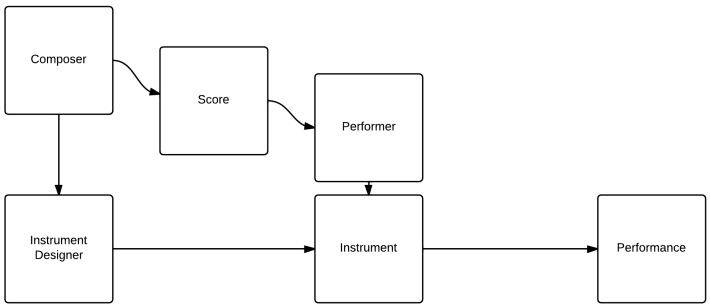

Figure 1.2: Composer creating an instrument for use with a score.

In *Wind Farm* (Gibson, 2012), composition specific instruments and interfaces are created by the composer (used with a score), for use in performance by the performers; these interfaces not only allow the creation of the required sounds but also dictate the method in which they are created (work flow illustrated in figure 1.2). In controlling the method and the interface of sound creation, the composer can also choose to deal with the issue of spectacle; Gibson (2012) in discussing its performance suggests,

> In Wind Farm, a spin performer can alleviate this feeling somewhat by exaggerating the physical production of his inertial scrolling trackpad gestures, by moving his arm more than is really necessary.

Cook (2001) is more selective, "copying an instrument is dumb, leveraging expert tech-

[3]While it is possible to envision a future performance-practice focused on historically accurate instruments, the laptop instrument's ability to alter itself would perhaps limit the need for dedicated hardware, instead relying on emulation of the instruments for such performances (Bonardi & Barthélemy, 2008).

nique is smart". The use of composition specific instruments discussed within Hwang (2012) contrasts against this statement. Within the 'What the What' composition series Hwang looks to emulate instrument interface methods (using Wii-motes) while producing synthesised sound outputs; in fact one wonders as to the reason for the use of laptop based synthesis as a mediator within the interface rather than the actual physical instrument[4].

While these instruments may be built for particular compositional use, perhaps they can actually offer greater use through redeployment in other works. Feenberg (1999) suggests that this be considered as a primary and a secondary instrumentalisation.

1.3.1.6 Modular Architecture

While composition specific instruments are effective and often useful the creation of them introduces additional concerns for the composer; the requirement of ongoing support to facilitate performance and the concern that required hardware will cease to exists and in doing so rendering the composition unperformable as the required software can no longer be run (Bullock & Coccioli, 2005). A solution to the issue of software obsolescence, encountered with the composition specific application, can be found through modular

[4]In personal correspondence with Hwang he replied,

> The use of laptop affords a transformational interface.
>
> I was interested in the different affordances of the Wii-Mote controllers as interface and meant to use the familiar gestural playing technique (instrument interface methods) (Bell ringing, Cello pluck, shaker, etc) and sonic qualities (bells, cello, shaker, etc) as a bridge for an unfamiliar audience. For What the Bells and What the Freq, I depart from the familiar sonic and gestural qualities – those from which an actual, individual handbell could not afford. So, to me, 'emulating instrument interface methods' may have been a definite beginning but also a point of departure. Using a 'laptop instrument' afforded a transformation of both gesture and sound which a traditional instrument does not (necessarily). An shaker can not change into a bell when you rotate it. If you consider the both gesture and the sound (and not even considering the coordination of control or information exchange), a laptop instrument allows a more transformable interface than those instruments that I emulated. And these combined transformations were definite compositional elements for the series.

laptop instrument software architectures, similar in construct to the MUSIC-N family of languages. Rather than supporting a custom program for each piece, a suite of applications can be maintained encompassing the functionality required.

The software *mouse-to-osc* [see appendix C] illustrates this modularity by providing functionality to send mouse movements over an OSC network. Internally it makes use of the *iilib* [see appendix C] Max library, developed by myself in collaboration with Samuel Freeman.

By creating applications with a modular design the burden of support is reduced as supported elements are reused, rather than re-implemented.

1.3.1.7 Network Connectivity

A modular instrument architecture naturally extends into network environments where the ability to link multiple computers together offers opportunities for real-time player collaboration and performance beyond the capabilities of a single computer system. Open Sound Control (OSC)(Wright et al., 2003) offers an easy method of linking applications both locally and across a network. OSC is a flexible, open protocol and consequently can be used in a number of ways. As a critical purpose of the use of the network is to allow modular style applications to be built I use the standard proposed in Hewitt & Tremblay (2008). OSC can be used alongside MIDI and audio to offer flexible and powerful inter-application and cross-machine communication. This network connectivity has also seen great exploration in producing systems such as GRENDL (Beck et al., 2010) for laptop orchestras and telepresence performances on computer based systems (Kapur et al., 2005).

The use of a network connection also offers interesting compositional game paradigms such as Scott Smallwood's composition *On The Floor* discussed in Smallwood et al.

(2008) and Angie Atmadjaja's piece *Hide and Seek*, detailed in Harker et al. (2008).

While network connectivity is a powerful compositional tool, the burden of supporting applications across diverse platforms and the required network setup is a significant and an unwelcome compositional burden and consequently unused by myself.

1.3.2 Instrument Mastery

When discussing an instrument, methods of establishing piece and player mastery require consideration - what constitutes a great performance, what would be a bad performance? Live coding grades, as listed on the TOPLAP site (Ward et al., 2009), have been suggested (rather tongue in cheek) as one method. In common, with traditional instrumental performance, the issue of practice has also received focus with rehearsal ideas explored (Nilson, 2007)[5]. Of course certain manufacturers offer accreditation in software, though this is generally not musical performance orientated.

When considering instrument mastery, the issue of virtuosity must be dealt with, Trueman (2007) observes that, "the notion of virtuosity, in some ways the antithesis of automation", yet automation is a strong asset of any computer system. Virtuosity is perhaps primarily a question of interface, referred to as players entering a state of flow as suggested by Burzik (2003). However due to the fluidity of the variable laptop instrument interface such physical interface mastery is unattainable; d'Escriván (2006) suggests,

> Reflect on how far electronic music making seems to be from the muscular virtuosity normally expended in the performance of nineteenth century music; One valid avenue of thought is that this is not a problem at all.

[5]The intensive rehearsal experiment documented in Nilson (2007) gave rise to both Hackpact and latterly Creativepact.

More in-tune with my own compositional intention the observation of Schloss & Jaffe (1993) is,

> Virtuosity is not a given in all musical traditions. Western culture, with its emphasis on the individual, is much more centered on individual accomplishments than many other cultures are.

While my personal practice is interested in player competency, it is focused on ensemble success rather than individual playing. Cooperation and collaboration are the primary tools of success and are for me the main criteria of assessment.

1.4 Within An Ensemble

As the number of individual laptop performers has increased it is only natural that these players have sought collaborative, cooperative playing opportunities amongst other performers, if only to enjoy the social element of music making (Schütz, 1951). This has led to the formation of many ensembles and of particular interest, the establishment of laptop orchestras. Trueman (2007) observes the peculiarity of this naming,

> One [the orchestra] serves to perform primarily European music from centuries ago, while the other [the laptop] is a convenient tool for editing text, crunching numbers, browsing the Web, and checking e-mail. Never the twain shall meet.

1.4.1 Laptop Roles in an Ensemble

Lacking a place within traditional instrument families, the role of the laptop requires constant negotiation, as possible ensemble utilisation varies in both role and sonority. These negotiations are informed by the previously discussed laptop instrument charac-

teristics within section 1.3.1 and through an understanding of the laptop instrument possibilities. It is therefore required that the notation conveys the purpose of the laptop instruments within the ensemble.

1.4.1.1 Subservient Role

A common role of the laptop instrument could be considered as its subservient role, that of supporting other instruments and in itself expressing limited creative player intent[6].

This processing role, perhaps expressed through either direct manipulation, responsive accompaniment or tape playback offers only a limited expression of laptop performance. My personally perceived expression of this role often suggests that the role of the laptop performer is more accurately seen as a technician and would perhaps be unrequired if a complete technological solution was applied. It should be noted that the laptop instrument deployed as an augmentation, as an extension to another instrument (such as the Hyperinstrument discussed in Machover & Chung (1989)) is not what is critiqued here, rather the lack of performative purpose of the role of the laptop player in such context.

Within my compositional practice I seek to centralise and empower the laptop performer and consequently I choose to avoid this style of relationship within my writing.

1.4.1.2 Dominant Role

Within an ensemble such as the Evan Parker Electroacoustic Ensemble the laptop performers (Lawrence Casserley, Walter Prati and Joel Ryan) enjoy equally weighted roles,

[6]This practice can easily be mapped back into the pre-computer domain, in Stockhausen's *Mixtur* the ring modulators illustrate this use especially due to the operator's involvement in varying the frequency of the modulating oscillators.

as they are dynamically negotiated, like the other performers roles, throughout the improvisations. In creating material, as well as processing sounds, players are able to provoke as well as to colour. While one presumes the player and instrument relationships are intended to be balanced, the sonority of these events often tend towards an electronic sound. The capability of the laptop instrument to overwhelm the acoustic instrument, especially when used for processing, may cause a polarisation of laptop roles. Either, the laptop as the dominant sound source or else as the subservient serving to augment other instrument's sounds.

Rather than being interested in this negotiation of role I prefer to focus on the laptop orchestra setting where the role of the laptop instrument is dominant. In removing the presence of other non laptop instruments, this also removes the default preference of typical western style notation.

1.4.1.3 Existing Laptop Orchestras

While the first modern laptop ensemble remains a contested issue, the heritage of the laptop ensemble is clearly found within the practice of the League of Automatic Music Composers and through the later works of the Hub (Brown & Bischoff, 2005). This heritage has been continued by a variety of ensembles, most notably in academic context by the Princeton Laptop Orchestra(PLorK) whos establishment is well document in Trueman (2007) and its compositional practice in Smallwood et al. (2008). Other ensembles, such as the Stanford Laptop Orchestra(SLork) and L2orK (Bukvic et al., 2010) also exist, relating themselves to an "orK" methodology of predominantly meta-instrument use.

While many of these ensembles have a formal academic foundation, ensembles such as PowerBooks Unplugged (Rohrhuber et al., 2007) also offer a performance practice based around a laptop instrument, though one which is software based rather than hardware

and software.

Other ensembles, such as the Huddersfield Experimental Laptop Ensemble (HELO), whose practice is detailed in Hewitt et al. (2010), have sought a non-uniform instrument approach. Rather than dictating the use of particular hardware and software, participant's are invited to establish their own instrument based on individual performer preference.

1.5 Early Conclusions

1.5.0.4 The Benefits Of A Laptop Orchestra

In addition to meeting the natural ensemble playing desire, other benefits of laptop ensemble performance are; "as an incubator for individual laptop performance practices" (Hewitt et al., 2010), learning live coding and developing participatory cultures (Ogborn, 2012) and as identified by Woolley (2012), "there are many obvious links between Laptop music making and the development of core and transferable skills". Additionally laptop orchestras offer ideal development grounds and research settings for software systems (Burns & Surges, 2008) (Beck et al., 2010) (Ogborn, 2012).

Laptop based orchestras offer opportunities for musical, pedagogical and research based activities even though they currently lack a formal structure of compositional communication.

1.5.1 A Composer And/Or Performer

This lack of standardised notation is perhaps due to the absence of a standard performer role.

Collins (2003) suggests that, "within the solo performance context, the role of laptop performer is a fusion between composer, performer and programmer". This presents complex notational issues to resolve, further complicated when the additional requirement of directing the ensemble is also present. The notation should offer the ability to direct the laptop musician in whatever role is required of them; be that performer, composer, instrument designer or programmer. Within the context of my compositional practice it becomes appropriate to consider the laptop musician as an improviser as many compositional activities are devolved to the performance moment, to empower the performer and facilitate ensemble performance.

1.5.2 Styles of Notation

With a lack of standardise notation and a wide variety in typical ensemble participant experience the use of historical western notation should not be considered a given, consequently experimental notation techniques will also be considered such as: The technical directions of the text score of Steve Reichs 'Pendulum Music'. The graphic notation of John Cage such as within 'Fontana Mix' and the score of Krzysztof Pendereckis 'Polymorphia'.

1.5.3 The Purpose of Notation

As attested to in Wise (2006), the challenge in composing for the laptop ensemble is complex due to the laptop features, combined with multiple player organisational requirements. Key to this difficulty is the lack of a common historical context and pre-established notational convention for either individual or ensemble playing.

Before considering the notational techniques used within my own practice, it is important to consider their potential purpose. In discussion of ensemble, Keller (2007) suggests,

"ensemble cohesion is predicated upon the musicians sharing a common performance goal, that is, a unified conception of the ideal sound" a goal that notation can facilitate. Keller (2007) goes on, "ensemble cohesion requires each performer to anticipate his or her sounds and the sounds produced by other performers". While establishing player understanding of each other is most easily facilitated through rehearsal, fully orchestrated scores can obviously facilitate establishing this cohesion, even with the infinite sonority feature of the laptop instrument.

Blacking (1981) suggests that, "musical value resides not in any piece or style of music, but in the ways that people address themselves to listening and performance". Within my compositional practice the laptop performer skill of listening, within the ensemble is essential and is to be aided and encouraged by the notation.

In establishing the importance of the performer roles what must we do to use the ensemble to perform and compose? Trueman (2007) poses this very question.

> What must we do before we can begin to make music with this [PLOrk] ensemble?
>
> (1) We need to design and construct instruments for each player (or perhaps have them do it themselves, if they are able) Further, we need to teach the players how to play these instruments, and they may need to practise to master them. (2) We need to decide how these players are coordinated, if at all.

Consequentially, my compositional work and its chosen notional methods focus on these two issues. Firstly, identifying a notation that facilitates the design and construction of laptop based musical instruments (ideally for use in both ensemble and solo settings). Secondly facilitating the participation of players with varied background in collaboration and rehearsal to deliver performances as desired.

The following chapter will detail the notational experiments undertaken, in the light of the personal biases presented previously, to explore these issues.

Chapter 2

Notation

Focused around the previously presented biases, within this chapter, I explore a variety of notational techniques. Seeking to identify a notation, that facilitates and supports laptop based musical instrument creation while also facilitates collaboration, direction and rehearsal of performances.

I initially explore video notation with the composition **Tri Play** and find significant value in its documentary nature. I go on to consider a direct notation of sonority, exploring it successfully with the graph style score of **Feedback Slide** but unsuccessfully, with the graphical score of **Christmas Carol Sonorities**. I continue to explore and eventually dismiss (for reasons discussed in 1.3.1.5) the code as score, the instrument specific composition method and in doing so establish a clear purpose for notation within my own work (see section 2.1.4).

Drawn by the familiarity of western notation my composition **Args#1** uses it successfully, but also identifies its possible representation as a graph or as data. I go on to explore graph notation with both composer-as-coder and performer-as-coder methodologies before exploring text based methods using my compositions **inCode Prime**

and **Human Shredders** with an increasing focus on performer-as-coder based performance.

2.1 A New Instrument, A New Notation

For a developing new instrument, one in which timbre[1] is predominant, western notation traditionally focused on discrete pitches mapped against time appears problematic. Through the following compositional experiments, concerns regarding notation for laptop instruments are explored and judged through my experience of directing the Huddersfield Experimental Laptop Orchestra (HELO), its sibling HELOpg, and through collaborative projects involving other ensembles.

2.1.1 Notate Physicality

Traditional western style notation can be considered as the directing of physical gesture in time, through which sonic outcomes are achieved. The score instructs the player to undertake a physical action, to realise the composer's intention. This direction of physicality is commonly undertaken through the discourse of a shared musical context and language. In the first instance, through the understanding of physical technique required to sound a required pitch, this physicality can however be even more explicitly set, through the use of fingering instructions.

This notation of physicality is enabled due to the common and static relationship of the instrument interface and the form of physical interaction required to create the requested sound, features mostly absent when using the laptop instrument.

[1] While the organ is an example of an instrument with variable timbre the main operation is still concerned with pitch and duration.

2.1.1.1 Video Notation

Within my composition **Tri Play**, a work for individual laptop performer, a Max patch is provided (*triplay.maxpat* within appendix C) which responds to OSC network traffic generated by track-pad movements[2]. **Tri Play** is designed to be played on a track-pad. The performance interface is dictated, simple and static and therefore the physical gesture is constant and consequently a standardised notation can be written for it.

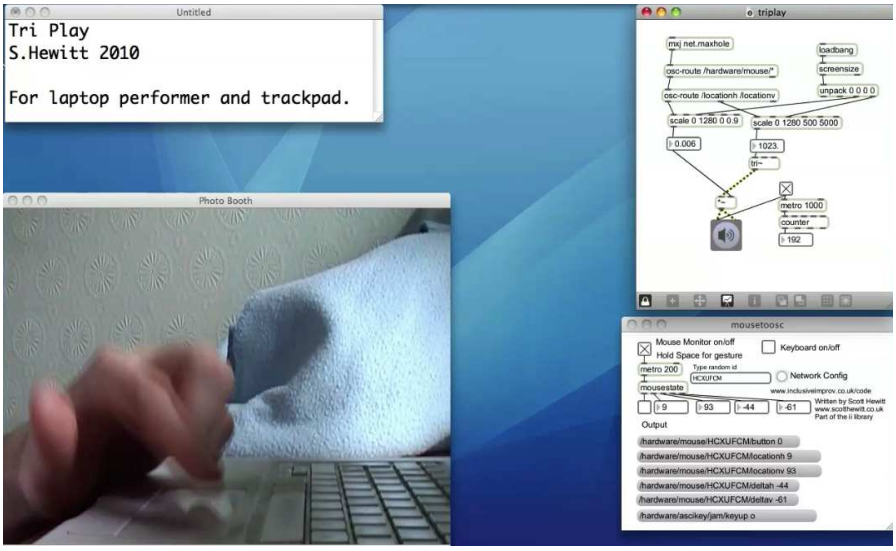

Figure 2.1: Tri Play image captured from video score.

The screen video playback presented is intended as a score, to facilitate the recreation of the piece. In itself utilising an interesting feature of the laptop instrument the ability to self-document, capture the gestural data and also record the sonic output.

Figure 2.1 shows a still image captured from the video score of **Tri Play**. The image shows the complete workings of the software instrument, as well as recording the movements of the mouse across the screen and the physical gesture on the track-pad. In doing so, the score provides all the information required to perform the score, with only a lim-

[2]In the score the track-pad movement data is captured and transmitted through *mouse-to-osc* part of *iilib*.

ited amount of interpretation possible or even required. In fact, the score is designed to facilitate historically accurate recreations of the performance, as well as offering the required information to render the performance using other technology. While this piece was written for the track-pad interface, it does not depend on the ongoing availability of the hardware or even the software, used in the initial performance.

With the audio left on the recording this style of score, inspired by the practice of live-coders (as documented in Collins et al. (2003)) is successful as a document; a record of activity.

In fact, once this experiment was completed, a significant personal observation occurred. What is the purpose of any additional performance? The score of **Tri Play** contains not only the instrument's code, but also all the data driving the audio engine. If a recreation of this piece is required, then surely the data could be extracted from the score and replayed. Though perhaps the video score could just be played back as the difference between that and an additional performance should be negligible.

After creating this composition, and its corresponding score, I actually see no reason for additional performances. The video created as a score is accurate, precise and offers no opportunity for interpretation. While the score of **Tri Play** fails in facilitating performance, which was my primary compositional goal, it does offer a complete method of notation - perhaps significant for analysis and historical purposes though not of interest to myself. Perhaps an additional score could be created, a reduction from the video score to facilate performance however this would required mediation through the video score and inevitably end up with a less precise notation.

If the failings of **Tri Play** as a method of notation are in it's over-documentation of gesture, perhaps a suitable approach to reducing the detail would be to focus on notating the sonority of a composition.

2.1.2 Notate Sonority

For an instrument capable of producing any sound, notating the sonority offers direct mapping of compositional intent with sonic output and the mediation through the process of creation as discussed in Toeplitz (2002). In creating such a score, the compromise of interactivity and computer/human bandwidth differences must be resolved. In resolving this issue of interactivity, a problem presents itself; the laptop can playback events at resolutions much higher than human performative capabilities, to the extreme as sub-sample accurate parameter variations though even standard audio rates or typical control rates are far in excess of human capability. While these parameter changes can be notated, such as in my composition **Feedback Slide**, these notations become representations, as a more accurate version of the gesture could be created within code. While the notation can be written in a human readable format, in a way that preserves all intended detail, it is with the performance as a human gesture that a loss of detail occurs, either as mistakes or just as a limitation of human bandwidth. These problems can be mediated to an extent through prolonged rehearsal such as employed within KERNEL (Toeplitz, 2002).

This method of sonority-driven notation perhaps sees fulfilment in tape-based works, where the non-realtime process allows high precision in sound design. However, I would suggest that generative code notation actually offers a more complete fulfilment, as it offers the opportunity for re-rendering to gain additional audio quality as technology progresses.

As the sonority of the laptop instrument can be dictated, exactly and preemptively to a guaranteed level of desired perfection, (as in an audio file) the purpose of notation for human consumption can not in my opinion be focused in this way, other than as a representation, such as in **Christmas Carol Sonorities** as discussed in 2.4. Otherwise, when aiming for accuracy, human performance of works for laptop instruments is

inevitably an inferior object. However these compositions retain relevance, as the incapability to perform them accurately should not prevent the experience of the challenge of doing so.

2.1.3 Notation Void of Gesture

With the concerns presented regarding the accuracy of interactivity, an approach could be conceived where the compositional intention is within the boundaries of the sonority offered, rather than the actual gestures occurring within it. This compositional decision occurs typically as the choice of instrumentation; however, in the case of my composition **Tri Play**, not only is the force (the laptop instrument) chosen, but also the software instrument and interface designed and programmed by myself. This "Ork", piece specific composition method is discussed fully in Smallwood et al. (2008).

2.1.3.1 The Coded Environment As A Score

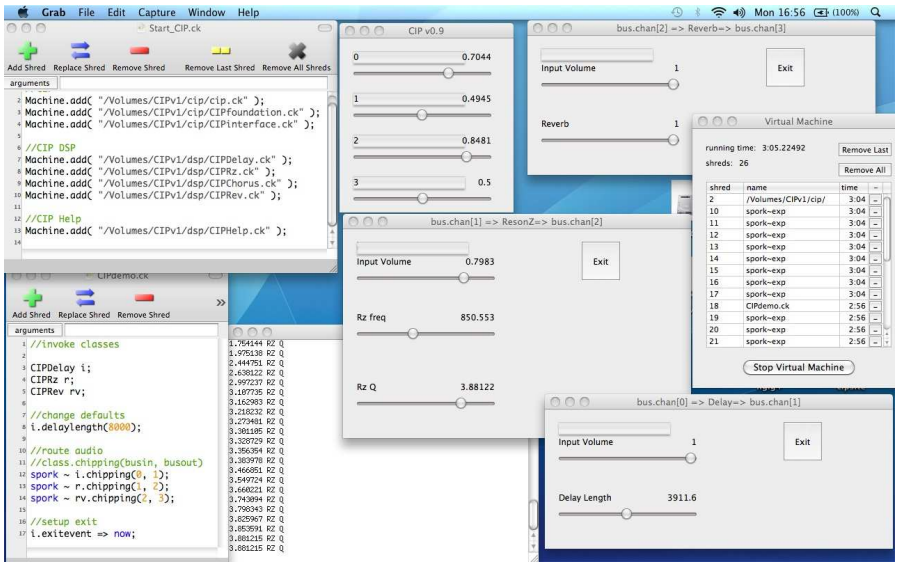

Figure 2.2: ChucK Instrument Processing (CIP) in use.

In considering the act of composition as the presentation of options, rather than the dictation of action, the development of *CIP* (ChucK Instrument Processing) was prompted (see appendix C and figure 2.2). While *CIP* does present some limitations, that could be perceived at sonority boundaries, I consider this instrument design, rather than composition, and while the instrument within **Tri Play** is simple and designed for this sole purpose, the *CIP* environment is built for reuse for the reasons discussed in section 1.3.1.6.

2.1.3.2 Facilitating Improvised Performance

While I would not claim *CIP* as a compositional outcome in itself, the designing of software instruments could now allow the more accurate stipulation of instrumentation. Rather than referring to the laptop instrument, we could make a reference to *CIP*, or a certain *CIP* setup. Stipulating the use of *CIP* setup and encouraging player performance through the dictated setup would perhaps be better considered as a bounded improvisation (McLaughlin & Tremblay, 2010)[3]. However, to claim all of the resultant sonic outcomes from *CIP* as being my compositions would be overbearing. *CIP* is an instrument, which could be used in performance or improvisation, not a composition.

2.1.4 The Purpose of Notation; An Early Conclusion

Considering the notation of gesture and sonority through **Tri Play**, and the development of *CIP*, leads me to state a number of observations regarding compositional purpose in my work.

- I consider there to be no such thing as a definitive performance, as a 'perfect' version could be made by myself notating the entire sonority as sample values

[3] Albert (2012) offers a thorough consideration of improvisation within laptop performance.

within a sound file.

- Interpretation puts at jeopardy compositional intention therefore, unless interpretation is sought, the greater precision of the sound file is available and would be preferred.

- The notation should seek to be more than a historical document of compositional intent.

- The laptop instrument offers the opportunity for the redefinition of the composer/-player/instrument designer paradigm. Notation should seek to facilitate this, not oppose this renegotiation.

As a consequence of composing **Tri Play** and programming *CIP*, I consider the primary purpose of the performer to no longer be the on-time delivery of gesture judged against a score, but rather, the suitability of the performed gesture within the context of the performance.

2.2 Western Notation

An obvious notation to consider is traditional western notation, based on using the stave. Western style notation is used for **Args#1**, a composition for five synthesis sound sources, including laptop instruments, as well as physical hardware synthesiers as shown in figure 2.3. In fact while **Args#1** was written with laptop performers in mind, it does not actually require laptop performers. The score is designed to direct the variation of three parameters in real time, while leaving the task of choosing the sonority of the instrument and the consequence of the parameter variation to the performer.

Figure 2.3: Args#1 being performed at the Week of Speakers 2008.

2.2.1 The Advantage of Familiarity

The main attraction and perceived advantage of western notation is the familiarity many performers have with it, if the common understood mappings are preserved. The **Args#1** score (see figure 2.4) is such and consequently is efficient and quick to use. Rather than explaning notional representations as required when rehearsing **Christmas Carol Sonorities** (discussed further in section 2.4), rehearsal time can be used exploring sonic possibilities and establish interactions.

While the historical familiarity can be useful, it is important to remain with the common understanding of the notation, as mappings subverting convention would rapidly undermine the benefits of using such notation.

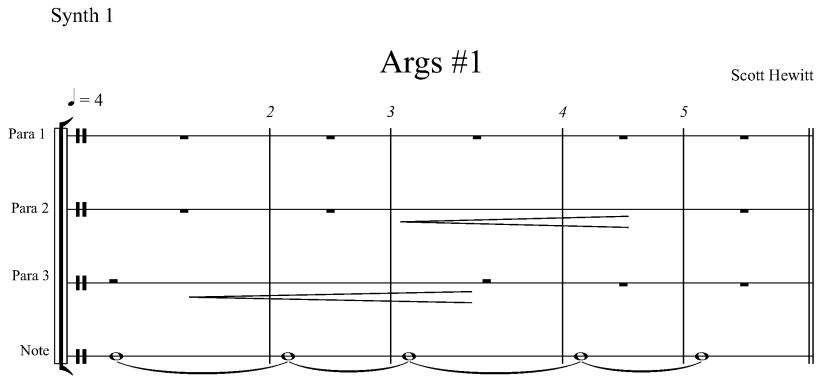

Figure 2.4: Args#1 score excerpt.

2.2.1.1 Efficiency

As a consequence of the preservation of tradition, the instrument interface may also become variable, perhaps even within the composition. Rather than mapping the physical gesture directly, such as in **Tri Play**, the gesture can be mediated through the representation on the score as in **Args#1**. Hence this style of notation can be suitable for use between a wide range of instruments, not just laptop instruments.

2.2.1.2 Pitch Notation

A difficulty with Western style notation is the representation of pitch, as the notation only offers representation of a limited set of frequencies, while the laptop instrument is capable of playing an infinite number, bounded in range only by the limitation of human hearing. Perhaps the limited number of pitches representable would be adequate for a given composition, but I suspect that such a method would offer limited reuse.

Args#1 is suitable for this kind of notation due to the insignificance of pitch within the composition; in fact, other than a rule governing when the pitch may be changed,

the selection of pitch is left to the performer. It is important for me that this is not a randomly generated pitch but rather one chosen by the performer, albeit with the expectation that the performer will call upon their performance and rehearsal experience in selecting the pitches.

2.2.1.3 Rhythmic Suitability

While there may be difficulties working with pitch, the temporal focus of western style notation is immediately useful.

Within **Args#1**, the notation indicates which player should play and for how long, while indicating changes in parameters over time. This temporal direction works even at the extreme tempo used by **Args#1**.

Args#1 demonstrates that western notation can serve effectively for laptop composition when the required performative parameters are contained with representations, mappable to pitch or rhythm. Within rehearsal the familiarity of the notation expedited preparation and performance of the piece.

However, this style of score offers little analytical assitance and loses its efficiency as more parts are simultaneously notated. While the piece is performable and the ensemble dynamic is significant in terms of pitch and sound selection, it should be noted that score-driven elements could be sequenced from a computer system with greater accuracy, a method explored within the composition **Chess onRadio**.

A concern I have regarding the use of western notation in **Args#1** is that while it facilates musical performers, perhaps it is an obstacle to other potential particpants without musical backgrounds. The notation of **Args#1** is dictating events in time, a relationship that could also be expressed as a graph.

2.3 Graph Based Notation

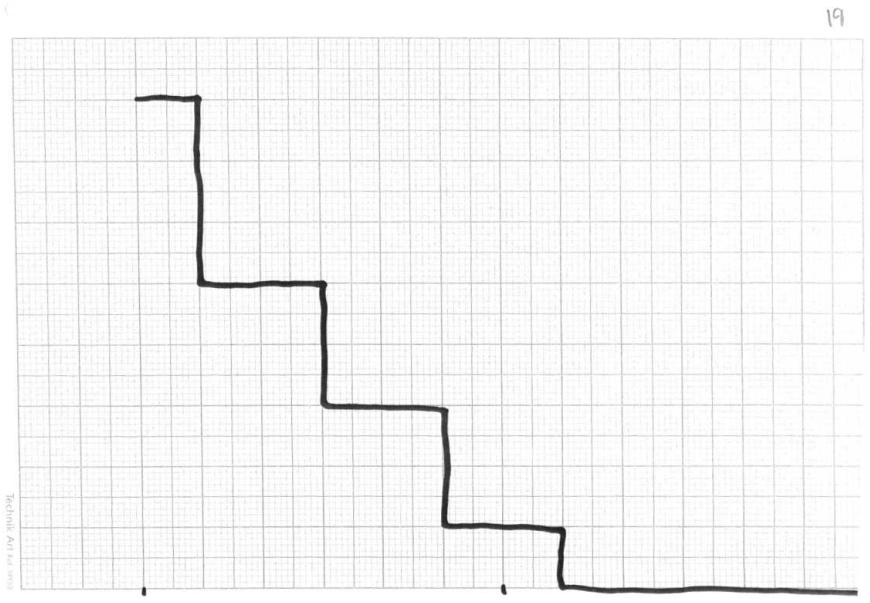

Figure 2.5: Tower Whisper part excerpt.

Feedback Slide and **Tower Whisper** are both compositions written for laptop orchestras using graph score notation. The graph representation dictates the change of parameters over time. These parameter changes are based on the compositional process of observed practice and composer intuition, there is no process other than composer choice being expressed here. Primarily, the scores are designed to direct physical gesture; in the case of the **Tower Whisper** (see figure 2.5), the physical gesture is literally notated, designed to be a direct mapping onto the stipulated interface, a MIDI slider.

The direct and literal mapping of **Tower Whisper** within the score, combined with the ready-made, ChucK-based instruments are designed to make **Tower Whisper** accessible to performers of varied backgrounds. It is the piece specific application, a compositional practice, with the composer as coder and instrument designer.

The notation is designed to be as simple as possible, with the passage of time across

the horizontal, and the manipulation of the performer's assigned slider illustrated on the vertical.

In building the synths, stipulating the interface, and notating the gesture, the sonority of the composition is fully composed. In fact, a perfect rendering of the piece could be created through a fixed recording; however, the purpose of the piece is to facilitate participation whilst it is performed. This goal is further enhanced through the stipulated performer, instrument and ensemble interactions. Performers are asked to gather around a single MIDI interface injecting an immediate collaborative aspect to the enterprise, as well as creating an interesting visual element to the performance.

However the idea of mastery seems lacking within **Tower Whisper**. It is a piece to play and is perhaps of only limited interest to more accomplished, technically proficient players; however, as an accessible piece of composition, it is a success.

2.3.1 The Software Instrument

In writing the composition **Tower Whisper** the *towersynths* (included in appendix C) were developed; this sadly creates an ongoing support requirement as discussed previously in section 1.3.1.6. These synths are written within the ChucK programming language (Wang, 2008) selected for its open source nature, as this offers greater future proofing (Puckette, 2001). In fact, the ChucK code could be used to extrapolate the relationship and re-implement the synths within another language[4].

This obligation of maintaining the performance software is undesirable, however the effectiveness of the notation is good as the notation facilitated, efficient direction in rehearsal and the consequent realisation of a performance.

[4]In line 1 of figure 2.6 the structure of the synthesizer can be seen; a saw wave oscillator is connected to a reverb unit and finally passed through a resonant filter. The interface relationship can also be extrapolated from the code in line 16, 20, 29 and 34.

```
1  SawOsc s => JCRev j => ResonZ rz => dac;
2  MidiIn min; MidiMsg msg;
3  if( !min.open(2) ) { me.exit(); }
4
5  function void freq()
6  {
7      while(1)
8      {
9          min => now;
10
11         while(min.recv(msg)){
12             if(msg.data2 == 16)
13             {
14                 if(msg.data1 == 176)
15                 {
16                     (msg.data3 * 5) + 220 => s.freq;
17                 }
18                 if(msg.data1 == 183)
19                 {
20                     msg.data3 * 0.0078 => j.mix;
21
22                 }
23             }
24
25             if(msg.data2 == 17)
26             {
27                 if(msg.data1 == 176)
28                 {
29                     (msg.data3 * 0.0078) + 0.001 => rz.Q;
30
31                 }
32                 if(msg.data1 == 183)
33                 {
34                     (msg.data3 * 5) + 220 => rz.freq;
35
36                 }
37
38             }
39         }
40     }
41 }
42
43 spork ~ freq();
44 1::day => now;
```

Figure 2.6: Tower Whisper Synth 1 Code.

The simple direction of physical gesture through the notation is within the boundaries of human performance. The instrument design and interface mapping, utilising key features of the laptop instrument, effectively facilitate the accessibility of the instrument mentioned in section 1.3.1.4. These notated gestures could be used to drive other synthesis parameters; in fact, through the use of different synthesisers the entire notated gestures could be re-appropriated; consequently, the instrument design, the synths themselves, are also significant elements of the composition.

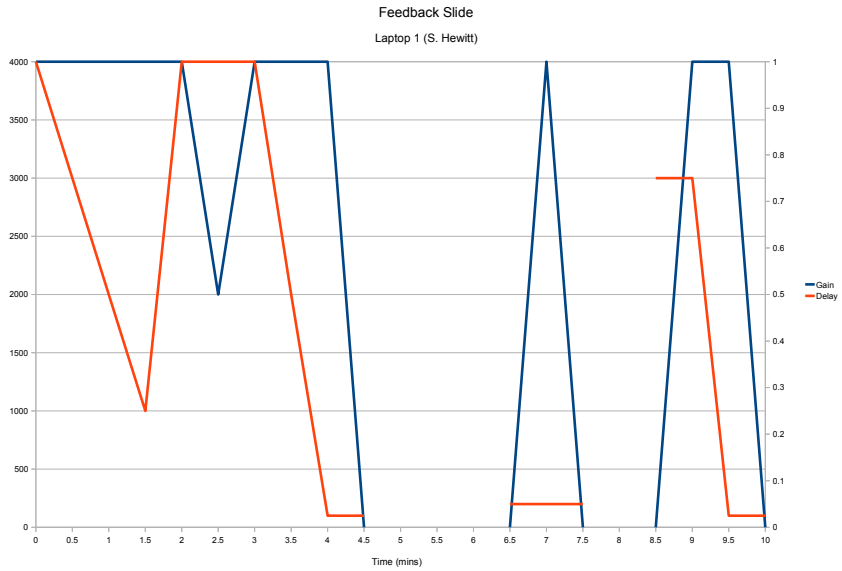

Figure 2.7: Feedback Slide graph score excerpt.

Feedback Slide shares the positive features of **Tower Whisper**'s score as it is also a graph based score that offers realtime readability and cross discipline familiarity (see figure 2.7). However, the significant development of **Feedback Slide** is in the description of the intended laptop performance instrument, rather than the provision of the software instrument. This method is discussed in Toeplitz (2002), as the transferring of the obligation to create the instrument to the performer, while retaining the composition potential of designing the instrument. The instrument design requirements are detailed

in the performer's score, for construction prior to the performance. In my own performances of **Feedback Slide**, the application *CIP* has been used while other performers have used software environments of their own choosing.

2.4 Rejection of Graphical Notation

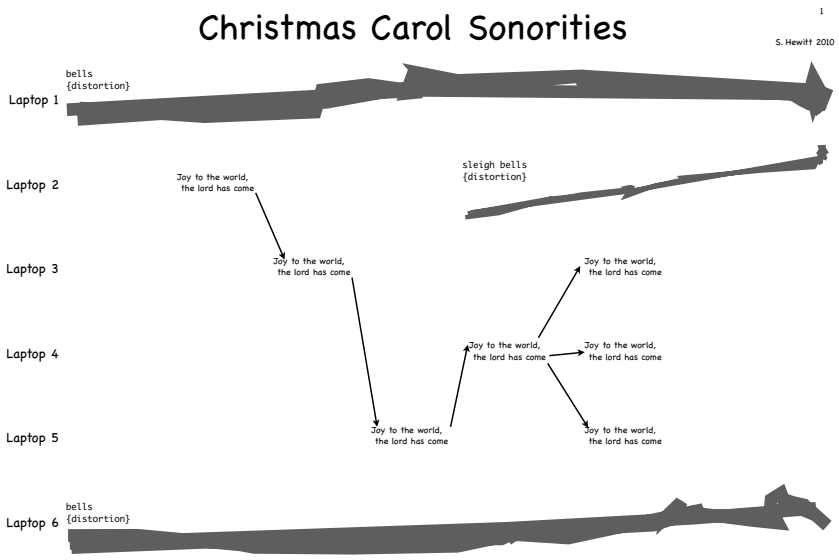

Figure 2.8: Page from orchestral score of Christmas Carol Sonorities.

An obvious development on the graph style notation is the use of graphic notation as suggested in Toeplitz (2002). Initially this was an area of interest, using the graphic element of a score to communicate additional parameter changes and also impart spectral information. This lead to the development of **Christmas Carol Sonorities**. However the piece never saw satisfactory performance; in fact, rehearsal was a struggle with performers failing to interpret the graphical elements (see figure 2.8) and appearing intimidated by its possibilities, it also appeared that solutions to these issues were unlikely to be transferable between other compositions (Hewitt & Tremblay, 2012) (paper included in

appendix D). As a consequence, no complete performance occurred and no recording is submitted. The difficulties encountered in rehearsal caused me to consider this type of notation unsuitable for the ensembles I was working with. I would suggest that graphic notation is too deep an abstraction, open to incorrect interpretation and suffers from a lack of standardisation which does not aid rehearsal.

2.5 Text Notation

Having found dissatisfaction with graphical scores, and only limited satisfaction with Western style notation further research led me to consider text based notation.

2.5.1 The Graph Score As Text

The composition **Envelope**, a graph score as written data, is a notation experiment and is not for performance. While the score is not written to be played by humans, it draws a foundational link between not only graph and text scores, but also computer code. Similarly the performer notes, at the start of **Feedback Slide**, dictating interface and laptop instrument characteristics (instrument design), can also be interpreted as code instruction (such as line 1 of figure 2.6).

Compositions such as **Feedback Slide**, **Tower Whisper** and **Args#1** could be rewritten as text scores, perhaps a list of values in time or the relationships caused by the score explicitly explained. However, the current notation methods offered for these pieces provide, realtime readability and ease of use as illustrated by the performances given. A text score could communicate the composition, but not always in a real-time readable manner as they almost invariably prompt the creation of additional user notes and pre-performance score analysis. Adding these extra features into preparation is in my case undesirable, as my intention is to offer a notation that encourages participation without

additional labour.

2.5.2 Video Notation As Text

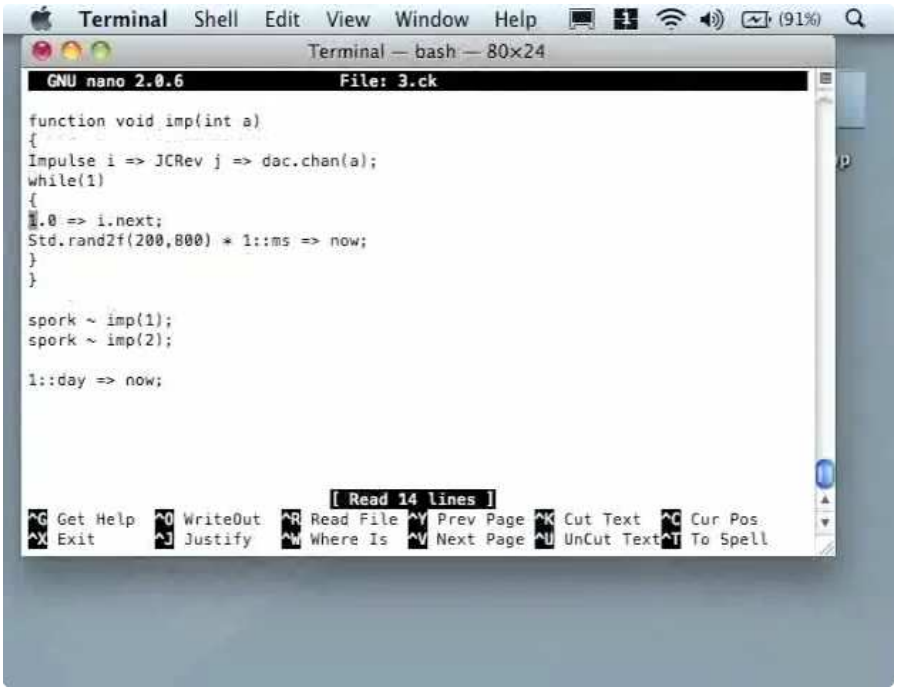

Figure 2.9: Screen shot from live-coding performance of Chuck CMJ Quarks.

The practice of live coding offers an interesting fusion between video notation and text. In **Chuck CMJ Quarks** (figure 2.9), the video recording presented is originally intended as a record of performance, a document of this performance in keeping with the TOPLAP tradition. However it could also see re-purpose as a video score, allowing the recreation of the performance. While it does not record the actually physical movement, as in **Tri Play**, it records the consequence of that action, which is the pressing of the ASCII keys. In this way it could be consider to offer more durability, as it is interface independent but also crucially the relationship of the ASCII characters and their interpretation by the ChucK programming language is known, as it is an open source language. However, while it could be used in such a way, similar difficulties to the reuse of **Tri Play**, as

discussed in 2.1.1.1, are present.

2.5.3 Text As Code

Due to the diverse participant backgrounds of laptop performers, using text scores requires the consideration of different backgrounds and the points at which they converge. In a simplified model, the typical backgrounds of performers of my works are;

- A musican, perhaps with experience of the text score tradition of Cage and Cardew.
- A computer programmer, with a background in logic and computer science.

While for musical participants the use of text scores may be considered experimental, within computer science written text instructions, referred to as code, are the predominant method of communication and direction. In fact the use of text as computer code representing algorithms has seen significant research such as in McAlpine et al. (1999). Consequently both groups of anticipated typical performers can be presumed to have an awareness of text score style constructs, and, with attention paid to the method of writing, this shared background can be exploited such as in my composition **Human Shredders** (see figure 2.10).

Even for participants outside of musical or computer science traditions, the text score offers a simple method of communicating ideas and structure important in creating accessibility. While the text score can be used to described unlimited detail, a more interesting characteristic is the openness that can be contained within it (Lely & Saunders, 2012).

Human Shredders

for humans with laptops

Scott Hewitt 2009

Do - execute sonic	Material 1	Material 2	Material 3
When			
Once or Until			
If			
Finish when			
At Start			

Figure 2.10: Human Shredders blank score.

2.5.4 Openness Of Text Scores

The score **inCode Prime**, written as an introduction exercise for newly formed laptop ensembles, seeks to exploit the openness of text score. By asking players to select a number of their choosing, from the possibilities presented within the score, different sonic outcomes can be driven from the score, encouraging repeated uses; necessary in the designed teaching role. **inCode Prime** is designed to prompt ideas of interface and flexibility in students who are selecting software to use as the instrument within the ELO methodology described in Hewitt et al. (2010) (paper included in appendix D). It is, however, of limited interest for performance, as once the numbers are selected, the outcome of the piece is fixed and could be more accurately produced by the laptop itself, rather than through the inaccurate mediation of the human performer. That is not to say the performative value of the piece is entirely removed. It still provokes instrument interface issues in participants and provides an easy first programming exercise, within

```
1, 300 10;
2, 700 50000;
3, 600 20000;
4, 500 50000;
5, 1300 10000;
6, 600 50000;
7, 874 20000;
8, 1164 60000;
9, 700 60000;
10, 647 20000;
11, 548 120000;
12, 1250 120000;
13, 1139 8000;
14, 679 8000;
15, 700 50000;
16, stop;
```

Figure 2.11: Chess on Radio Score Part 1

the instrument designer paradigm.

2.5.5 Durability Of Text Scores

In addition to the openness of the score, text scores also offer durability and portability. In placing no dependence on ongoing support from the composer, or availablity of hardware or software, text scores can contain the required elements to facilitate recreation beyond the lifetime of an implementation.

2.6 Code Notation

The differences between the text score of **inCode Prime** and **Human Shredders**, and code notation works such as **Chess onRadio**, is the moment of human interaction. Within the composition **Chess onRadio**, the code notation is designed for playback by the *onRadio* system, not directly by the performer.

The composition is contained within the data files, (score data excerpt included in figure 2.11) in a format designed to be readable to the Max object *coll*, not a human performer. The data within the file is then loaded in the application *onRadio* and transferred via OSC (over a network) to a player built instrument. Essentially, the code and the structure of the onRadio application combine to create a score that changes parameters over time (similar to a Csound score file (Boulanger, 2000)). However, unlike the methods explored above, multiple parameters can be changed, at rates in excess of human performance with expectation of increased accuracy. This method of performance was also explored in collaboration with the pianist Sebastian Berweck using *onRadio MIDI* (Berweck, 2012).

2.6.1 Features of Code Notation

By using code as notation and sequencing[5] data, an exact, robust version of a compositional idea can be conveyed. While its use is inherently realtime through playback system manipulations it could be converted to non-realtime, either faster or slower. These variable playback speeds could be used to expedite rehearsal and facilitate personal preparation.

This code notation is also extensible; additional parameters can be added and ranges changed, with the mediation of such events happening within supporting documentation.

[5]While this is very similar to traditional MIDI sequencing, it should be noted here that by operating out side of MIDI, unlimited parameter data resolutions can be used, hence the transport being OSC and the association with MIDI sequencing being unhelpful.

2.6.2 Purpose of the Score

Having created the *onRadio* system to explore the potential of code scores and being satisfied with the performances created, it causes me to question the purpose of a score within my own compositional practice.

In my opinion the use of code notation causes;

- The creation of the instrument to become of primary importance.
- The re-purposing of the performer (as events in time may now be programmed and therefore may no longer require player triggering).

Code notation offers a fixed score document which could be rendered exactly through the computer software to the laptop instrument. Therefore I would suggest it is only the possibility of variation within the instrument design that offers purpose in repeat performance. This is possible as this instrument design is unspecified and open, lacking the specification of interface and interaction stipulated within **Feedback Slide**. While questions of ownership could be formed, this is not a personal concern, rather it is the nature of the performance activity, the primacy of non-realtime instrument programming. This style of score presents the non-realtime activity of instrument design as being of primary focus. It strips away the performance activity and removes opportunities for live human interaction not only within the ensemble but also between the performer and the audience. It should however be noted that while this practice has been useful for solo laptop composition it has limited purpose within my ensemble writing.

Chapter 3

Conclusions

3.1 The Laptop Instrument

The laptop instrument offers:

- **The facilitation of participation**; through variable instrument interfaces each specifically designed for either a composition or a performer.
- **Increased democratisation**; through the renegotiation of performer, instrument designer and composer roles.
- **Access** to a potentially unlimited range of sounds, an infinite sonority.
- **A wide range of interface methods**, offering control of either micro or macro elements of the sonority.
- **No default role** within individual or ensemble practice and, consequently, a more dynamic performer / composition relationship.

These instrument features relate to my composition practice as follows:

- The current instrumental interface, that of keyboard and track-pad, offers significant scope and within my work only requires limited augmentation for particular purpose such as in the composition **Feedback Slide**[1].

- The standardisation of interface I have latterly sought (around the ASCII keyboard) should facilitate the development of specialisation and muscle memory and more significantly could lead to expression of instrument mastery.

- In offering such a wide range of methods of performance, the choice of any purpose must be clear - in my work the purpose is focused around the performer interest, not the audience (**Human Shredders**).

- There is no such thing as a definitive performance and it should not be sought, as a 'perfect' version could be rendered as sample values within a sound file requiring no performer for future performance (**Tri Play**).

3.2 The Role Of The Performer

As suggested above various performer roles are available. The performer within the previously discussed composition **Tower Whisper** perhaps should be considered as a *slider performer*, not a laptop performer; they are using the laptop as tool (as another device could be used) rather than as an instrument. In fact their interaction does not require the user interface of the basic laptop instrument rather the extended laptop instrument interface, that includes the slider. However, in the case of **Feedback Slide**, the instrument luthier role required of performers, causes a laptop performer status to be confirmed. In adopting this position a paradox occurs; the activity which characterises the nature of the performer is actually occurring outside of the performance

[1] I would suggest that this use of additional interface modification should be consider as either the performance of the interface, or the emergence of another instrument i.e. the *laptop v.2*.

activity[2].

3.2.1 Programmer or Player

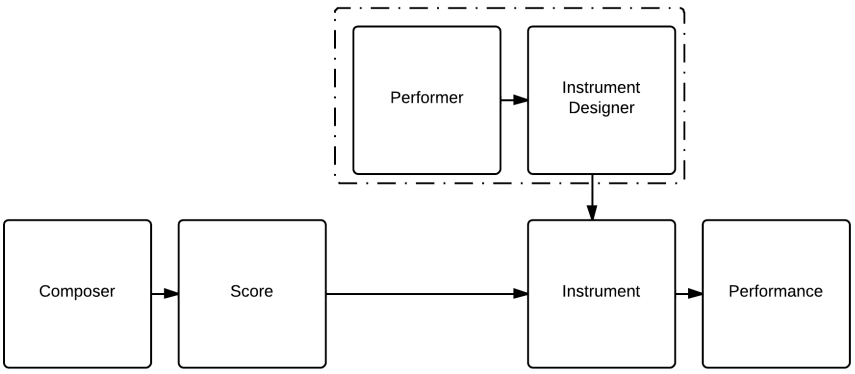

Figure 3.1: Participant role relationship within OnRadio.

As the composition **Chess onRadio** illustrates, the main performer focus can be notated away from the real time act of performance and instead redirected to the coding activity (see figure 3.1). **Human Shredders**, however, celebrates the need for human performance and performative judgement skills, despite still expecting coding skills. **Human Shredders** works well as composition by exploiting this duality in performers, expecting both musical listening skills and programming skills, not only in the tool required to perform the part, but more significantly in the writing of the parts itself. The deliberate use of familiar-looking code constructs is designed to encourage players to work within their experience of programming.

This duality of programmer/player is further developed through the common method of instrument practice, laptop performers developing and supporting performance system themselves; composers or performers are themselves expected to sustain systems suitable for their use and remain creative whilst doing so.

[2]It should be noted that performance practices such as live coding offer inclusion of the coding activity into the performance situation.

3.3 The Laptop Performer

Only in the hands of the performer does the laptop become an instrument. However players are not necessarily required for the action of playing events in time rather, their inclusion can be based on other performer roles. Within my work, this manifests alongside a renegotiation of the composer/performer roles. Rather than create a rigid part to play, the composer should seek to direct or empower the performer to create an instrument suitable for the realisation of ideas. This can occur through the devolution of choice regarding either the selection of sounds (as in the composition **Args#1**), or methods of sounding (as with in **inCode Prime**), and in doing so, allows the performer not only greater expression, but also the flexibility to engage how they wish, within whichever common ensemble grounding they choose.

Rather than judging the composition based on the quality of the performance, my satisfaction, as the composer, is in the player enjoyment in the act of performing. However, players who are choosing to perform in front of an audience are likely to be concerned with the audience's enjoyment; this is acceptable and the openness within scores can be used to achieve this.

For those seeking perfection in performance, the laptop can be used to provide exact renderings of pieces (such as in the composition **Tri Play**); however perhaps this is best achieved without a performer at all. This style of use would suggest the use of a technician rather than performer and consequently cause the laptop to be considered not as an instrument, but as a tool.

For composers and performers seeking to use the laptop as an instrument, the laptop instrument offers significant flexibility in both role and diversity in sonic participation, offering the opportunity for involvement in wide ranging ensemble activities.

3.4 Coding as Instrumentalist Practice

While in earlier electroacoustic and computer music the role of the technician existed, current laptop performance practice does not usually exhibit such a role, rather uniting the role of coder, performer and technician together. However my composition **Chess onRadio** and the *onRadio* system highlight the importance of coding within laptop performance practice. I would suggest that this is the key indicator between the laptop as a tool and the laptop as an instrument. Consequently, notation should facilitate such performance methods.

3.4.1 The Instrument Builder

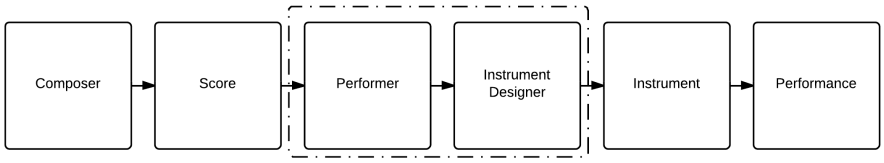

Figure 3.2: Composer, instrument, performer, score relationship within inCode Prime.

Perhaps the laptop performer should therefore be consider primarily a coder and the interaction directed by the notation should be targeted appropriately. For instance, in my composition **inCode Prime** the player is required to create a software instrument to play sound files to meet the demands of the piece (see figure 3.2). This instrument could be a simple playback method, or a more completely solution that offers performance of the entire composition. The choice of sound and prime number offered to the performer in **inCode Prime** gives purpose to repeat performance; this is the moment where the human expression is preferable, as the performance activity of playing sounds in time could be accurately done through code.

3.4.2 The Score Replacement Paradox

Chess onRadio extends this instrument designer role further, shifting the performative focus from the temporal event of the performance, to the writing of the code to be used in the performance. It is the presentation of the parameters, their range and rate of change, which is the composition, the use of which is mediated through the performer and directed into their custom (ad-hoc) built solution (see section C for example *onRadio* synths).

The realtime direction of the player's perfomance actions against the fixed passage of time, a purpose of the traditional score, is unnecessary, as the data within the software can trigger these events. Critically, if the compositional output could be rendered audio, offering perfection, what is the purpose of not producing this render?

3.4.3 To Be Played

The composition **Human Shredders** offers purpose to the performer by shifting the arrangement of sounds in time to the performer, governed by rules written by themselves and therefore not programmable in advance. As many of the scores attest (see the example scores included in appendix B) often these are written with analysis elements, when something happens, is heard, a feeling is established. These are constructs that are complicated to automate, but simple for a typical human performer to do.

In offering purpose to additional performance **Human Shredders** illustrates the key point of concern established and now present in my work. The laptop instrument offers precision, while the human performer offers approximation. The laptop can be exact, while the human can interpret and it is through that act of approximation and interpretation that the human performer finds purpose. Furthermore, this interaction offers purpose to ensemble performance; while **Chess onRadio** contains all the parts for a

single player to work with, **Human Shredders** relies on ensemble performance and player interaction.

3.5 The Coding Laptop Performer

A feature of the laptop performer's role within in my work is the expectation of instrument design. Cook (2001) suggests, "make a piece, not an instrument or controller" an approach taken to the extreme by supplying no software, leaving the performer to create an instrument of their own design with an interface suitable for their own individual use as expected in, **Feedback Slide**, **inCode Prime** and **Human Shredders**.

Consequently within my work, command of the laptop instrument is expressed by performers not only through the use of real-time musical judgement, but also through creative programming. The successfulness of a given performance is judged by the performers, while critically reflecting on the possibilities offered by their software design decisions. In celebrating the activity of the laptop instrument designer, or the performer as a coder, alternative methods of performance are also offered.

As explored through my composition **Chess onRadio** the performance activity is shifted away from the typical audience performer concert hall relationship. Paradoxically the celebrated labour is the non realtime, hidden programming activity. The practice of laptop performer coding is another method of performance which notation should facilitate.

3.6 Laptop Notation

The laptop instrument does not seek composers, performers or technicians; rather, it seeks musicians capable of fusing practices across the now artificial definitions of tra-

Western Notation		
Composition	**Advantages**	**Disadvantages**
Args#1	Familiar Quick	Discrete pitch Limited range Rigid structure

Video Notation		
Composition	**Advantages**	**Disadvantages**
Tri Play	Accurate Document Suitable for analysis	May require transcription Definitive render (closed)

Graph Notation		
Composition	**Advantages**	**Disadvantages**
Feedback Slide Tower Whisper	Accurate Simple Direct mapping to gesture	Inflexible Replaceable by code Requires stipulated instrument interface

Graphical Notation		
Composition	**Advantages**	**Disadvantages**
Christmas Carol Sonorities	Simple time representation	Requires interpretation Lack of convention

Code Notation		
Composition	**Advantages**	**Disadvantages**
Chess onRadio Tic Tac Toe onRadio Envelope	Accurate Exceeds human limitations Performance in Instrument Design	No dialogue with audience No performative action

Text Notation		
Composition	**Advantages**	**Disadvantages**
inCode Prime Human Shredders	Flexible Software agnostic Interface agnostic Familiar Code like constructs	Open to wide interpretation Requires additional preparation

Table 3.1: Advantages and disadvantages of notational approaches

ditional musical roles, hence its notation must offer opportunity for expression to all these roles. Table 3.1 is a list of explored notation styles and their advantages and disadvantages.

Traditional western style notation can be used to direct laptop performance as done within my composition **Args#1**. The immediate familiarity amongst musical participants and the rich syntax, is effective in facilitating direction and performance. However, to exploit this familiarity features of the laptop instrument and some methods of performance must remain unused such as, complex timbral transformations and extreme continuous pitch changes.

This use of graph based scores, as within my composition **Feedback Slide**, offers an effective way of maintaining a clear temporal progression while removing the unhelpful conventions of traditional western notation. Graph notation methods allow clear direction of continuous parameter changes and if required directing mapping of the corresponding physical gesture if a provided or stipulated interface is used. Additionally it is inherently suitable to be absorbed within a provided composition specific application, or to be supplied as code.

Undermining the use of graph and traditional western style notation is the ease of presequencing musical phrase in time, as code, into the laptop instrument. While these graph and traditional western notation represent events in time, the laptop instrument offers the opportunity to render the most challenging of parts to a single, simple button press. However this does not render these methods obsolete, as I have found the graph score, combined with code or the specification for code, useful for stipulating an actual desired sonic outputs.

The use of graph notation also provides ease of access for nontraditionally trained musical participants, such as computer scientists. In fact, such a style of notation offers ease of use to a wide range of participants with its simple mapping of physical action (such as

the movement of a slider as in **Tower Whisper**) against time.

These approaches work when either an application and interface are stipulated or, the parameters and architecture of software is dictated. Outside of these stipulations both graph and western style notations fail due to the lack of a default, standardised instrument. Consequently within my work I have found the graph score, combined with code or the specification for code, useful for causing the performance of a desired audio output.

When the focus of the work is not stipulating the exact sonic output, code based notation (as used within **Chess onRadio** and **Tic Tac Toe onRadio**) offers a unique performance model. The score is subsumed into the computer system, no longer for human consumption but rather to trigger action from the performer built, laptop instrument. By removing the human performer, the limitations of human bandwidth in performance cease to be relevant in relation to accuracy and resolution. Rather the code based notation offers opportunity for performer action in instrument design.

For this reason, text based scores such as those used within the compositions **inCode Prime** and **Human Shredders** have become my personal preference. The written text score offers ease of use to participants of various traditions, while remaining interface and software agnostic. In not demanding the use of an interface, through notation written for a particular interface, players are free to use tools of their own choosing. They may build an instrument of their own preference, to facilitate their own personal performance practice.

Finally any of the notational methods discussed, could be combined with the self documenting video score method (used to create the score for **Tri Play**) providing an historical document of performance for analysis.

3.7 Notation For The Laptop Ensemble

Within the laptop ensemble setting, graph based notation has proved effective in facilitating rehearsal and the communication of compositional intent. This style of notation has been used with composer supported software, such as within the composition **Tower Whisper** and also without supported software such as with the composition **Feedback Slide**. The suitability of the notation to work without a composer built application is very useful in ensemble work, as otherwise support for multiple laptop architectures and platforms would be necessary.

However the text score is again personally preferred, due to its capability of directing dynamic performer interactions and facilitating coder as performer methods. Additionally the notation is accessible to a wide cross section of performers.

3.8 Future Work

I intend to continue using graph scores to realise particular sonic outcomes and further challenging the performer/laptop instrument interface relationship. Within the ensemble setting, I intend to continue using text scores to explore inter-performer interaction while further challenging individual players with their level of instrument mastery. Using code based notation I intend to explore the possibilities of using live coding in response to instrument design instructions, not only within the single instrument context, but also in the network ensemble context.

Bibliography

J. Albert (2012). 'Improvisation as Tool and Intention: Organizational Practices in Laptop Orchestras and Their Effect on Personal Musical Approaches'. *Critical Studies in Improvisation/Études critiques en improvisation* **8**(1).

N. Armstrong (2006). *An Enactive Approach to Digital Musical Instrument Design*. PhD, Princeton.

S. D. Beck, et al. (2010). 'GRENDL: grid enabled distribution and control for Laptop Orchestras'. In *ACM SIGGRAPH 2010 Posters*, SIGGRAPH '10, pp. 84:1–84:1, New York, NY, USA. ACM.

S. Berweck (2012). *It worked yesterday: On (re-)performing electroacoustic music*. PhD, University of Huddersfield.

J. Blacking (1981). 'Making artistic popular music: the goal of true folk'. *Popular music* **1**(1):9–14.

A. Bonardi & J. Barthélemy (2008). 'The preservation, emulation, migration, and virtualization of live electronics for performing arts: An overview of musical and technical issues'. *Journal on Computing and Cultural Heritage (JOCCH)* **1**(1):6.

R. Boulanger (2000). *The Csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing and Programming*. The MIT Press, Pap/Cdr edn.

C. Brown & J. Bischoff (2005). 'Computer Network Music Bands: A history of the League of Automatic Music Composers and the Hub'. In A. Chandler & N. Neumark (eds.), *At a Distance: Precursors to Art and Activism on the Internet*, pp. 372–91. MIT Press.

I. I. Bukvic, et al. (2010). 'Introducing L2Ork: Linux Laptop Orchestra'. In *Proceedings of the international conference on New interfaces for musical expression*, Sydney.

J. Bullock & L. Coccioli (2005). 'Modernising live electronics technology in the works of Jonathan Harvey'. In *Proceedings of the International Computer Music Conference*.

C. Burns & G. Surges (2008). 'NRCI: Software Tools For Laptop Ensemble'. In *Proceedings of the International Computer Music Conference*.

A. Burzik (2003). 'Go with the flow'. *The Strad* **114**(1359):714–718.

L. Casserley (2007). 'A Digital Signal Processing Instrument for Improvised Music'. *The Journal of Electroacoustic Music* **11**.

J. Chadabe (2000). 'Remarks on computer music culture'. *Computer Music Journal* **24**(4):9–11.

M. Clarke, et al. (2004). 'Sybil: Synthesis by Interactive Learning'. In *Proceedings of the International Computer Music Conference*, Miami.

N. Collins (2003). 'Generative Music and Laptop Performance'. *Contemporary Music Review* **22**(4):67–79.

N. Collins, et al. (2003). 'Live Coding in Laptop Performance'. *Organised Sound* **8**(03):321–330.

P. Cook (2001). 'Principles for designing computer music controllers'. In *Proceedings of the 2001 conference on New interfaces for musical expression*, pp. 1–4.

J. Croft (2007). 'Theses on liveness'. *Organised Sound* **12**(1):59.

J. d'Escriván (2006). 'To sing the body electric: Instruments and effort in the performance of electronic music'. *Contemporary Music Review* **25**(1-2):183–191.

S. Emmerson (1998). 'Acoustic/electroacoustic: the relationship with instruments'. *Journal of new music research* **27**(1-2):146–164.

A. Feenberg (1999). *Questioning technology*. Routledge, New York and London.

R. Fiebrink, et al. (2010). 'Toward understanding human-computer interaction in composing the instrument'. In *Proceedings of the International Computer Music Conference*.

R. Fiebrink, et al. (2007). 'Don't forget the laptop: using native input capabilities for expressive musical control'. In *Proceedings of the international conference on New interfaces for musical expression*, pp. 164–167, New York. ACM.

J. Gibson (2012). 'Wind Farm, A Composition For Laptop Ensemble'. In *Proceedings of the Symposium for Laptop Ensembles and Orchestras*, Baton Rouge.

A. Harker, et al. (2008). 'The Worldscape Laptop Orchestra: Creating Live, Interactive Digital Music For An Ensemble'. In *Proceedings of the International Computer Music Conference*.

R. Henke (2007). 'Live performance in the age of supercomputing'. `http://www.monolake.de/interviews/supercomputing.html`.

S. Hewitt & A. Harker (2012). 'Security In Network Connected Performance Environments'. In *Proceedings of the International Computer Music Conference*, Ljubljana.

S. Hewitt & P. A. Tremblay (2008). 'Sound communication: A Standard Syntax For Inter-Application, Inter-Device And Inter- Player Communication Over OSC'. In *Proceedings of the International Computer Music Conference*, Belfast. ICMA.

S. Hewitt & P. A. Tremblay (2012). 'Notational Approaches for Laptop Ensembles'. In *Proceedings of the Symposium for Laptop Ensembles and Orchestras*, Baton Rouge.

S. Hewitt, et al. (2010). 'HELO: The Laptop Ensemble as an Incubator for Individual Laptop Performance Practices'. In *Proceedings of the International Computer Music Conference*, New York.

N. Hwang (2012). 'Composer Perspective On What The What: A Collection Of Laptop Orchestra Works For WII-Motes'. In *Proceedings of the Symposium for Laptop Ensembles and Orchestras*, Baton Rouge.

A. Kapur, et al. (2005). 'Interactive Network Performance: a dream worth dreaming?'. *Organised Sound* **10**(3):209–219.

P. E. Keller (2007). 'Musical ensemble synchronisation'. *Proceedings of ICoMCS December* p. 80.

J. Lely & J. Saunders (2012). *Word Events: perspectives on verbal notation*. Continuum, London.

E. Lyon (2006). 'Solo Concert, GEM Days Huddersfield UK'.

T. Machover & J. Chung (1989). 'Hyperinstruments: Musically intelligent and interactive performance and creativity systems' .

T. Magnusson, et al. (2011). 'DVD Content'. *Computer Music Journal* **35**(4).

K. McAlpine, et al. (1999). 'Making music with algorithms: A case-study system'. *Computer Music Journal* **23**(2):19–30.

S. McLaughlin & P. A. Tremblay (2010). 'SpectralConway: Cellular Automata Off The Grid'. In *Proceedings of the International Computer Music Conference*. ICMA.

C. Nilson (2007). 'Live coding practice'. *Proceedings of New Interfaces for Musical Expression* .

D. Ogborn (2012). 'Composing for a Networked, Pulse-Based, Laptop Orchestra'. *Organised Sound* **17**(01):56–61.

J. Oh, et al. (2010). 'Evolving the mobile phone orchestra'. In *Proceedings of the International Conference on New Instruments for Musical Expression*.

J. Paradiso (1998). 'Electronic Music Interfaces'. `http://web.media.mit.edu/~joep/SpectrumWeb/SpectrumX.html`.

M. Puckette (1991). 'Something Digital'. *Computer Music Journal* **15**(4):65–69.

M. Puckette (2001). 'New public-domain realizations of standard pieces for instruments and live electronics'. In *Proceedings of the International Computer Music Conference*, pp. 377–380.

B. Ratliff (2000). 'MUSIC; Young Improvisers at the Far Reaches of Jazz'. *The New York Times, 7th May 2000* .

J. Rohrhuber, et al. (2007). 'Purloined letters and distributed persons'. In *Music in the Global Village Conference (Budapest)*.

B. Ruviaro (2010). 'From Schaeffer to* LOrks: an expanded definition of musical instrument in the context of laptop orchestras'. In *SEAMUS National Conference, Miami, FL*.

P. Schaeffer (1966). 'Traité des objets musicaux'. *Editions du Seuil. Paris. France* .

W. Schloss & D. Jaffe (1993). 'Intelligent musical instruments: The future of musical performance or the demise of the performer?'. *Journal of New Music Research* **22**(3):183–193.

W. A. Schloss (2003). 'Using Contemporary Technology in Live Performance: The Dilemma of the Performer'. *Journal of New Music Research* **32**(3):239–242.

A. Schütz (1951). 'Making music together: A study in social relationship'. *Social Research* pp. 76–97.

S. Smallwood, et al. (2008). 'Composing for Laptop Orchestra'. *Computer Music Journal* **32**(1):9–25.

K. T. Toeplitz (2002). 'L'ordinateur comme instrument de concert aussi une question d'ecriture ?'. *Journees d'Informatique Musicale* (9):199–207.

P. A. Tremblay, et al. (2007). 'Real-time processing on the road: a guided tour of ?iks?'s abstr/cncr setup'. In *Proceedings of the International Computer Music Conference*. International Computer Music Association.

D. Trueman (2007). 'Why a Laptop Orchestra?'. *Organised Sound* **12**(02):171–179.

G. Wang (2008). *The ChucK Audio Programming Language: A Strongly-timed and On-the-fly Environ/mentality*. PhD, Princeton.

A. Ward, et al. (2009). 'LivecodingGrades - Toplap'. `http://toplap.org/index.php/LivecodingGrades`.

D. Wessel & M. Wright (2002). 'Problems and Prospects for Intimate Musical Control of Computers'. *Computer Music Journal* **26**(3):11–22.

B. Wise (2006). 'But Where Do You Put The Capo on a Laptop?'. *The New York Times, 30th April 2006* .

J. Woolley (2012). 'Using laptop composition and performance as a pedagogical tool within Higher Education'. In *Proceedings of the Symposium for Laptop Ensembles and Orchestras*, Baton Rouge.

M. Wright, et al. (2003). 'OpenSound Control: State of the Art 2003'. In *Proceedings of the conference on New Interfaces for Musical Expression*, Montreal.

Appendix A

Recordings Appendix

Audio and video on included USB drive and available online at

http://phd.scotthewitt.co.uk/media

Args#1 performed by

Track 1 Scott Mc Laughlin, Richard Glover, Scott Hewitt, Adam Janch, Joseph Kudirka and diffused by Samuel Freeman at the Week Of Speakers *June 2009*.

Feedback Slide performed by

Track 2 HELOpg *April 2010*

Track 3 The Huddersfield Experimental Laptop Orchestra (HELO) *May 2010*

Tower Whisper performed by

Track 4 HELO and The Manchester Metropolitan University Laptop Ensemble (MMULE) *Febuary 2010*

Chess onRadio performed by

Track 5 Oliver Larkin

Track 6 Samuel Freeman

Track 7 Scott McLaughlin.

Tic Tac Toe onRadio performed by

Track 8 Oliver Larkin

Track 9 Samuel Freeman

Track 10 Scott McLaughlin.

Human Shredders performed by

Track 11 HELO at the HCMF Revolutionaries *November 2009*

Track 12 HELO and MMULE *October 2009*

Track 13 The Noise Upstairs Laptop Orchestra *October 2009*

Video A The LSU Laptop Orchestra of Louisiana *November 2012*

Tri Play performed by

Video B Scott Hewitt *2010*

Chuck CMJ Quarks performed by

Video C Scott Hewitt *December 2011* included on the Computer Music Journal DVD Volume 35 2011.

Appendix B

Score Appendix

Scores included with submission and available online at

`http://phd.scotthewitt.co.uk/scores`

Human Shredders

For any number of performers with any instruments capable of playing three sounds.

Example Human Shredders scores also included.

Envelope

For a laptop. (Not for performance)

Christmas Carol Sonorities

For six laptops.

Feedback Slide

For five laptops with variable delay lines, microphone inputs.

inCode Prime

For any number of performers with laptop instruments.

Tower Whisper

For three laptops and 12 performers.

Args#1

For five synths or laptops.

Chess onRadio

For a laptop.

Tic Tac Toe onRadio

For a laptop.

Composition Index

Appendix C

Software Appendix

Software available on USB Drive and online at

`http://phd.scotthewitt.co.uk/software`

ChucK Instrument Processing

ChucK Instrument Processing (*CIP*) is a DSP instrument processing environment built in ChucK and taking advantage of the graphical MAUI elements within the miniAudicle.

`http://cip.ablelemon.co.uk`

on Radio

onRadio, built within Max/MSP, generates control data for use in driving synthesis systems built by the performer over a network connection.

on Radio Synths

I am grateful to the following performers for creating synthesisers to be driven by *onRadio*, used within *radiotuner.maxpat* as an abstraction.

Oliver Larkin *crossfeedbackosc.maxpat*

Samuel Freeman *onradio_sfreeman.maxpat*

Scott McLaughlin *scottMsynth.maxpat*

onRadioMIDI

onRadio MIDI, built within Max/MSP, generates control data for use in driving synthesis systems built by the performer over a MIDI connection.

Presented at the RMA Study Day: Collaborations in Practice Led Research, Leeds, 2010 by myself and Sebastian Berweck.

Tower Synths

The three *towersynths* synthesisers, written in ChucK (not dependent on the miniAudicle).

Tri Play

The Max/MSP synth *triplay.maxpat* used for in partnership with *mouse-to-osc*.

iilib

iilib a library of Max/MSP externals, developed in collaboration with Samuel Freeman.

mouse-to-osc

mouse-to-osc an application and set of Max patches designed to convert mouse movements to osc network data, used in **Tri Play**.

Software Index

Appendix D

Published Paper Appendix

Published papers within submission and available on USB drive and online at

`http://phd.scotthewitt.co.uk/publications`

Security and Stability in Network Connected Performances Environments

Scott Hewitt and Alex Harker

International Computer Music Conference, Ljubljana 2012

Abstract

In this paper we highlight security issues generated by the use of network connectivity in performance. We argue that an awareness of these issues can lead to more secure and stable software, in both a technical and a musical sense. Potential exploits which might compromise performance integrity are illustrated along with suggestions for methods

that alleviate such concerns.

Notational Approaches for Laptop Ensembles

Scott Hewitt and Pierre Alexandre Tremblay

Symposium on Laptop Ensembles and Orchestras, Baton Rouge 2012

Abstract

In this paper the authors will explore the notational approaches used while directing the Huddersfield Experimental Laptop Orchestra (HELO), HELOpg and other non meta-instrument based laptop ensembles. We will discuss the different notational methods used within my own compositional practice, suggest desirable notational features and suitability of such methods based on my own practice. In comparing western notation, graphic, graph, video, code and text scores we aim to identify a notational method suitable for the transfer of compositions between diverse ensembles.

HELOpg, Lessons Learned (So Far)

Scott Hewitt, Samuel Freeman, Julian Brooks

Symposium on Laptop Ensembles and Orchestras, Baton Rouge 2012

Abstract

A review of how the Huddersfield Experimental Laptop Orchestra Postgraduate Group (HELOpg) has developed its current methodology and practice through performance and collaboration since November 2009 through a consideration of the groups chosen

approach to software, hardware and sound-reinforcement strategies as developed in and informed by:

- (1) regular weekly rehearsal; (2) various performances in different settings; (3) recent recording sessions
- divergence from more common laptop orchestra approaches
- defensive methods to ensure performance capability even through device failure

Building on lessons learned (so far), HELOpg outline how their practice might inform the development of innovative models for improvisation which, whilst affording furtherplayer interaction, will not affect each individuals conceptual and sonic identity. HELOpg also introduce the SLIME System, a new methodology being developed by the group for use in networked performance.

HELO: The Laptop Ensemble As An Incubator For Individual Laptop Performance Practices

Scott Hewitt, Pierre Alexandre Tremblay, Samuel Freeman and Graham Booth

International Computer Music Conference, New York 2010

Abstract

In this paper we seek to outline the methodology and philosophy of the Huddersfield Experimental Laptop Orchestra (HELO). Placing the ensemble in context of similar work, we discuss the Do-It-Yourself (DIY) laptop instrument design paradigm, and the incubatory benefits that arise from participant-centered approach to ensemble rehearsal and performance.

Sound Communication:

A Standard Syntax For Inter-Application, Inter-Device And Inter- Player Communication Over OSC

Scott Hewitt and Pierre Alexandre Tremblay

International Computer Music Conference, Belfast 2008

Abstract

This paper offers a standard message format for easy intercommunication over a network, between laptop performers within a drop-in improvisation session. It is based on OSC-like reserved namespaces, namely /test, /setup, /chat, /app, /user, /time, /documentation and /hardware. We detail the syntax of use and provide worked examples. The proposed standard offers interoperability, extensibility and flexibility across network applications.