Epistemic Tools

The Phenomenology of Digital Musical Instruments

Thor Magnusson

Submitted for the degree of Doctor of Philosophy

University of Sussex

July, 2009

In memoriam
Andrew Gartland-Jones (1964-2004)

Declaration

I hereby declare that the work presented in this thesis is original and has not been submitted for a degree, either in the same or different form, to this or any other university.

Signature:

Supervisors: Dr. Chris Thornton & Dr. Nick Collins

Examiners: Dr. Matthew Fuller & Dr. Geraldine Fitzpatrick

Acknowledgements

I want to thank *many* people. Andrew Gartland-Jones, for inviting me to carry out this research and work at the University of Sussex. Drew was a hugely inspirational researcher and musician, whose untimely death was a great loss for our community. I want to thank my long-time collaborators Enrike Hurtado Mendieta and David Bausola for being wonderful friends and inspirational characters, who own more in this thesis than could ever be explained. My supervisors Chris Thornton, Nick Collins, and Marcelo Wanderley have been immensely benevolent forces in the process of this study. I would like to thank the people in my research committee, whose yearly meetings were always extremely educative and inspiring: my supervisors, Sam Hayden, Geraldine Fitzpatrick and Graham McAllister.

Thank you Tom Ottway, Greg Hooper, Graham Wakefield, Christopher Frauenberger and Elsa Eiríksdóttir for excellent reading, comments and discussion; and Andy Hunt, Sergi Jordà, Alex McLean, Nick Collins, and Newton Armstrong for many and often long email conversations. I am also heavily indebted to the people that participated in the surveys and interviews that are reported on in this thesis. Too many to be named here, but you know who you are.

I have been lucky to be surrounded by excellent colleagues at the University of Sussex: Chris Kiefer, Anna Jordaneos, Matthew Yee-King, Alice Eldridge, Cian O'Connor; at the University of Middlesex: Tony Gibbs, Nic Sardiland, Gordon Davis, Magnus Moar and Stephen Boyd Davies (the last three of who were my old mentors whilst studying for my MA, and they still are!); at the University of Westminster: Rupert Latimer and John Eacott; and last but not least at the University of Brighton, where Holger Zschenderlein, Kersten Gladien and Jean Martin have been extremely supportive during the last stages of writing.

Digital music centres and academic institutions other than those of Sussex and Brighton have been instrumental in outlining the focus of this research: STEIM in Amsterdam, The Digital Research Unit of Huddersfield (DRUH), Buchsenhausen in Innsbruck, the IDMIL Lab at McGill in Montreal… Thanks.

I want to thank the members of my improvisation bands for providing the platform for testing the ixiQuarks in hard-core action, and for being so wonderful musicians.

Furthermore, it is important to give credit to James McCartney and the developers of SuperCollider for making a system that serves equally as a toolbox enabling the thinking about music, composing music, making musical instruments, and performing music.

The greatest thanks go to my family and friends for all the support, in particular my girls in Brighton, Birta and Mirra, who, during the period of this research, endured the company of an embodied being whose absentmindedness at times must have made them feel more like being in the presence of a zombie. I hope I can pay back that patience.

Finally, since this is beginning to sound like a certain British actress at the Oscar's, I want to thank the gods for this imperfect world, so flawed indeed that one cannot even tune a musical instrument properly. These imperfections makes life truly interesting and worthy of a study.

Abstract

Digital music technologies, and instruments in particular, are the result of specific systems of thought that define and enframe the user's creative options. Distinctive divisions between digital and acoustic instruments can be traced, contrasting the conceptually based design of software with the affordances and constraints of physical artefacts. Having lost the world's gift of physical properties, the digital instrument builder becomes more than a mere luthier. The process of designing and building the instrument is transformed into a process of composition, for it typically contains a greater degree of classification and music theory than its acoustic counterpart.

Part I of this thesis begins by framing musical systems in the context of the philosophy of technology. Here technological conditions are questioned and theories introduced that will assist the investigation into the relationship between creativity and technology. After this general grounding, the ramifications digital technologies pose to the human body are explored in the context of human expression through tool use. The human-machine relationship is described from phenomenological perspectives and relevant theories of cognitive science. This analysis serves as a foundation for the concept of *epistemic tools*, defined as the mechanism whereby techno-cultural models are inscribed into technological artefacts. The cultural element of tool use and tool origins is therefore emphasised, an aspect that is highly relevant in musical technologies. Part I thus frames the material properties of acoustic and digital instruments in relation to human culture, cognition, performance and epistemology.

Part II contextualises these theories in practice. The *ixiQuarks*, the live improvisation musical environment that resulted from the current research, are presented as a system addressing some of the vital problems of musical performance with digital systems (such as the question of embodiment and theoretical inscriptions), proposing an innovative interaction model for screen-based musical instruments. The concept of *virtual embodiment* is introduced and framed in the context of the ixi interaction model. Two extensive user studies are described that support the report on ixiQuarks. Furthermore, comparative surveys on the relationship between expression and technology are presented: a) the phenomenology of musical instruments, where the divergence between the acoustic and the digital is investigated; b) the question of expressive freedom versus time constraints in musical environments is explored with practitioners in the field; and c) the key players in the design of audio programming environments explain the rationale and philosophy behind their work. These are the first major surveys of this type conducted to date, and the results interweave smoothly with the observations and findings in the chapters on the nature and the design of digital instruments that make up the majority of Part II.

This interdisciplinary research investigates the nature of making creative tools in the digital realm, through an active, philosophically framed and ethnographically inspired study, of both practical and theoretical engagement. It questions the nature of digital musical instruments, particularly in comparison with acoustic instruments. Through a survey of material epistemologies, the dichotomy between the acoustic and the digital is employed to illustrate the epistemic nature of digital artefacts, necessitating a theory of *epistemic tools*. Consequently *virtual embodiment* is presented as a definition of the specific interaction mode constituting human relations with digital technologies. It is demonstrated that such interactions are indeed embodied, contrasting common claims that interaction with software is a disembodied activity. The role of cultural context in such design is emphasised, through an analysis of how system design is always an intricate process of analyses, categorisations, normalisations, abstractions, and constructions; where the design paths taken are often defined by highly personal, culturally conditioned and often arbitrary reasons.

The dissertation therefore dissects the digital musical instrument from the perspectives of ontology, phenomenology and epistemology. Respective sections in Part I and Part II deal with these views. The *practical* outcome of this research – the ixiQuarks – embodies many of the theoretical points made on these pages. The software itself, together with the theoretical elucidation of its context, should therefore be viewed as equal contributions to the field of music technology. The thesis closes by considering what has been achieved through these investigations of the technological context, software development, user studies, surveys, and the phenomenological and epistemological enquiries into the realities of digital musical instruments, emphasising that technology can never be neutral.

Table of Contents

List of Figures and Tables

Figures

Tables

PART I

Chapter 1

Introduction

1.1 Prelude

This thesis investigates the function and role of digital systems in contemporary music composition and performance. It explores the nature of our digital instruments, the embodied relationship we have with them, and what kind of conceptual models we design into the functionality presented to their potential users. The aim is to question the technology used in today's music from a philosophical standpoint. This involves a threefold investigation into ontological, phenomenological and epistemological perspectives on technology. Technology itself can be considered as an extension of the human body on the one hand, and an external tool for thinking and expressing artistic content on the other. The three philosophical threads, and these two core focal points, will be picked up and explored repeatedly in the different chapters of this thesis. The aim is to engage coherently and descriptively with one of the most difficult areas in human-machine relations, namely musical performance systems.

This work encapsulates the results of what amounts to a decade of research in the field of *music technology*: a highly interdisciplinary field where music, engineering, computing, art, culture and philosophy merge. The background of this work is philosophy and music. In the late 90s, through an interest in the philosophy of mind, I decided to study the wicked craft of programming computers. As a natural derivation of the understanding of programming, I quickly realised that one could write music in the form of code. Not only music, but also musical instruments, generative music systems, and interactive musical environments. A philosophical enquiry into these phenomena is a natural methodology for me, and one that has not been traversed often in the field of music technology. I argue that for a clear understanding of music technology we need to explore the concept of technology in general, and this is where we embark on our journey [chapter 2]. We also need to understand how musical instruments extend the human body, and what alterations take place when our instruments become digital and software based. The phenomenology of human technologies, where the subject of embodiment is explored, is therefore necessary [chapter 3] before we can enter the central topic of epistemic tools where the pronounced conceptual and epistemic nature of digital musical instruments is identified [chapter 4]. The aforementioned chapters make up **part I** of this thesis. It paves the way for **part II**, which contains a more empirical and practical side of this research: the analysis of digital music technologies and their implications for our musical work [chapters 5 to 8].

This thesis therefore aims to identify and outline certain properties innate to digital musical instruments. A general trend in the discourse of digital tools is to use a loose notion of "embodiment" to signify the main problems of human-machine interaction. But what does this really mean? Furthermore, if some of our musical instruments have become entirely virtual, what level of intelligence and musical modeling is inscribed into the instrument itself? This requires a comparative study, and what follows is a clarification of the different nature of our acoustic, electronic and digital tools, through an analysis of such tools in general, and the ixiQuarks[1] software in particular. Consequently, by virtue of our journey through the philosophy of technology and phenomenology, we are able to develop the concepts of epistemic tools and virtual embodiment, both of which are useful when designing, composing for, or performing on digital musical instruments.

Having lost the acoustical world's gift of physical properties, the digital instrument maker becomes more than a craftsperson. The process of designing and building an instrument becomes at the same time a process of composition, for the digital instrument typically contains greater degrees of classification and music theory than its acoustic counterparts. Conversely, in computer music composition involves the channeling of human actions into parameters of a symbolic machine, to the extent that the composer enters the realm of engineering. Mapping becomes composition and vice versa. This distinguishing feature in the nature of digital musical systems motivates the investigation of epistemic tools.

The practical side of this research has resulted in the ixiQuarks software package [further explored in section 1.3]. Through its design and related research, this work has contributed to the field of music in general and improvisation in particular. The ixiQuarks propose ways in which improvisation can flow more easily whilst using digital tools in a live context. The focus is on screen-based interfaces, but the research does not exclude physical interfaces. In collaboration with the Owl-project,[2] physical interfaces are being developed that plug into the ixiQuarks environment and allow for more variation in embodied control over the digital instruments. However, it should be emphasised that the principal focus of this thesis is the software engine of the musical instrument. I assert that it is here that the unique qualities that define the digital instrument are located. One of the main contributions of this thesis is therefore the emphasis on the epistemic nature of digital tools, an approach that has been rare in previous writings on digital music systems.

---

[1] The ixiQuarks can be downloaded freely (both application and source code) here: www.ixi-audio.net
[2] http://www.owlproject.com

In short, the current agenda relates to the following quote by the French philosopher Bernard Stiegler: "Technical evolution results from a coupling of the human and matter, a coupling that must be elucidated: technical systematicity is here embedded in a 'zootechnological' determinism" (Stiegler 1998: 46). Since culture and technology are so intrinsically intertwined, usually through concealed structures, my aim is to elucidate this relationship and attempt to bring to light some of the factors that define much of contemporary musical software (hence musical culture) in particular, and digital culture in general. However, in this process, determinisms of all kind will be questioned.

1.2 The Research Field

Electronic music is perhaps the ideal cultural domain suited to the task of analysing how technology affects human performance: hardly any music reaches our ears today without having been touched by a computer and its algorithms. Not only is the computer used in recording studios, live venues and radio stations, but it is increasingly preferred by the musician as the instrument and/or compositional environment of choice. The design of computer music software has therefore become an integral part of our cultural sphere, with the result that the designer of a technological artefact (be it a hardware box or software) can seem to be the sole initiator of various musical styles, forms and social structures.[3] Furthermore, given the subtlety of musical expression in terms of bodily dexterity and the sophistication of the acoustic instruments as sound generators with complex timbre, it is evident that the design of digital instruments is a great challenge (Miranda & Wanderley 2005). If musical instruments are the peaks of human inventiveness, design and engineering at any one time (Moog 2002) digital musical instruments are an ideal point of convergence between the hard sciences of engineering, programming, hardware design, user interface design and software production on the one hand, and the arts of analysing, composing and performing music on the other (Norman 2004).

Creating musical tools and instruments for the computer is a thus a difficult but immensely interesting endeavour where important areas of human psychology and cutting-edge areas of software engineering and ergonomics meet (Wanderley & Orio 2002). It can easily serve as the ideal case study of human-machine interaction, and one that can clearly appertain to other fields of technology, for the following reasons:

Space and organisation. Musicians are used to working in studios full of equipment, such as mixers and rack devices, spatially laid out, where the production logic is designed with careful attention to the ergonomics of temporal and spatial workflow. The simulation of the professional studio on the computer screen has been problematic and often resulted in

---

[3] The picture is more intricate than this, as we will see later. In fact, as Georgina Born (1985) has demonstrated, technology's relationship with music can be highly political and result in cultural practices where certain groups are ostracised at the favour of the dominant discourse.

frustrating and disembodied work processes experienced particularly by musicians used to the physical devices.

Embodied action Musicians have trained themselves to play their instruments over a long period of time, and through this process the instrument has become almost a prosthetic extension of their body. Dexterity or finesse in motor control and knowledge of the instrument's subtleties define a good instrumentalist. When creating digital musical instruments, much of this embodied knowledge is lost. Musicians are increasingly dissatisfied by the state of digital interfaces, which has resulted in a new research field often called NIME (New Interfaces for Musical Expression)[4] that tries to respond to the question of embodiment, control and mapping in the domain of digital instruments.

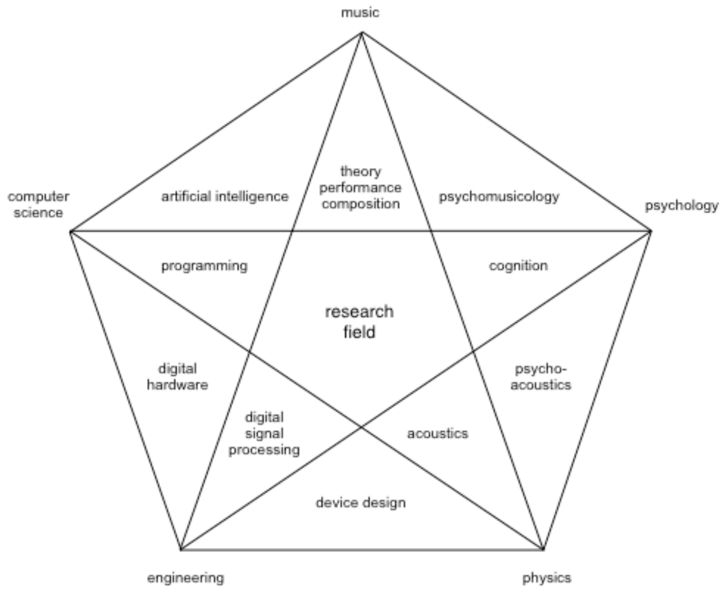

Figure 1:1 An illustration representing the cross-disciplinary nature of this thesis. Adapted from Moore (1990)

Time and latency. Considerations of time and latency are rarely of as much fundamental importance in computing as in the field of music technologies. A signal latency of a sample (typically 1/44100th of a second) can result in a glitch in the sound and a haptic latency above 20 milliseconds is noticeable to the musician performing the instrument (Roads 1996). Fast algorithms and effective controllers are therefore vital. Unlike much graphical or video editing software, realtime music instruments cannot wait while the program applies a filter or renders. To make things even more complex, digital musical instruments or sequencers tend to work as parallel streams (or channels) of musical events that have to happen at the correct time point, as opposed to the one-action-at-a-time process of off-line work.

Unnatural mappings. In digital instruments the control device and the sound source are arbitrarily related, unlike in acoustic instruments (Bongers 2000). The multimodal control mechanism used to generate the sound always affects the character and style of the performance. As an example, we might not hear if a piano in a song is a real piano or a synthesized piano, but we would definitively detect a synthesized trumpet if it was played on a

[4] See http://www.nime.org

keyboard. It might not necessarily be the sound itself that is different, but rather *what* is played with it. Playing a trumpet, with three fingers on the keys and with lips on the mouthpiece, is obviously a very different control mechanism from the situation where the synthesised trumpet is played using ten fingers on a keyboard and where the mouth "embouchure" is lost.

Programmed music theory. Here the term "programmed" is written with an etymologically-aware sense, as in *write forth* (*pro* + *graphein*), where it denotes the act of prescribing human actions. The music software has inscribed in it a music theory defined by the designer of the system. This has important effects on the user and the musical culture in which the technology is used. This area, in particular, is the focus of this thesis.

Considering the above, it is important to note that digital music systems are not merely simulations of acoustic instruments, studios or hardware technologies. They also include novel musical instrument designs, often with interfaces inspired by acoustic instruments. Here we also observe an important distinction between the acoustic and the digital. Whereas the musician practising the acoustic instrument incorporates knowledge of the instrument (it becomes a tacit knowledge), the digital musician's story is different: their education is primarily one of symbolic communication with the machine, of understanding user interface and interaction design conventions, and getting a technical grasp of the nature of digital audio. The mechanical properties of acoustic instruments are very different from the properties of digital algorithms and signal processing. Thus, the required knowledge of those two types of musicians is completely different. In short, the process shifts from an incorporated knowledge of the instrument's character to a knowledge residing in the symbolic realm, which includes the complex ways in which we design, understand and communicate with our machines.[5]

This research originates from the cultures of improvisation, electronic music and digital musical instruments (Nyman 1974; Bailey 1993; Nettl & Russell 1999; Bowers 2002; Jordà 2005). But more specifically, it arises out of a counter-culture in the sense that it is contextualised in the practices of musicians who build their own instruments, compose algorithmic music or make interactive installations facilitated by artists' skill in programming computers (Mansoux & de Valk 2008). Members of this culture have created open source audio programming languages to research acoustics and design systems of electronic music. It is a culture quite different from the world of commercial software where the "studio solution" packages that can do "everything you want" insist that it is "only your imagination that's the limit." This work thus originates from a viewpoint (which is a mixture of DIY street culture and

[5] A peculiar process of inversion can be witnessed here: the initial attraction with software like Cubase or ProTools was that these tools simulated the actual studio that musicians knew so well (where work practices from the physical realm could be transferred into the virtual realm). However, the situation is now that people learn the simulation before they ever have the experience of entering a real studio. The virtual now precedes the real; the map is known before the territory.

a broad interdisciplinary academic research field) that is more realistic about the positive and negative aspects of using computers in music.

In a recent research document by the *Sound and Music Computing* group funded by the European Commission (6th Framework Programme in the Future and Emerging Technologies), a definition of the field was provided as follows:

> Sound and Music Computing (SMC) research approaches the whole sound and music communication chain from a multidisciplinary point of view. By combining scientific, technological and artistic methodologies it aims at understanding, *modelling* and generating sound and music through computational approaches. (Serra et al. 2008: 9, my italics)

Modelling is succinctly defined as: "the representation of knowledge through algorithms and tools" (ibid). If there is one distinction above others that defines the digital musical instrument from their acoustic counterparts, it is this *act of modelling*: of defining a space, abstracting it, generalising, and encoding it into a technological artefact used for human expression.

We can now place this investigation in the context of other research. Many studies of digital musical instruments have been conducted from the perspective of performance, embodiment, ergonomics, and musical issues in general. Academic departments and conferences (e.g., SARC, IDMIL,[6] NIME, ICMC) and research centres (IRCAM, STEIM) have been established that partly focus on the human gesture, the gestural interface and the question of mapping gestures to sound. Researchers in this field engineer technologies that enable instrument building in the digital domain where sensitivity, expressiveness, low latency and multimodality are all important features. They also perform scientific studies of human perception, action, and engagement with the gestural device. This approach could be called a *phenomenological-engineering* study of digital musical instruments. However, there is another aspect of the digital instrument that has been less studied, namely the informational, compositional, theoretical, cultural, and epistemic nature of the technologies we work with. It involves the study of the computer as a symbolic system that is necessarily based on a strong theoretical understanding of acoustics and music theory. I will call this the *epistemological-engineering* approach, and that is the research area established in this thesis.

In other words, the motivation for this research can be defined as the attempt to elucidate, or make explicit, three highly interconnected areas:

- *Theoretical aspects in digital tools*: here the enquiry is into the prescribing tendencies of digital technologies, of how their design always implies a certain worldview and encapsulation of musical theory.

---

[6] Input Devices and Musical Interaction Laboratory - http://www.idmil.org/

- *Embodiment in digital musical instruments:* a focus on the nature of embodiment relationships with the digital instrument as one that is defined by its arbitrary mappings. A comparative account of the acoustic and the digital becomes helpful here; an account that can be useful in the *design* of new interfaces for musical expression.
- *Experimentation in the design of musical instruments as semiotic machines:* the ixiQuarks are a software environment developed as an experimentation and testbed for innovative ideas in interaction design in new musical instruments. Unlike most digital music software systems, it aims at live performance and the possibility of adapting the instruments in a live situation. The ixiQuarks are used in surveys and interviews that explore the topics described above as motivations a) and b).

1.3 Background and Context: ixi Software

The background and context of this research is ixi software. For nearly a decade, members of ixi audio (a small collective comprising Enrike Hurtado Mendieta, David Bausola, and myself) have developed software tools called ixi-software. The digital music system I focus on in this thesis is called ixiQuarks. It is *one of many* ixi applications; one that I started designing after having defined the research topics of this current research. The ixiQuarks build on the experience gained in the work with ixi software and conceptualise, encapsulate and amplify the discoveries and the inventions made though the last decade.

An essential part of our activities with ixi software, and a methodology relevant to this thesis, is the direct engagement with the community of musicians using the software. This accounts to an ethnographic work that supports many of the findings in this thesis. We have worked directly with musicans and sound artists, we maintain a website where people can download our applications for free, a mailing list, a net label, SVN repository for open source code, programming tutorials for various audio programming languages, and other general information for the sound artist. We regularly give workshops, introducing our technologies and ideas, assisting people to commence with audiovisual programming. Furthermore, we teach courses at university levels in related areas.

1.3.1 Introduction to ixi Software

The computer interface is more than a neutral instrument or a tool. It is a graphical or tangible manifestation of musical ideas and work processes. An interface is at once the aesthetic platform defining musical structures *and* the practical control-base for the underlying sound-engine. It can therefore be seen as a musical ideology. It defines possibilities but also the limitations of what can be composed or played – or phrased in a more technical language: it provides the affordances, and outlines the constraints of the artefact. Here I am primarily thinking of the graphical user interfaces of audio software, but the argument could be extended to audio programming languages as well: the objects or classes ready at hand in any given language define what can be expressed in that language [this topic is dealt with in chapter 7].

In the work with ixi, we have explored a basic idea of abstract graphical user interfaces, and created various applications, each representing a specific mode of interactivity. The interface (that which faces two systems, here the sound-engine and the human performer) is here seen as an important and influential factor in musical performance. It can evoke emotions, encourage direct responses, trigger ideas and open up unknown paths in a live situation. Musicians working with acoustic instruments know only too well how an instrument can have a unique personality, inspiring distinct playing styles depending on its character: suffice to point at the difference in playing a Fender Stratocaster or a Gibson Les Paul. The same applies to the digital instrument, arguably to a more pronounced degree.

The explorations we call *ixi* began as a thorough questioning and re-evaluation of the rather uncritical manner in which the commercial music software-houses build their interfaces on already established work processes known from the analogue studio or from musical traditions such as score writing. The two-dimensional computer screen and the mouse are good for many things, but not particularly effective for controlling a mixer with hundreds of knobs. We are not dismissing the ingenious work being done in some of the software houses, but we found there was a big gap that had not been investigated, where the computer (with the typical equipment people use, such as the screen, mouse, keyboard, sound card, etc.) is taken on its own premises, and its intrinsic nature explored. In the field of commercial music software there has been little investigation into the possibilities afforded by the computer's capabilities of memory, repetition, perfect timing, multitasking, graphical representation of sound or other processes, artificial intelligence, artificial life, pattern recognition and machine learning.

1.3.2 Related Work

The ixi project started in year 2000. At the time, few environments existed that allowed for fast creation of audiovisual systems. Languages like C/C++ and Java were the most efficient environments for the development of digital art, but they were hard to learn and the process of programming in them was slow. Other possibilities included Flash and Director, which have been used extensively by artists. Max/MSP was primarily for sound, although there existed a 3rd party video library for it. In these early days, the creative audiovisual work tended to be visualisation of sound, where amplitude or FFT (Fast Fourier Transform) data was translated into graphical representations (Techniques used in the visualisers of mp3 players, such as WinAmp and iTunes).

For ixi, inspirational work came from web design, game design and acoustic/electronic instruments. Stanza's soundtoys website[7] was an important playground for experimentation by its multiple participants, but as the name suggests, it was not aimed at instrument creation, but rather relishing in the playfulness of this new interactive modality between visuals and sound. Antoine Schmitt (2001) developed ingenious technologies for audio-visual communication in 2000 and released one of the first generative music CDs (Magnusson & Magnusson 2007). The interactive soundscapes called Shift Control[8] released on a CD-Rom by AudioRom were influential as well, although these were not tools for musical expression but rather aimed at explorative enjoyment in the manner of the soundtoys. In Japan, Toshio Iwai made impressive works, such as the Electroplankton games for the Nintendo, that were unknown to us until years later. Finally, starting in 2000, Golan Levin (2005a) made audiovisual performances where he would perform by drawing on stage, using spectacular visual tools to map graphics to sound. Levin had an artistic approach which means that his musical systems were always highly personal and tended to be exclusively written for his own expression. There was little generality in his work, rendering the software somewhat limiting for other musicians. The uniqueness of ixi, in this techno-cultural landscape, is the way it developed musical instruments that were general tools for personal expression. It focused on graphical objects controlling sound and introduced an innovative interaction model [explored in section 6.3.3] that people found inspiring and useful.

This was the situation until 2002, when the audio programming languages Pd and SuperCollider[9] were released as open source, Python brought impressive audiovisual libraries, and Processing[10] appeared as a coding environment developed by artists for other artists, thus minimising the computer science element. In 2001-2002 we presented ixi in various UK venues and festivals, such as London Dorkbots, and other London-based media arts related events which were abundant. In 2003 ixi was invited to the first organised NIME conference,[11] where we had the chance to introduce our work to other inventors and people in academia/research labs. Conferences such as NIME and the general DIY scene that appeared concurrently with the open source culture have resulted in a field of creativity where ixi has been a relevant player, as seen by the demand after the ixi workshops in various European countries and, of course, by the popularity of the software which gets around 10.000 downloads a month on average.

[7] http://www.soundtoys.net
[8] http://www.dor.co.uk/drift/audiorom.html
[9] http://www.puredata.org, http://supercollider.sourceforge.net
[10] http://www.processing.org
[11] It had taken place a year earlier as part of CHI; then with very few, but important participants, such as Cook (2001) who presented various inventions, Orio et al (2001) contextualising digital musical instruments within HCI, and Jordà (2001), who presented new music making paradigms.

1.3.3 The History of the ixi Technology

Initially the ixi instruments were standalone applications that enabled users to import their own samples into the instrument, and work with them in the way that the instrument proposes through its graphical interface. Gradually, a visual language evolved which was utilised in various ways in the design of new interfaces. For example, the location of objects on the vertical axis could control pitch, the horizontal might represent panning, size could indicate volume, and rotation some other parameter. Similarly, a movement or blinking of the object could indicate tempo or repetition. But instead of determining this language too rigidly, it was kept open and flexible. There were no definite rules for what a horizontal location should mean; but rather language games where meaning is defined by its context (Wittgenstein 1968).

The earliest versions of the ixi instruments had a sound engine and user interface that were unified in one software application. This allowed the production of a plethora of simple instruments, each of which would explore a new mode of interaction (rotating wheels, looping boxes, Markov systems represented as a plumbing structure with agents, predators and prey, spatialisation of sounds with moving microphones, etc.). However, following the release of Pure Data (Pd) and SuperCollider as open source software in 2001 and 2002, it seemed to be more sensible to use these environments as sound engines and use the newly invented OSC (Open Sound Control) protocol[12] to communicate control data from the ixi interfaces to the open source sound engines. This resulted in a change in the nature of our tools that could be described as a transition from an instrument to a controller, i.e., the application was not a "unity" anymore, but a front-end pattern-generator for an underlying sound-engine. Notably, this sound engine could be created in any environment that supported OSC, and we were thus faced with the complex, yet interesting, problem of the arbitrary mapping from human gesture to sound.

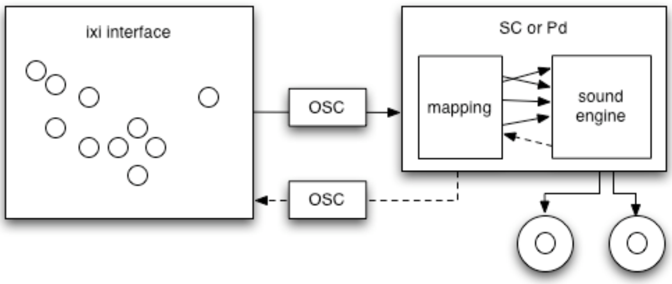

Figure 1:2 The ixi instrument as an interface to an audio programming language. The communication between the two elements takes place through OSC.

1.3.4 Instruments and Controllers

Using open source programming languages for music points the attention of the musician towards a field that ranges from the level of signal processing and the microstructure of a sound (where the focus is on the details of its texture), to the macrostructure of the music, i.e., how

[12] http://www.opensoundcontrol.org

musical events are organised over large time scales. The sound programming environments under discussion are open and flexible for almost all known synthesis and compositional techniques used today, but lacking has been the power to develop interesting and visually pleasing graphical user interfaces as part of the audio work.[13] In the design of ixi software, we have been interested in the aspect of musical performance in digital instruments, in both studio and live contexts. The aim of our research is therefore not slick design, but rather research and experimentation in the *mode of interaction* afforded by the instruments. The focus has been on how any given software application inheres a specific music theory or model, and how users adapt to or reject the inscriptions of the technology with which they engage.

As mentioned above, by splitting the pattern-generating graphical interface from the sound-engine, we switch from an instrument to a controller. The OSC protocol now allows us to create controllers that are open in their mapping structure, such that any action on the interface can be mapped to any parameter in the sound-engine. This mapping (Hunt & Kirk 2000; Hunt et al. 2002; Wanderley 2001, 2002; Bevilacqua 2005) can take place on all levels of musical composition, from the micro-level of synthesis to the macro-level of score arrangement. What we have explored in ixi, and contrary to most MIDI controllers, is the notion that there need not necessarily be any musical reference in the events that happen at the interface level of the controller. We make use of metaphors and metonyms derived from gaming, physics, artificial life, physical actions, etc., and these are not necessarily related in any way to the world of music. In fact, our design model encourages us not to use such visual metaphors, as they contain too much history and aesthetic implications in terms of tonality and structure. The beauty of OSC is precisely the fact that it is an open standard that does not impose any musicality or incorporate a certain musical tradition like MIDI does.[14]

1.3.5 Case Studies

This section illustrates some key points in the development of ixi software since 2001. On the ixi website there are currently over seventeen applications available for download, all created with a mixture of different technologies. Here I will make case studies of four applications that exemplify the changes ixi has undergone: SpinDrum, Connector, Picker and the ixiQuarks.

1.3.5.1 SpinDrum

SpinDrum is one of the earliest ixi applications, an experiment with alternative sequencing

---

[13] Initially, it was not the objective of Pure Data or Supercollider to support the creation of visuals (although there are interesting video and OpenGL libraries for Pure Data and QuartzComposer views for SuperCollider, they are not practical when building screen-based interfaces). We have therefore had to come up with different techniques to make this technology possible.

[14] Arguably MIDI is not very practical for certain musical cultures whose music is more microtonal or polyrhythmic.

technique, and also a parody of traditional sequencer software. Instead of linear boxes representing the steps, typically a 16-step sequencer, there are rotating wheels with 1 to 10 petals that can rotate at different speeds. Each wheel has a sound sample attached to it; when the petal reaches the top position (12 o'clock), it triggers the sample. The location of the wheel controls the pitch and the panning, and the size of the wheel represents the volume. Steve Reich's concept of "phasing" (Reich 2002) comes to mind: the software might consequently be described a phase sequencer. The musician is able to create any number of wheels with different number of petals, move the wheels around, adjust their speed, pause them, sample audio input and trigger those samples, save patterns, and move between stored settings, often resulting in interesting polyrhythms.

SpinDrum was made in year 2001 and serves as a good example of a specific type of digital instrument. As a tool providing certain affordances and constraints, it frames the user into a context for performing, particularly since it exists independently of other musical systems. It is not a plug-in for general digital audio workstations, nor is it a controller for other systems. Participants in one of our surveys compared it to a found object that is either picked up and explored one either gets tired or becomes obsessed with it, often using it as the main tool in live performances.

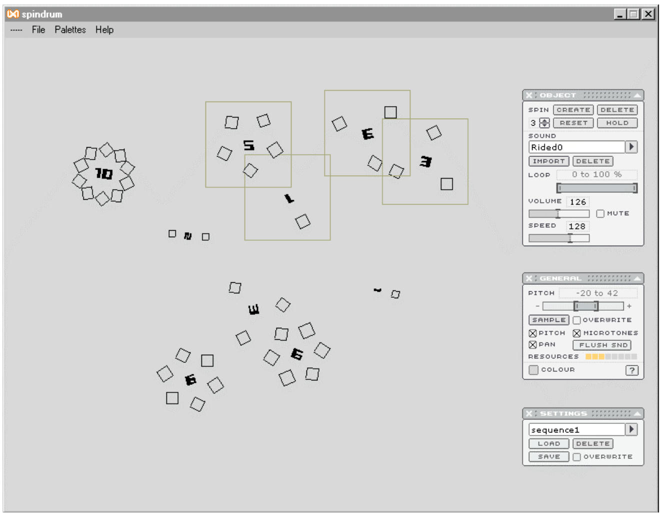

Figure 1:3 A screenshot of spinDrum. The wheels contain sounds that get triggered by their rotation. Pitch and panning is controlled by location and parameters such as speed can be adjusted.

1.3.5.2 Connector

Connector is an application that explores stochastic movement and change, with varied degrees of predictability. The interface consists of an environment in which one can build a plumbing system by the use of *connectors*, i.e., units with different number of outlets through which *actors* travel according to probability tables set by the user. Each connector has a probability graph that defines the possible movements of the actor which it contains. Unlike common Markov-models, it is not the actor that has knowledge of its movements, but rather the system itself that is a pattern of possible flow. The connectors "possess" sound samples and MIDI values, which are assigned by the user. These are triggered when one of the eight actors enters the connector according to a timing specified by the user. The actors can also move by stochastic timing, providing the user to sidestep the common enframing of musical software with regards to rigid time.

Connector was made in 2002. In this application we extended ixi software to become more than just instruments triggering samples by implementing MIDI capabilities. Connector can therefore communicate to any device (hardware or software) with an open MIDI channel. The use of MIDI highlighted many possibilities for us, where an interface (the actors and connectors) was mapped to parameters in any given sound engine. However, MIDI is arguably an outdated protocol (it was invented in a very different technological context of hardware synthesizers in 1982) and it became clear to us that it did not work well in software environments due to the limitations of 8 bit numbers, 16 channels, non-meaningful mapping (numbers and not names), slow transmission rate, and more besides.

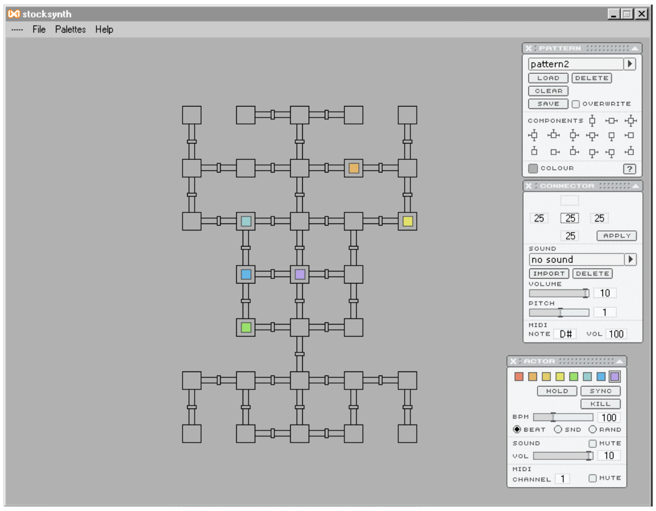

Figure 1:4 A screenshot of Connector. The system contains properties of sound and the coloured actors that run through the system are the performers.

1.3.5.3 Picker

Our first experiment with OSC was in 2003 with a controller called Picker. As the name indicates, the application picks up colour values from the interface and sends them as OSC messages to the sound-engine. The software is distributed with SuperCollider and Max/MSP sound engines. The application has bitmap, video and webcam layers whose ink (rendering modes), transparency and blending can be controlled. The creation of a complex graphical texture is therefore possible. The pickers can be placed in specific locations anywhere on the screen, move according to an automation algorithm, or be animated through the user recording a path. The user can affect the instrument with the webcam, thus creating a controller responsive to human gesture. The pickers send out colour values (RGB) and corresponding XY locations on the screen. This information is streamed out through OSC, and a matching mapping engine written in any language will retrieve these messages and translate them to sound.

Picker explored the possibilities of creating software controllers for unspecified sound-engines. It showed the positive aspects of limiting oneself compositionally to a certain instrument (or pattern generator) in order to explore other aspects of the composition (here the mapping and the sound).[15] This approach has been adopted in other projects as well, for example the IanniX OSC “poly-temporal meta-sequencer” by the French company The Kitchen.[16] The problem with this approach is the necessity of, and emphasis on, mapping. This is not a negative as such, but when it takes too much attention from the act of performing, the instrument becomes more an experimental exploration of compositional concepts than a solid musical instrument for performance. It becomes a compositional tool. A survey we conducted showed that users of ixi software found Picker too complex and intricate for their musical aims. The survey also showed that the people who had the required skills for such intricate mapping of interface and sound parameters through various mapping-techniques would typically want to build their own interfaces and not inherit a tool (which, per definition, also constitutes a manner of thinking) from another musician-programmer.

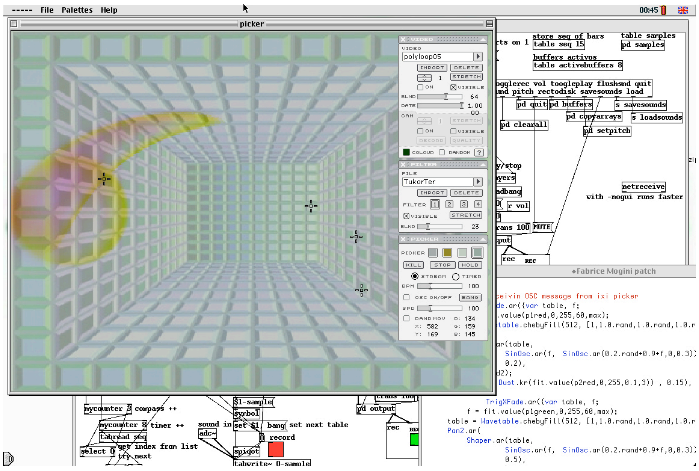

Figure 1:5 A screenshot of Picker. Behind the pattern generating interface we can glimpse both Pd and SuperCollider code windows that are receiving OSC messages.

1.3.5.4 ixiQuarks

With the background in creating technologies such as the above, having identified the research area of this thesis, I started exploring what kind of programming environment would serve best to implement the practical side of this research. The ixiQuarks came out of thorough testing of various audio programming languages and environments for graphical programming. The first premise for the choice of language was that it should be high-level enough to allow for rapid prototyping and be an inspiring work environment. As such, interpreted languages such as Python, Ruby or SuperCollider were preferable to compiled languages, such as C++ or Java; although processor speed would thereby be sacrificed, development time would be gained. After working with Python, Java and SuperCollider together for some time, I chose to use SuperCollider for both user interface and sound engine. Recent additions to SuperCollider (the UserView for mouse event-handling and the Pen class for drawing) made this work possible in the form that I intended.

[15] This idea of composition as mapping, or the emphasis on mapping as compositional technique, will be explored throughout this thesis.
[16] http://iannix.la-kitchen.fr/

Given the goals of the current research, working exclusively with SuperCollider was evidently the right decision; it allowed the creation of independent and limited instruments that can be easily *extended* through the extremely powerful programming language of SuperCollider. The ixiQuarks are the result of this work, where the focus has been on live improvisation with screen-based digital software instruments [see chapter 6]. The ixiQuarks effects and instruments can be loaded up from menus and played independently, or used as part of a complex signal routing system designed by the user. The instruments send out OSC

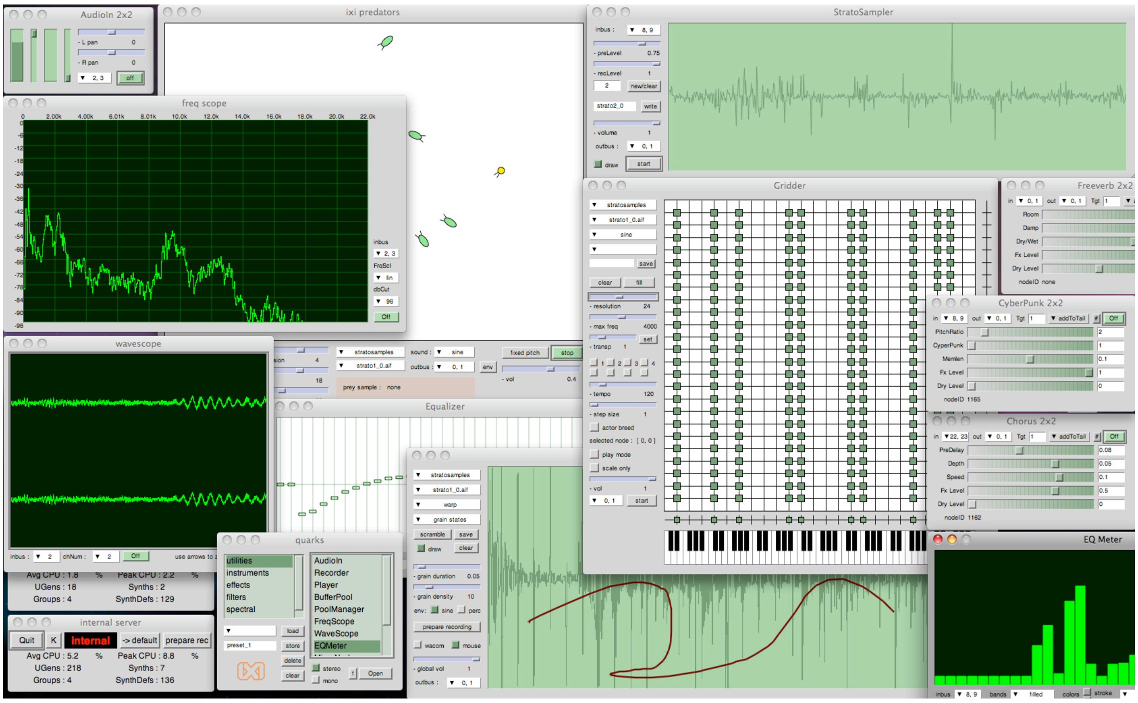

Figure 1:6 A screenshot of ixiQuarks. Visible are some of the tools and instruments that ixiQuarks consists of. Sound buffer pools, filters, effects, spectral manipulation, plus instruments of various kinds that provide novel interactive modes with sound objects.

messages to the SuperCollider sound-engine, but equally, any sound-engine could be used. The benefit of this default setup is that the user does not have to worry about mapping unless wishing to, enabling immediate exploration and play.

The raison d’être of the ixiQuarks instruments is not that of nice visuals or slick front end, but rather that of designing user interaction or workflow for a live performance. The ixiQuarks environment thus serves as a workshop for pattern generating applications, as the environment is infinitively extendable due to its modular nature. This focus on modular pattern generation and audio manipulation encapsulates much complexity under a simple front end. By hiding the engineered and compound sound-engine, and providing playful and simple interfaces with their unique expressiveness and limitations, the ixiQuarks allow the musician to forget about technology and concentrate on simply playing music with the ixiQuarks toolbox. However, and this I wish to emphasise, underneath the simple and constrained interface lies the practically infinite expressive power of SuperCollider itself as a programming language.

1.3.5.5 Conclusion

The preceding sections demonstrated the various technological and ideological changes ixi software has undergone, culminating in the development of the ixiQuarks as the testbed for this research. I find that ixi now provides the best of two worlds: the creative joy of playing with a constrained tool *and* the underlying infinite possibilities of SuperCollider as a sophisticated programming language for audio and music.[17] This current research thus explores the expressive space that appears when technologies become approachable enough for creative individuals to create their own systems. It studies how software can function as a semiotic machine that provides an epistemically mediated embodiment due to its dynamic and arbitrary mappings. Furthermore, the interest is in the conceptual, aesthetic, and music theoretical implications of designing musical instruments; particularly in digital musical instruments as they become epistemic to a degree hitherto unforeseen in musical tools.

1.4 Research Questions

Given the interdisciplinary nature of this research, it inevitably represents a field of activities that take place at the *borders* of the acoustic and the digital, the body and the machine. The methodologies are derived from the fields of music, philosophy, HCI (Human-computer interaction), and computer engineering. The research questions are coloured by this interdisciplinary stance, and the thesis consequently contributes to various fields of research. The main research questions are the following:

- Both the thesis of the extended mind and enactive theory demonstrate how the environment is used as part of cognitive processes. Our cognition can extend to external objects and artefacts. Provided that our thinking extends into the environment, how do our semiotic machines (such as software), affect our thinking? [chapter 3 and 4]
- Embodiment is normally seen as a precondition of virtuosity; if instrumental learning happens through a non-representational (non-symbolic) embodied process where instrumentalists incorporate musical knowledge through their body, what transpires in the learning of a computer-based tool that is essentially a symbolic machine? How do the learning processes of acoustic and digital instruments differ? Considering the epistemic dimension of digital tools, I ask if it is possible to become a virtuoso of digital musical instruments; in what sense we can talk about *virtual embodiment* in this

[17] Not unlike the way in which Apple has designed Aqua as the GUI system for OS X: underneath the slick and user-friendly front end, there is the powerful Unix system that can be controlled from the command line and programmed with shell scripts.

context? Do digital instruments provide a more pronounced epistemically mediated embodiment ? [chapters 3 and 7]

- Acoustic and digital instruments have significantly different material properties. Through a material epistemology, the differences in these material fundaments are studied and compared. What can digital systems afford that physical materials cannot? Which are the unique characteristics of the digital medium for musical expression? [chapters 4 and 6]
- The ixiQuarks form the practical basis of this research. The main questions here are: to what degree can a unique interaction model, originating from a personal practice, benefit other musicians? Can the logic of such an interaction model be polymorphic, i.e., easily transferrable between the diverse instruments? How do users engage with software of such a high degree of constraints that represent a concrete music theory, which might differ substantially from their own? [chapter 5]

These are the principal questions, but they merely represent the tip of a more intricate investigation that traverses multiple perspectives concerning the design of expressive digital technologies. Areas within the philosophy of technology, cognitive science, phenomenology, and human-machine interaction are the principal tools used in this enquiry. The main focus is on the affordances of digital technologies; how their immateriality makes them necessarily dependent on specific categorisations of the world, where human actions, thought, and musical composition are systematically defined in one engineered stroke.

1.5 Research Methods

This thesis operates within a strong interdisciplinary research field of which the author has been an active participant for over a decade. It builds on the collaborations, artist residencies, installations, interviews, concerts, and workshops ixi has done with various people and institutions over the years. This background presents a unique opportunity for research methods based on participatory and situated design. However, the research has not rigidly followed orthodox research methodologies, but rather used a mixture of ethnography, qualitative surveys, phenomenology, actor-network theory, and genealogy, in addition to various HCI methods; each applied when considered appropriate. Although the research methods were often highly integrated, a rationalised account of them can be presented as:

- As an active member of the computer music community, both in the fields of research and performance, this work has been strongly inspired by ethnographic methodologies. These have been used in the following activities:

  - Developing and performing with ixiQuarks, the software environment designed in the context of the current research. I have played with the software in my own musical practice (with two improvisation bands) and distributed it freely online on the ixi website, resulting in a user base of 37000 people, of which a considerable portion has registered for user-feedback and survey participation.
  - Teaching audio programming (synthesis, algorithmic composition, instrument design and interactive installations) in the Music Informatics course at the University of Sussex, Digital Music and Sound Arts at the University of Brighton, Sonic Arts at Middlesex University, and the Music Informatics course at Westminster University has given me valuable knowledge of the steps students go through to habituate themselves with specific compositional environments. In particular, it has been interesting to observe how musicians trained on acoustic instruments prime their mind to the work practices of digital music.
  - ixi workshops. Over the last seven years we (ixi) have hosted workshops around Europe, focusing on audiovisual programming and interactive techniques. In these workshops we have engaged and adapted to the different approaches of artists, designers, musicians, robot builders, mathematicians and architects to the problem of building creative or interactive systems.
- Actor-network theoretical and genealogically inspired analysis was performed on the origins and design of three different types of musical instruments: the acoustic, the electric/electronic, and the digital.
- Surveys. Four surveys were conducted during the research period: 1) The phenomenology of musical instruments – a survey with practitioners of acoustic and digital instruments. 2) ixiQuarks user survey – questioning the design ideology of the software. 3) Code as material – a survey with practitioners of audio programming. 4) Creating creative software – a report on a questionnaire sent out to the main designers of audio programming languages.
- Interviews. Focused interview and think-aloud sessions with novice ixiQuarks users were conducted. They were aimed at both general understanding of digital musical systems and ixiQuarks in particular.
- Finally, ixiQuarks have been used for teaching music technology and audio synthesis at Angela Ruskin University in Cambridge, Sonic Arts Research Centre in Belfast, Music Informatics at Westminster University, Digital Music and Sound Arts at the University of Brighton, and in the South Eastern Regional College. Surveys and informal interviews were conducted with these students, focusing on the experience of using ixiQuarks in their studies and creative work.

1.6 Contributions

This thesis is the result of a decade's work in designing, using and teaching the creation of digital musical systems. From this practical background of designing and performing with digital instruments on the one hand, and working with contemporary philosophy of technology on the other, a novel combination is derived that contributes to the field of music technology (and other related research areas) in the following ways:

- A philosophical elaboration of the theory of *epistemic tools*, i.e., how technology, as an extension of the cognitive system, programs users in ways which they are often unaware of. This emphasises the complex techno-cultural lineage of any technological artefact, articulates its nature as primarily a symbolic system or machinery, and exemplifies the way such technologies affect the thought of their users. This theory has its roots in the philosophy of technology, cognitive science, and empirical user-studies performed during the research. [chapter 4]
- An elaboration of the concept of *virtual embodiment* in the context of musical instruments; presented as the manner in which virtual (or screen-based) symbolic design enables the phenomenological experience of working with real objects. The metaphors used in manipulating these objects derive from real-world actions (such as creating, moving, rotating, scratching, touching, resizing, deleting, etc.), but also from video games design, web design and the properties of acoustic instruments. [chapters 5 and 7]
- An account of how learning and mastering acoustic and digital instruments differ in essential ways. Researchers of human-machine interfaces in general, *musical instrument design* in particular, will benefit from the phenomenological approach taken in this thesis of the body's relationship with technology, the concept of virtuosity, and the experiential differences of working with acoustic versus digital instruments. This argument has its roots in theories of enactivism and non-representational learning. The thesis illustrates how acoustic instruments require primarily embodied learning, whereas digital instruments entail an approach to mastery that is primarily symbolic. [chapters 3 and 4]
- A contribution to the field of software studies, cultural computing (or ethnocomputing), and software art, through an analysis of the ixiQuarks. This includes a demonstration of various design paradigms for audio programming languages and how their design differs according to the philosophical ideas that support them. Surveys and analysis of the creators and users of the various audio programming languages are presented that would be of interest to this research field. [chapter 4 and 5]

- To the field of HCI, and creative software in particular, this thesis provides a novel approach to graphical interface design, illustrating how design based on a trans-domain scope (where design influences are drawn from unrelated fields, such as gaming, new media design and physical actions) can yield greater affordances for creativity and innovative compositional methods. Then, a coherent interaction model that supports the design of the ixi software is also presented. This model serves as an example of how a semiotic language or design strategies can guide a group of people in creating a coherent set of tools. [chapter 6]
- More specifically still, this work contributes to the field of computer music with its explorations into alternative ways of designing musical pattern generators. It provides an exemplary framework where constrained interfaces are supported by a rich and expressive underlying programming language that extends the basic software environment. Studies have been performed of how users utilise the underlying power of the language. [chapter 6]
- This work adds to the field of music software design through the elucidation of four surveys, extensive user studies, interviews, and general questionnaires conducted as part of this research. These were described shortly in section 1.5 and further in the main chapters. [chapters 5, 6 and 7]
- This research contributes to the field of music technology with an extensive taxonomical analysis of digital musical systems, where taxonomies of instrumental functions, activities and actions are presented. A comparative genealogical and actor-network theoretically inspired study is made of three materially distinct instruments: the saxophone (acoustic), the MiniMoog (electronic), and the reacTable (digital). Furthermore, a dimension space for the analysis of the musical epistemic tool is presented, focussing on the informational aspects of digital instruments. It is hoped that this will be a useful tool for others in the analysis of digital creative systems. [chapter 6]
- Finally, to the field of musical performance, this work contributes novel music software intended for live improvisation and quick prototyping of musical instruments in the form of code. The unique focus of this software is to present a toolbox of instruments that are highly constrained, yet allowing for infinite extensions and adaptations through the underlying interpreted programming language in which they are built. [chapter 5]

1.7 Dissemination

Various sections of this thesis have been peer reviewed and published in conference proceedings, journal publications, and as book chapters.

Parts of chapter 3 and 4 are included in:

- Magnusson, Thor. (2009). "On Epistemic Tools: Acoustic and Digital Instruments as Cognitive Scaffoldings" in *Organised Sound*. vol 14 (2). Cambridge: Cambridge University Press.

Chapter 5 contains material from the following papers:

- Magnusson, Thor. (2009). "The Phenomenology of Musical Instruments: A Survey" in *eContact: Improv*. vol. 10 (4) Canadian Electroacoustic Community.
- Magnusson, Thor. (2007). "The Acoustic, the Digital and the Body: A Survey on Musical Instruments" in *The NIME 2007 Conference Proceedings*. New York: New York University.
- Magnusson, Thor. (2006). "Affordances and Constraints in Screen-Based Musical Instruments" in *The Proceedings of the NordiCHI Conference*, Oslo: Dataforeningen.

Chapter 6 inludes material from the following papers and a book chapter:

- Magnusson, Thor. (forthcoming). "Interface Investigations: The Thinking Behind Tools for Musical Thinking" in *The SuperCollider Book*. Massachusetts: MIT Press.
- Magnusson, Thor. (2007). "The ixiQuarks: Merging Code and GUI in One Creative Space" in *Immersed Music: The ICMC 2007 Conference Proceedings*. Copenhagen: Re:New.
- Magnusson, Thor. (2006). "Screen-Based Musical Instruments as Semiotic Machines" in *The NIME 2006 Conference Proceedings*. Paris: IRCAM.
- Magnusson, Thor. (2006). "The ixi Instruments as Semiotic Machines" in Multidimensionality: *The ICMC 2006 Conference Proceedings*, New Orleans: Tulane University.
- Magnusson, Thor. (2005). "ixi software: Open Controllers for Open Source Audio Software" in *Free Sound: The ICMC 2005 Conference Proceedings*. Barcelona: Pompeu Fabra University.
- Magnusson, Thor. (2005). "ixi software: The Interface as Instrument" in *The NIME 2005 Conference Proceedings*. Vancouver: University of British Columbia.

Chapter 7 contains large sections of the following book chapter:

- Magnusson, Thor. (2008). "Expression and Time: The Question of Strata and Time Management in Art Practices using Technology" in *The FLOSS + Art Book*, Poitiers: goto10.

Various sections in the thesis contain material from the following publications:

- Magnusson, Thor. (2008). "Working with Sound" in *The Digital Artists Handbook*. Lancaster: Folly.
- Magnusson, Thor. (2007). "Generative Schizotopia: SameSameButDifferent v02 – Iceland" in *Soundscape: The Journal of Acoustic Ecology*". Vol 7. Nr 1. Winter.
- Magnusson, Thor & Magnusson, Runar. (2007). "SameSameButDifferent v.02 – Iceland" in *YLEM Journal: Artists Using Science and Technology*. San Francisco: YLEM organization.
- Magnusson, Thor. (2005). "Intelligence Tools in Music" in *Artificial*. Copenhagen: Artificial.dk.

1.8 Content Description

This thesis is divided into two main parts; the first is more theoretical, general, and covers the fields of ontology, phenomenology, and epistemology of technological artefacts. The second part introduces the practical work of this research and contextualises it in the prism of the theoretical context outlined in the first part.

Part I

Chapter 2: Technology as a Condition of Thought – This chapter defines the concept of technology and introduces two main strands in the thinking of technology, i.e., technological determinism and social constructivism. It succinctly explores other approaches to these two schools of thought, such as interactionism, structuration and hermeneutics, but finally identifies and introduces two theories of technology that are useful in the context of this research: the actor-network theory, whose proponents include Latour, Callon and Law on the one hand, and Stiegler's theory of technology as epiphylogenesis on the other.

Chapter 3: Embodiment: The Body and Technology – After this exegesis of the philosophy of technology, as contextualised for the analysis of music technology, the effect digital technologies have on musical performance is explored. A general phenomenological account is given, introducing Don Ihde's phenomenology of technological instruments. This phenomenological delineation leads to the study of embodied learning, habituation, and of the body as a space for writing. This is done through the theoretical scope of two distinct, but related, theories in cognitive science: the theories of the extended mind and enactivism respectively. This chapter studies technological inscriptions in relation to embodiment.

Chapter 4: Epistemic Tools: Thinking Through Technology – Having established the complex and symbolic nature of digital technologies, it is relevant to analyse how systems of knowledge and representations are inscribed into technological artefacts. After a study of ethnocomputing and an excursion into externalist thinking in HCI, this chapter introduces the concept of *epistemic tools*. Epistemic tools are defined as technologies that influence our thinking due to intrinsic mechanisms that can potentially be defined as cognitive. Here digital technologies are seen as cultural machines (affecting culture through influence on their users) and the chapter ends by discussing the relevance of a new research field called software studies.

Part II

Chapter 5: A Theory of Digital Musical Instruments – A comprehensive analysis of digital instruments is provided in this chapter. Subsequent to the introduction of an extensive survey on the phenomenology of musical instruments, a comparison in the design, use and reception of acoustic, electronic and digital instruments is presented. Three selected instruments (the saxophone, the MiniMoog and the reacTable) are analysed and explored through actor-network theoretical (Latour) and genealogically (Foucault & Nietzsche) inspired analysis. This analysis paves the ground for a taxonomical analysis of digital musical instruments with emphasis on bodily gestures or expressivity. Finally, a dimension space for the analysis of digital musical instruments *as epistemic tools* is presented.

Chapter 6: ixiQuarks: Theory and Practice – This chapter introduces the ixiQuarks, a screen-based digital musical instrument aimed for live performance and improvisation. Much of the theoretical and empirical body of this thesis is based in the ixiQuarks, as it served as testbed for experimentation in design and user testing. The ixiQuarks are presented as epistemic tools, and an introduction to the thought process of designing a musical instrument is provided where the relationship between technological affordances and creativity is emphasised. Finally, surveys and user studies on the ixiQuarks are discussed and interpreted in the context of the theoretical issues described in part I.

Chapter 7: Designing Digital Tools: Instruments as Systems – After the investigations in the preceding chapters, it is now possible to revisit and reinforce the concepts of epistemic tools and virtual embodiment. Digital musical instruments are seen here as epistemic or semiotic machines onto which we delegate parts of our cognitive process, but, as opposed to provisions in Andy Clark’s work or in enactivism, the designer’s role in these machines for thinking is emphasised. External technologies are recognised as cognitive engines that influence and inform the thinking of their users. The above perspectives are then succinctly investigated in two surveys, the former with *users* of audio programming languages, and the latter with their *designers*.

Chapter 8: Conclusion – This thesis traverses a wide interdisciplinary field; in the conclusion the central questions and themes are revisited. Future research is proposed and the thesis ends with a note on the cultural responsibility of the systems designer, i.e., an encouragement to programmers and designers to engage with, and question, their philosophical, music theoretical, and aesthetic stance when inventing new musical systems.

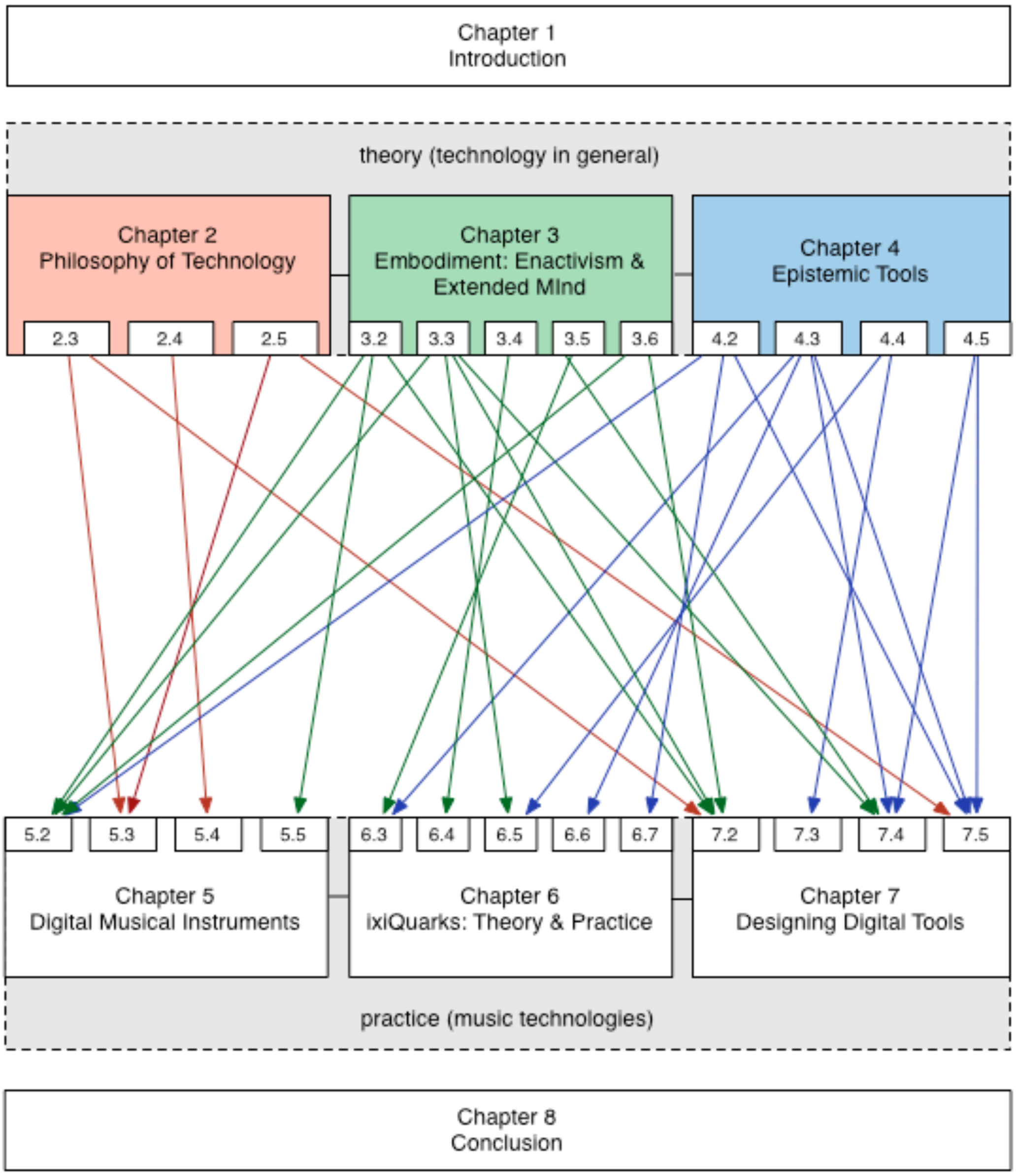

Figure 1:7 A graphical overview of how the theory in Part I relates to sections in Part II

Chapter 2

Technology as a Condition of Thought

This chapter defines the concept of technology and introduces two main strands in the thinking of technology, i.e., technological determinism and social constructivism. It succinctly explores other approaches to these two schools of thought, such as interactionism, structuration and hermeneutics, but finally identifies and introduces two theories of technology that are useful in the context of this research: the actor-network theory, whose proponents include Latour, Callon and Law on the one hand, and Stiegler's theory of technology as epiphylogenesis on the other.

2.1 Introduction

Our writing tools are also working on our thoughts. (Nietzsche in Kittler 1999: 210)

To think about *music technology* entails thinking about technology in general. What is technology and how does it affect human activities? In order to establish a solid understanding of technology as an active agency in human culture, we need to study the origins, use and effects of technology – and its prevailing discourses.

The branch of philosophy called *philosophy of technology* is a recent research field. Karl Marx was arguably the first thinker to take seriously the role of technology in our culture and the power it has over our work, on thought, and social organisation. The philosophy of technology is often associated with the turn to "applied philosophy" (Bradie 1983) that became a popular research activity in the middle of the 20^{th} century. It can be argued that it was only when modern technology became grave and sombre enough (with nuclear weapons, failing power plants, new communication technologies, changing social habits, environmental disasters, genetic manipulation of human DNA, etc.) that the thinking about technology got the momentum it deserves as an independent branch of philosophy.

The question concerning technology, its relationship to culture in general and the human subject in particular, has been approached from various angles: from the sociologist Pierre Bourdieu's reluctance to define technology as an object of study to the philosopher Bernard Stiegler's definition of technics as the constitution of the human. It is not necessarily philosophy, but sociology, cultural studies, biology and psychology that have dealt with the questions technology poses to the most satisfactory extent. This chapter on the philosophy of technology will therefore also survey various extra-philosophical traditions in order to gain a proper overview of the field.

A series of questions immediately present themselves querying why the philosophy of technology is such a young research field. How could philosophers ignore this highly important factor of human life to such a degree that we can actually talk about a deliberate suppression of the subject? The reasons extend far back into antiquity starting with Parminides' (520 BCE) focus on eternal being and pure rationality as opposed to the phenomenal situation of humans and their senses, thus introducing the dualistic transcendental ideal. Plato continues this line of thinking further with his focus on the absolute ideas, defining craftsmanship and skill as secondary activities whose nature are simulation and not truth. For Plato, the function of technics (Greek: *techne*) can be a dangerous affair, in particular when it becomes mnemotechnics such as (the then newly invented) alphabetical writing:

> If men learn [writing], it will implant forgetfulness in their souls; they will cease to exercise memory because they rely on that which is written, calling things to remembrance no longer from within themselves, but by means of external marks. What you have discovered is a recipe not for memory, but for reminder. And it is no true wisdom that you have to offer your disciples, but only its semblance, for by telling them of many things without teaching them you will make them seem to know much, while for the most part they know nothing, and as men filled, not with wisdom, but with the conceit of wisdom, they will be a burden to their fellows. (*Phaedrus* - Hamilton & Cairns 1989: 520)

At the very initiation of Western philosophy we encounter a view of technology (in this case writing) as a supplement, or a support device alien or external to human essence. For Plato, technics is an unnecessary and dangerous device, an external simulation of the true inner, and he treats in the spirit of the dualistic project of separating the inside from the outside (the soul from the world, pure ideas from the simulacra, or *physis* from *techne*). This stance towards the technology of writing is not merely a millennia old archaeo-Luddism. An echo of Plato's criticism of the written word can be found in the hermeneutics of Gadamer (1989: 368), who believes strongly in the power of the spoken word over the written. This "logocentric" belief has been strongly attacked in a thorough post-structuralist critique by Derrida in his spectacular reading of Plato's *Phaedrus* (Derrida 1981).[18] In a more recent work by Bernard Stiegler (1998), Plato is portrayed as stuck in an aporia where he creates the dualism of the inside and the outside by forming a transductive relationship between the terms (they constitute each other). Stiegler points out that there can be no interior without the exterior, that the idea of the

---

[18] Which resulted in the famous 1981 encounter in Paris between Gadamer and Derrida, documented in Michelfelder & Palmer (1989). What is at stake here are two differing views of conversation and language. These views could be roughly (thus unjustly) portrayed as Gadamer having an idea of language as essentially meaningful and thus the possibility of conveying truth and Derrida rejecting this view, claiming that language is merely an indeterminate play of floating signifiers.

inner comes from the realisation that we are always already exteriorised, differentiated and prosthetic. Stiegler is here referring to Derrida's concept of *différance* and the original supplement here transposed as viewing *technology* as "originary technicity," as something that constitutes the human. Technology is simply not something that can be defined as the object of human culture and history where the human is the subject. In this view, the relationship between the "who" and the "what" in terms of humans and technology is not uncomplicated:

> The natural, originary body does not exist: technology has not simply added itself, from the outside or after the fact, as a foreign body. Certainly, this foreign or dangerous supplement is 'originarily' at work and in place in the supposedly ideal interiority of the 'body and soul'. (Derrida 1993: 15)

Derrida points out that although technology is a set of artefactual, corporeal and semiotic realities[19] that can be treated as a discursive reality originating in the inventions of modernity, it should rather be seen as constitutive of modernity than its result. Technology is not outside discourse, it is rather by force of technology that we can "think, signify, make sense and represent who we are" (Mackenzie 2002: 5).

It is no coincidence that when human technology took a mutational jump into the future with sciences such as cybernetics and digital computing in the 1940s and 1950s, that philosophy of technology started to form itself as a unique discipline through the work of Martin Heiddegger, Karl Jaspers, Dilthey, Ortega y Gasset, Lewis Mumford, Gilbert Simondon, Bertrand Gille and Jacques Ellul. However, the field is still fumbling for foundations and often mixes with history or sociology, especially history of technology and science, and sociology of science. In sociology there is a school called *social constructivism* that has been an avid critic of philosophy of technology, or perhaps the lack of it:

> Philosophers tend to posit over-idealized distinctions, such as that science is about the discovery of truth whereas technology is about the application of truth. Indeed, the literature on the philosophy of technology is rather disappointing. We prefer to suspend judgement on it until philosophers propose more realistic models of both science and technology. (Pinch & Bijker 1987: 19)

The attempt to put forth a "realistic model of both science and technology" is arguably an impossible task. Don Ihde, a genuine philosopher of technology, proposes an ontological shift inverting the equation which implies that technology is "applied science" to one where science is seen as the tool of technology. Instead of the Platonic ideal where science deals with the eternal truths of immaterial forms that take part in a coherent whole, we now get a view where science is a result of a particular technological culture conditioned by its historical praxis.

[19] A semiotic reality much visible in the current techno-culture of West-Coast hype, global marketing and Nasdaq share indexes

Therefore, technology is not "possible as a pure neutrality; it is a 'choice' of a possible way of being in the world. Thus ultimately, science is not a form of contemplation of the eternal forms, but is the arrangement of human social, political and individual action engaged with the material world" (Ihde 1979: xxvi). Ihde here points out the practical or materialistic foundations of philosophy of technology, and therefore the difficulty some sectors of philosophy have had in accepting it as an actual branch of study within philosophy:

> One of the features of philosophy of technology that differentiates it from other styles of philosophy is its necessary sensitivity to the concrete, to materiality. The traditions of philosophy that are predisposed to precisely this concreteness are the praxis philosophies that include pragmatism, some strands of Marxism and neo-Marxism, and the phenomenology and hermeneutic traditions. It is not accidental that there is very little "analytic" philosophy of technology, and neither is it accidental that philosophy of technology associates with the praxis directions. (Ihde 2003: 1)

In this chapter we will examine different views of the relationship between humans and technology. Are we determined by technology or do we shape technology to our needs? Is technology a neutral, external supplementation of our activities or does it have a stronger role in constraining how we perceive the world and act in it? This enquiry into the nature of technology is needed for a proper understanding of the relationship between musicians and their technologies. This enquiry primes the understanding of musical software as *epistemic tools,* whose role in our musical expression is not as transparent or given as it might otherwise appear.

2.2 Defining Technology: Not a Single Phenomenon but Plurality of Forms

Necessity is often not the mother of invention. In many cases, it surely has been just the opposite, and invention has been the mother of necessity. When humans possess a tool, they excel at finding new uses for it. The tool often exists before the problem to be solved. Latent in every tool are unforeseen transformations. (Nye 2006: 2)

One of the first tasks of a work dealing with the philosophy of technology (and music technology in particular) should be to define the term "technology." This is a difficult task as technology is used in (and often is the condition of) an infinite number of contexts of human life. Today we often associate technology with digital "gadgets," cyberspace, fast transportation, robotics and space-age interplanetary explorations. In certain discourses it has become synonymous with "digital," but this view is more likely to be shaped by strong commercial interests in the techno-economy that have infiltrated and taken possession of the media-ecosystem.

2.2.1 Technology: A Phenomenon Resisting Definitions

A quick excursion into the etymology of the term "technology" shows us that *techne* is used by the Greeks for the act of producing things, of creating something out of another material (Heidegger 1977). For Plato and Aristotle, *techne* is the capacity for action, founded in specialist knowledge. Aristotle defines it as the business of bringing something into existence out of other materials, thus having its "origin in the maker and not in the thing made" (*Nicomachean Ethics* book 6 chapter 4). However, the word "technology" is not used frequently by the early Greeks. Aristotle only uses the word four times in his *Rhetoric*, but then with reference to the "craft of the word" (*techne* and *logos*). In his celebration of rhetorics, Aristotle parts ways with his teacher Plato who criticises rhetorics as an art (*techne*) that focuses on means and is divorced from reason. Cicero (106-43 BCE), a writer who celebrates the way humans transform their environment as it if was their "second nature," transliterates the Greek term into Latin only once (*Letters to Atticus* 4.16). We have to wait until the 16th century before we see the word used in Latin again, then used by the French rhetorician Peter Ramus (1515-1572). The word enters English in the 17th century and the same development, albeit a few decades later, takes place in France and Germany. Mitcham (1994) points out that in the latter half of the 17th century, the Western ontology of matter is completely transformed. This transformation happens under the influence of Galileo, Descartes and Newton who revolutionised the way we understand matter, from seeing it as a lifelike and spirited material to a view that presents it as wholly inert and devoid of spirit. This ontological shift creates the possibility of *technology* as we understand it:

> [I]t was through the modern hiatus that human beings began to imagine the possibility of a *logos* of *techne.* Thus it began to make sense to use a term originally applied to the study of the manipulation of words, then to the organization of systems of words, to name the study of the manipulation of nature. (Mitcham 1994: 133)

For the 18th century German philosopher Christan Wolff, technology is "the science of the arts and of the works of art"[20] (ibid: 131). It is not until Jacob Bigelow's *Elements in Technology* (1831) that the word becomes popular and takes on its familiar use. Bigelow's text is the first work with the word *technology* in its title. He claims that he adopted a word "found in some of the older dictionaries" in order to refer to "the principles, processes, and nomenclatures of the more conspicuous arts, particularly those which involve applications of science" (Mitcham 1994: 131). As an example of cultural inertia, the word had not gained enough buzz yet and Bigelow's work was later published under another title: *Useful Arts considered in Connection with the Applications of Science*.

[20] "Art" here of course understood as techne, as the craft of making things. In the old Greek texts, techne is normally translated as "art."

By the end of the second World War, technology had become a strong cultural parameter. In 1954 Heidegger firmly asserts that “the essence of technology is by no means anything technological” (Heidegger 1977: 4). The same year, Jacques Ellul, in his *The Technological Society,* defines technology as a combination of the object, the technique and the social context of its use: “The machine represents only a small part of *technique*...we could say not only that the machine is the result of a certain *technique*, but also that its social value and economic applications are made possible by other technical advances” (Ellul 1964: 4). In *The Machine in the Garden*, Leo Marx (1964) illustrates how the word *technology* became a powerful denominator of new and hitherto unthinkable practices. In the early 20th century, the capitalist large-scale industry had grown remarkably fast and suddenly there was a “semantic void” between the material reality and the lack of contemporary terms to describe the changing world of human practices. *Technology* became the word of force. Ruth Oldenziel describes how the word began to carry a certain power to frame social realities. She claims that the uncritical use of the term creates a constant problem when people fail to realise the distinction of technology as an “analytical tool” and as a “lived experience” (Oldenziel 2006).

Technology is often defined as “applied science.” This definition is losing its currency due to recent research showing how many of the greatest inventions in history were arrived at without any scientific understanding of the materials used. Examples include the Wright brothers and aviation, Edison and electricity, or Newcomen’s steam engine (Nye 2006: 11). This fact led Cyril Stanley Smith, a metallurgist at MIT, who participated in the building of the first atomic bombs, to declare that “[t]echnology is more closely related to art than to science – not only materially, because art must somehow involve the selection and manipulation of matter, but conceptually as well, because the technologist, like the artist, must work with unanalysable complexities” (Smith quoted in Nye 2006: 10). From a phenomenological perspective, technology is not merely understood by the mind, but more importantly *experienced* through the body. There is a culture around every tool and within this culture we see the transmission and transferring of skills through imitation and guided acquisition. Technologies are not simply the tools used, but also the skills needed to use them and it is precisely through this embodied interaction with technologies that many important scientific discoveries have been made (Baird 2004).

Bernard Stiegler defines technology as: “the discourse describing and explaining the evolution of specialised procedures and techniques, arts and trades – either the discourse of certain types of procedures and techniques, or that of the totality of techniques inasmuch as they form as system: technology is in this case the discourse of the evolution of that system” (Stiegler 1998: 94). Stiegler encourages us to use the word technology as we use the words “sociology” or “psychology.” From a more sociological context, Michel Foucault, a philosopher concerned with the technologies of power and the self, divided technology into four forms: 1) technologies of production (producing, transforming and manipulating things); 2) technologies of sign systems (allowing us to use signs, meanings, symbols and significations); 3) technologies of power (domination, objectification of the subject); 4) technologies of the self (operations on the body and mind, thinking, conduct, a way of being)[21] (Foucault 1988). We get a wider picture where technologies are understood as the fundament that supports our being-in-the-world; as strategies of living.

As demonstrated, the endeavour of defining the term "technology" neutrally and outside of a specific context is impossible. Leading theorists in the field have stated that it is "unnecessary to devote much effort to working out precise definitions" (Bijker et al. 1987), whilst others have pointed out how the wide area of the field (and its different definitions) renders it all but impossible to become a specialist in all its complexity (Winner 1993: 362). We could choose to develop general perspectives and observations from the various sectors of technological practices or we could dive into a specific sector and become an expert, but either way is a reduction of technology into the hermeneutic stance of the theorist: there will never be an objective view of technology.

2.2.2 The First Questioning of Technology

Perhaps the most influential work in the canon of philosophy of technology is Martin Heidegger's 1954 essay *The Question Concerning Technology*. In this essay, Heidegger attempts to outline the essence of modern technology and understand the conditions in which we can establish a "proper relationship" with it. Contrary to how many people read Heidegger, he is not a technological pessimist. His intention is to establish a free relationship between the human being and the essence of technology, an essence which is nothing technological in itself, but rather a result of how western culture has metaphysically defined nature and the human subject all the way back to the early Greek thinkers. For Heidegger "modern technology" has three interrelated meanings: firstly the new production processes of industrialism, techniques, devices and systems; secondly, modernity's secular world view and rationalistic, scientific, commercialist, utilitarian, and anthropocentric way of being-in-the-world; thirdly, the modern, technological mode of understanding or revealing the world epitomised in the way the rational mind or industrialism in general relate to the world.

Heidegger refrains from simply subscribing to either the instrumental ("technology is a means to an end") or the anthropological ("technology is a human activity") definitions of technology. Rather, his aim is to show the ontological set of conditions necessary for this

[21] Various authors have pointed at how music has now become a typical example of the technologies of the self. For example, DeNora (2000) and Bull (2000) point to how people use music and mobile music players as technologies for changing their mood; creating their own custom soundtrack for life; communicating through exchanging music, using the playlist to affirm their identity, and demarcate their private space in public.

particular technological worldview. He believes that the human mind has, throughout history, undergone some important conceptual and ontological movements that condition its thinking. These are not worldviews in themselves but rather the rudiments necessary for the emergence of a particular worldview. The worldview of the technological human is to consider the entire world as something that exists primarily as a source for consumption. The whole world thus becomes a *standing reserve* (*Bestand*). In this system, for something "to be" means for it to be raw material for the self-enchanting technological system. This essence of modern technology shows itself in what Heidegger calls *enframing* (*Gestell*). "[Enframing] is nothing technological, nothing of the order of the machine. It is the way in which the real reveals itself as standing reserve" (Heidegger 1977: 23). Enframing is the manner in which we see the world, and this essence of technology is the way the world now discloses itself. Technology as enframing becomes the condition for modern science. It becomes the origin of the possibility of the scientific world view that sees the world as a standing reserve. Technology thus precedes science:

> Because the essence of modern technology lies in enframing, modern technology must employ exact physical science. Through its so doing the deceptive illusion arises that modern technology is applied physical science. This illusion can maintain itself only so long as neither the essential origin of modern science nor indeed the essence of modern technology is adequately found out through thorough questioning. (ibid: 23)

For Heidegger, if technology is the condition of science, of our being-in-the-world (as we will see in the next chapter), it also monopolises the way we experience the world and excludes other ways, such as the religious, the aesthetic, or the ethical view. The technological enframing leads to crisis in values, and instead of an authentic world of beings, we get a desacrilised world of standing reserve, which exists only for and because of us; the humans whose mythologies tell them that the world is there for them to use.

Heidegger traces this productionistic metaphysics not only back to the industrial revolution of the 18th century, but to the Greeks. The history of the West is the story of how the Greek metaphysics transformed into modern technology. What we need, according to Heidegger, is a proper relationship with Being. Only then can humanity enter into a meaningful relation with the essential nature of technology. The remedy is a phenomenological understanding of the nature of technology but more importantly the use of art as an alternative way of revealing. Through etymological analysis, he shows that the Greek word *techné* meant not only technology, but also art. Both art and craft production are modes of disclosing, what the Greeks also called *poiésis*.

> There was a time when it was not technology alone that bore the name *techné*.
> Once that revealing that brings forth truth into the splendour of radiant appearing

> also was called *techné*. Once there was a time when the bringing-forth of the true into the beautiful was called *techné*. And the *poiésis* of the fine arts also was called *techné*. (Heidegger 1977: 34)

Heidegger finds in art a world-view that is not enslaved by the essence of technology. Just as with technology, art is *techne*, but it embraces a different way of revealing Being; a way that is not that of enframing or ordering reality in a standing-reserve. Thus, art has the potential to disclose the being of entities, to reveal things as things, in their individuality, their depth, their being – not as standing reserve, stock of energy or commodities. Art can free us from the total enslavement to technology by opening up another sensibility, and by teaching us to use technical objects appropriately. He ends his essay on this prophecy of art as the "saving power" from the danger of enframing.

Heidegger was writing in the 1950s, an age where technology and art were considerably different from today. He was occupied by modern industrial technology such as nuclear or hydro-electric energy and not focussing on the craftsman's tools as he did three decades earlier in *Being and Time.* Later work in the philosophy of technology refrains from making these distinctions between modern and pre-modern technologies, and, as we will see later, it hardly makes sense to separate technology and art in the contemporary situation. However, it is important to be aware of and acknowledge the fundamental influence Heidegger's writings have had on all succeeding thinking about technology.

2.2.3 Technoculture: Technology as a Condition of Culture

Traditionally, technology is defined as the *other* of something else. It has been defined against nature, culture or the human subject: technology is crafted by humans; it is a static and cold means to an end, and it does not reproduce. It is a dead mechanism, external to the human function. This view of technology has been challenged recently by philosophers such as Bernard Stiegler, who argue that technology constitutes the human as its external memory. Furthermore, modern biotechnology is erasing the gap between technology and life.[22] Technology is often used as the "other" of human culture and society. Here it is used as means to impose "rationality" upon social problems and it is seen as something rational, objective and sensible as opposed to social problems or political ideologies (Williams 2000). However, as we will see later, technology and its uses have been analysed as being innately political (Winner 1977; Fuller 2003), which forces us to be critical and reflective regarding the ethical implications of new technology (Whitby 2008). In fact, all social institutions are conditioned by the technologies they use and have been so through the ages. Society cannot be conceived without reference to its technological mediation (Latour 1994). Arguably, it is precisely through

---

[22] Engineered life can even be patented: http://www.wipo.int/patent-law/en/developments/ biotechnology.html

technology that human society is made possible by virtue of written law, thus creating the conditions for cities in the centuries 700-500 BC. (Stiegler 2003a).

We find ourselves in an ever more complex society that is run by technological means on all levels. As Foucault points out, the institutions of mental health or imprisonment are technological tools used to maintain a certain power structure within society (Foucault 1977a). Technology is a form of power and, as many of the thinkers of the Frankfurt school argued, it materialises hidden ideology. When society becomes this complex, we often talk about *technocracy.* This is a term defined by Theodore Roszak as being the "society in which those who govern justify themselves by appeal to technical experts, who, in turn, justify themselves by appeal to scientific forms of knowledge." (Roszak 1969: 7). Feenberg questions to what extent the technocratic administration is actually scientific and shows that it is rather in *rhetoric* than practice that societies are technocratic (Feenberg 1999: 4). Technology has become the condition of cultural products and creativity. Not only do we think, sketch, communicate, produce, distribute and consume through technological products, but we are also defined by the limits that technology provides. The relation between humans and the world is a mediated relationship of a symbiotic nature, where the distinction between subject and object dissolves (Ihde 1990: 73). The affordances or potentialities of technological products have become increasingly strong factors in our social life. It is not only what technology provides that defines us, but also our adaptations, avoidances, subversions or resistance to various technological forms.[23]

Bertrand Gille (1978: 70) detects two traces in modern technoculture that can be seen as dynamic arcs through which technology advances: a) scientific progress – invention – innovation and b) invention – innovation – growth. In the first arc we see that scientific progress is driven by demands of the technological industry and its applications in society (for Gille, innovation is invention applied and incorporated into society). Now the economy, with its demands for eternal growth, demands innovation as an essential function of its movement, as if it sucks (like in a vacuum) invention into the sphere of society through the filters of innovation. A quick mock-up of the music industry might provide a good example: the music industry continually demands new acts to market. These new acts have to show a certain degree of innovation or originality in their music for it to become relatively interesting. This originality, as we will see later, will often occur in the specific context of working with musical tools. The software houses respond to these demands with ever more sophisticated music software, drawing from latest scientific research in acoustics, digital signal processing and human-computer interaction.

[23] This last point is explored in the work of Ross (1991).

R.L. Rutsky, in his book *High Techné*, analyses how the culture of postmodernity has ended up becoming one of techno-culture. As our habitual world becomes increasingly liable to technological digital reproduction, all distinctions between technology and culture begin to vanish. When all our communication and information is transmitted and stored by means of digital media, technology itself is increasingly seen in terms of cultural data that make up what might be called the techno-cultural memory, a phenomenon that has become too dense and complex to be thought or represented as a whole. Technology is now perceived to be beyond human instrumentality and control; not subjected to the modernist instrumental rationality anymore, technology takes on a much more unpredictable "techno-logic" of its own. Technology is no longer chained to the notions of functionality or instrumentality, but becomes related to the notion of aesthetics. High-tech (or its subversion) becomes a style, and in turn, we talk about technology in terms of being "state-of-the-art." Not only has our conception of technology undergone a mutation, but technology itself has come to be seen as a mutational process or logic which is beyond our conceptual understanding (Rutsky 1999). Perhaps we have come to live in a reality Heidegger sees as one possible future, where "throughout everything technological, the essence of technology may come to presence in the coming-to-pass of truth" (Heidegger 1977: 35)[24] i.e., we have become so framed in technology that it has become the *physis* of our environment, embodying our practices, ideas and values.

2.3 Theories of Technology and Culture

We shape our buildings, thereafter they shape us. (Churchill 1960)

The two most apparent strands of thought in the history and philosophy of technology are arguably technological determinism and social constructivism (or technological constructivism). The former is often attributed to Karl Marx, who writes in *The Poverty of Philosophy*, "The hand mill gives you society with the feudal lord; the steam mill society with the industrial capitalist" (Marx 1971: 109). Of course, we acknowledge that Marx is no technological determinist – the first volume of the *Capital* (on how social relations shape technology) proves otherwise. Marx is interested in the social and technological factors that characterise historical periods, and his view is that technology and society have a reciprocal relationship. Indeed, the bipartition of authors into the two camps (determinist/constructivist) is a crude act of defining general trends, almost a caricature of what is to be found in the texts. Nevertheless, it can serve as a useful distinction, since people can normally be situated on this continuous scale between

[24] Heidegger provides this at the end of his essay on technology as a less ideal alternative to the possitiblity of art being a way to break out of the enframing of the technological mindset [see section 2.2.2].

the opposite extremes. Often the same thinker can be found swinging like a pendulum from one side to the other, even in the same text. I therefore will not present the following strands as rigid schools of thought, but rather as different perspectives on technology as it appears in its multiplicity and omnipresence.

2.3.1 Technological Determinism

Technological determinism promotes the idea that technological innovation is the main cause of changes in society. Karl Marx, although, as mentioned above, is no strict technological determinist, argues that "the mode of production of material life determines the general character of the social, political, and spiritual process of life" (Marx 1964: 64). Here, technology is seen as "the prime mover of industrialization and social change" (Burns 1969: 35). People therefore have to adapt to these new changes and arrange their work and leisure according to new technological situations. Technological determinism states that we find ourselves in a situation where we are incapable of controlling technology, despite our efforts. More importantly, the technologies we have invented to satisfy our needs are now so strong that they construct and determine *those very needs* themselves. This view holds that technology is separate from social development, and that the political structure is always reflecting a technological situation. Kranzberg's second law of technology comes to mind: "Invention is the mother of all necessity" (Kranzberg 1986).

Technological determinism has a long history and in its modern form it can be traced back to John Ruskin and William Morris who, in the late 19th century, responded to the increase of automatisation in work by founding the Arts and Crafts movement that would emphasise crafts; the skilled hand and personal style. Even earlier advocates could be found in Dickens, Emerson, Thoreau and Twain. A well known technological determinist in the field of media is Marshall McLuhan, who is famous for his "the medium is the message" mantra that has been influential in the field of media studies. McLuhan has written that "such inventions as the horse collar quickly led to the development of the modern world" (McLuhan & Watson 1970: 121). Another cultural critic, Jacques Attali, provides a typically deterministic view of technology when he writes: "In music, the instrument often predates the expression it authorizes, which explains why a new invention has the nature of noise; a 'realized theory' (Lyotard), it contributes, through the possibilities it offers, to the birth of a new music, a renewed syntax" (Attali 1985: 35).

Technological determinism is often attributed to dystopianism and negative portraits of technology. It is seen as an inhuman force that drives society into emotionally detached behavioural structures, where ergonomics, efficiency and rationalisation are the guiding principles. Technology thus causes alienation and dehumanisation. Philosophers such as Adorno and Horkheimer (1972), Marcuse (1964), Habermas (1970, 1984), and Ellul (1964) are all

Marxist thinkers who have characterised technology as a way of revealing the world, a form in which we experience it: technology as a cultural absolute. In the technocratic society we become mindless cogs in the machinery of the state. Instead of shaping technology we are shaped by it. Once the liberation of humanity from its enslavement to nature, technology has now become a means of (political) domination. We are destined to be eternally adapting to the rational forces of technology. There are political, sociological and psychological issues that have to be dealt with when technology is seen from this perspective and the above mentioned philosophers together with sociologists like Blauner (1964) and Braverman (1974) have addressed this issue in their work. Although Langton Winner rejects being categorised either as a technological determinist or a social constructivist, his 1980 paper "Do Artefacts have Politics" has been influential in the camp of technological determinists. Winner demonstrates that technology is never neutral and can have strong political agenda and/or effects. Firstly, technologies are designed (consciously or unconsciously) to have social effect. They are "designed and built in such a way that it produces a certain set of consequences logically and temporally *prior* to any of its professed uses" (Winner 1980: 30). Secondly, some technologies are *inherently* political in the sense that they require a political infrastructure to take control over them, such as the various nuclear technologies. Winner's determinism has been criticised for being naïve and lacking in interpretative flexibility, where the object of analysis could have been deconstructed from a much wider and less subjective stance (Woolgar 1991: 34).

Technological determinists often see technology as an autonomous force that we have lost control over (Ellul 1964). Ivan Asimov writes

> The whole trend in technology has been to devise machines that are less and less under direct control and more and more seem to have the beginning of a will of their own…. The clear progression away from direct and immediate control made it possible for human beings, even in primitive times, to slide forward into extrapolation, and to picture devices still less controllable, still more independent than anything of which they had direct experience. (Asimov 1981: 130)

Ellul similarly portrays technology as something cold and impersonal, devoid of culture: "Technique has become autonomous; it has fashioned an omnivorous world which obeys its own laws and which has renounced all tradition" (Ellul 1964: 14). The danger here for Ellul is that technology becomes an end in itself rather than a means to an end. Feenberg (1999) uses the word "essentialism" for determinism and states that it reduces everything to functions and raw materials. Technology is principally goal orientated, asphalting over all practices that embody human meaning. "Efficiency sweeps away all other norms and determines an autonomous process of technological development" (ibid: viii).

Even at the Massachusetts Institute of Technology (whose name instantiates one of the first public usages of the word *technology* in the modern sense when it was founded in 1861) technology is seen as posing problems. Rosalind Williams, a professor at the institute and former dean of students and undergraduate education, reports on the constant tension between technology and culture within the institution. She quotes a colleague who observes when choosing custom made software for administration: "If the vendor's worldview is the same as yours, you're OK. If not, one or the other has to change" (Williams 2000: 651). She quotes another college in engineering who declared: "We have to break the old culture here" (ibid.) when faced with how "old" cultural conventions within the institution did not fit with a new "improved" and more "rational" software system. Williams concludes: "even at MIT 'technology' can be seen as an outside force that invades and disrupts the indigenous world. This is a reminder of the reflexive nature of modern technology. It is not simply an assemblage of devices sent forth from centers of invention to 'impact' less advanced cultures. Technology reflects back, in complicated and unpredictable ways, on the inventive centers too" (ibid.).

From the perspective of technological determinism, music software is a machine that produces two products: the musician and the music. The plethora of musical tools are but one hybrid music machine that is set into action by marketing powers of the commercial world (the music, hardware, and software industries). We could analyse how the production, reproduction and marketing strategies of the music world condition the way music is composed and disseminated today. Following Attali's statement above, an analysis could be made of how certain software packages have literally created musical styles that can be heard in nightclubs all over the world. Or echoing Ellul (1964), we could show how aesthetic ends are so implicated in the technical means used to realise them that it doesn't make sense to separate ends and means. We might analyse how the electric guitar and amplifier created Hendrix's style or how the drum machine resulted in the rap movement. Here we might see beatboxing as the epitome of technology's success where we find the human imitating the machine, reversing the idea that the machine should be to the service of the expressive human. The intention here is not to engage in such a jejune analysis. There are many other (f)actors at play, too intricate to be reduced into this one-dimensional picture, as we will see later in this work.

2.3.2 Social Constructivism

Of course, there may be one dominant use of a technology, or a prescribed use, or a use that confirms the manufacturer's warranty, but there is no one essential use that can be deduced from the artefact itself. (Oudshoorn & Pinch 2003: 2)

The thinkers of social constructivism are more optimistic than their peers in the deterministic camp. They argue that technologies do not arise from a void as they are always an answer to problems that originate in social contexts. The properties of technological artefacts and social processes are consistently interdependent and the interesting task is to analyse this relationship. Here the market is often seen as a grounding force where developers of technology are driven by the desire to sell their products: they are prompted to design a product which already is *needed*, and when the product is out they have to respond to the ways in which people actually *use* their products. But the market is not that simple; it is driven by rhetoric and hype, which is the engine behind the consumerism of contemporary society. "From this perspective, the prime mover is the rhetoric, at the hands of the powerful, rather than the technology *per se*" (Grint & Woolgar 1999: 26).

Social constructivism, also called technological constructivism, looks at technology from a more human centred position and emphasises not only the ability but the duties of humans to engage critically with technology. Bijker criticises technological determinism for its dystopian worldview where humans have lost their say in the face of the unstoppable force of technological progress, thus rendered politically powerless as this fetishism for progress "inhibits the development of democratic controls of technology because it suggests that all interventions are futile" (Bijker 1995: 281). And further, "if we do not foster constructivist views of sociotechnological development, stressing the possibilities and the constraints of change and choice in technology, a large part of the public is bound to turn their backs on the possibility of participatory decisionmaking, with the result that technology will really slip out of control" (ibid). Furthermore, the idea of technology as a layer on top of society that we cannot engage with and change is but a social paralysis: "technology is assumed to have objective effects which can be measured and predicted and which are largely unaffected by the human actors involved. We refer to this tendency as technicism" (Grint & Woolgar 1997: 7). For Grint and Woolgar, technology should never be treated as a given, as something objective and unproblematic. Technology is not beyond the domains of sociological analysis.

Technological constructivism also aims to show how both science and technology are socially constructed cultures. (Pinch & Bijker 1984). The study branched out of the sociology of scientific knowledge (SSK) (Woolgar 1991) and in the 1990s it established a broad consensus that the meaning and form of technological artefacts are socially shaped and originating from the users of the technology. Here a new technology (such as the camera or the telephone) is seen as containing a wide interpretive flexibility by the users initially, but then eventually "technology stabilizes, interpretative flexibility vanishes, and a predominant meaning and predominant use emerge" (Oudshoorn & Pinch 2003: 3). The principle of "symmetry," as argued by the constructivists, states that there might be alternative technological inventions that would serve similar purpose as the one that gains acceptance. However, it is the social structure, the local circumstances, that define what technology is taken into use. Social groups determine, by selecting for survival, the designs that solve their problems or fulfil their desires. New technological artefacts tend to be open in meaning initially but concretisise when a social group believes that their goals with the technology have been fulfilled. This is called a “closure” (Pinch & Bijker 1987). The closure produces a “black box” which is not called into question after it concretisises and its social origins are forgotten.

The focus of sociological constructivism has been more on technological artefacts and their adoptions in society, rather than the study of design and creativity of technological inventors. Two of the main thinkers in the field describe the social shaping of technology thus:

> Technologies do not have a momentum of their own at the outset that allows them … to pass through a neutral social medium. Rather, they are subject to contingency as they pass from figurative hand to hand, and so are shaped and reshaped. Sometimes they disappear altogether: no-one felt moved, or was obliged, to pass them on. At other times they take novel forms, or are subverted by users to be employed in ways quite different from those for which they were originally intended. (Bijker & Law 1992: 8)

Grint and Woolgar (1997) reject the notion that technology contains any essential properties that affect human culture. They call this stance “technological anti-essentialism”: technology’s meaning is the outcome of interpretation and human context. Albeit not a social constructivist, Langton Winner takes a similar position, illustrating that technologies and their use are actually “forms of life” in the Wittgensteinian sense (Winner 1983). It is the use of technologies that define their meaning. Winner traces this idea back to Marx and Engels who talk about *modes of life* in their *German Ideology* and emphasises the ethical role of users of technology: “As we ‘make things work’ what kind of *world* are we making?” (Winner 1983: 112).

Bijker, Hughes and Pinch propose a contextualist approach that attempts to bridge the internalist (the development of technology from its own logic) and the externalist (technology is always a response to human needs) historical approaches to technology (Bijker et al. 1987). Many of the historical accounts of technology and society are good at describing situations in society, and how technological change occurs in limited time periods, but they are not good at explaining the dynamics of historical situations; of how people in different periods have interacted with technologies through active involvement and change.

2.3.3 Socio-Technical Interactionism and Structurational Approach

The dualism of technological determinism and constructivism is rejected by thinkers whose view is that technology and the context in which it is used are always already interdependent. Let us take for example the old schism between science and technology that can be traced back to Plato’s account of *episteme* and *techne*. Technology is not merely applied science and there is no science without technology: we need an account that describes the interaction between the two fields. One such view is represented with MacKenzie and Wajman’s collection *The Social*

Shaping of Technology (1984) where many of the authors subscribe to a so-called *socio-technical interactionism,* although most authors in the book have roots in social constructivism. The view here is that technological artefacts are always designed from a specific sociological perspective that inform the design of the object. However, when a technological artefact is released to the world, social structures take to use it in ways that can be unpredictable and unintended. For Grint and Woolgar, the authors in this volume still end up "struggling with a dualism between 'technology' and 'the social'. Does technology… determine, or is it determined by, the social?" (1997: 21).

A proposed solution is provided with the theory of *structuration*. In his book *The Constitution of Society*, Antony Giddens tries to bridge the dichotomies between agency and structure, subject and object, micro and macro. The theory of structuration tries to avoid technological or social determinism. Giddens looks at social practices "across space and time" (Giddens 1984: 2) and finds that structure and agency are complementary phenomena that constantly influence each other. From the perspective of technology we see that new technological products change social patterns, but at the same time social behaviour can re-interpret and readapt technology to functions that it was never designed for. Here the socio-technical relationship is seen as a dialectical process. A prominent writer advocating this view is Wanda Orlikowski (2001) who defines the premises of technology as twofold:

> that technology is created and changed by human action and that technology is interpretively flexible and its use (consequences and trajectories) depends on human agents. Because the structures of both organizations and technologies pose the capability to produce and reproduce, and human agents possess the capability to make sense of their surrounding environment and to reflect their knowledge in their behaviors, the distinction between intended (inevitable, planned, predicted, or managed) and unintended (emergent, unplanned, unpredicted, or unmanaged) narratives become far less clear. (quoted in Lin & Cornford 2000: 197)

Structures are composed of generative rules and are the frameworks that surround and influence human action. Technological structures thus guide and direct human action. However, the source of all structure is action, and they are produced and maintained through continual social action. This reciprocal relationship is continuous where neither part can be analytically separated. There would never be a situation where either structure or action takes priority over the other in human existence.

2.3.4 Technological Hermeneutics – Technology as Text

In their book, *The Machine at Work,* Grint and Woolgar (1997) propose that technologies should be treated as texts written by the designers, producers and marketers of the technological artefacts that have to be "read" (i.e., interpreted) by the users or consumers. The writers of the technological text can have different strategies for the intended usage of the artefact. A quick glance at the recent Apple iPhone shows how adamant the producer is to keep a strong grip over 3rd party development and thus customer usage of the device. Apple is limiting the scope of interpretations and affecting the potential meaning of the artefact. The users, by contrast, are always constantly interpreting the device in new ways, recontextualising its functionality and purpose. "[T]he view of technology as text suggests that the likely 'impact' of new technology is 'built in' during the process of evolution and design and reconstructed and deconstructed during usage" (Woolgar 1991: 37). As pointed out by Hutchby (2001), this view can be seen as too open, where all readings of the text are possible and equally valid. Grint and Woolgar try to refrain from suggesting such openness, but they can hardly resist: "In disassociating the upshot of reading and interpretation from any notion of the inherent quality of the text… we do not mean to suggest that any reading is possible (let alone that all readings are equally possible), although in principle this is the case" (Grint & Woolgar 1997: 72).

But what role can hermeneutics, a science of textual interpretation originating from the German theologist Schleiermacher in the late 18th and early 19th century, perform in the study of modern technology and its relationship with humans? How is it relevant to the study of technology?[25] Wilhelm Dilthey (1900) argues that the methodologies of the natural sciences would never work in the humanistic sciences. The humanistic sciences deal with texts, verbal utterances, art, actions and expressions whose "mental content" or intentions that need interpretation. Dilthey develops a methodology of interpretation that focussed on "understanding" (*Verstehen*) as opposed to the natural sciences explanatory knowledge (*Erkennen*). There is objective truth to be found, we just need solid systematic methods to interpret texts and human actions correctly. In later works, Dilthey develops a circular method (a hermeneutic circle) where the author's historical context had to be studied in order to understand the meaning of the analysed object.

In *Being and Time* (1927) Heidegger rejected all such objectivism, undermining his teacher Husserl's notion of *bracketing* (where the subject/interpreter is removed and the object analysed in its pure essence). Heidegger rejects Dilthey's notion of objectivity as all interpretation will always be performed from a historical standpoint that is compromised by the interpreter's lifeworld. (*Lebenswelt*). For Heidegger, humans, as being-in-the-world (*Dasein*), are always perceiving things according to how they encounter and use them in their daily life. All understanding is therefore an *existential understanding*, a result of our "thrownness" into the

[25] Although outside the scope of this work, the last decades has prompted various philosophers and technologicians to use hermeneutics to point at the mistaken optimism inbuilt in the project of artificial intelligence. Dreyfus (1972) points out how mental processes cannot be formalised and Winograd and Flores (1986) argue for the embodied and hermeneutic condition of human existance. These authors question the cognitive science paradigm in AI by referring to areas such as common sense reasoning and natural language processing.

world. Heidegger builds up his hermeneutics as a theory of existential understanding rather than a theory of interpretation. Classical philology's focus on biological facta is supplanted with a focus on a conscious recognition of one's own world. Instead of Dilthey's hermeneutic circle, Heidegger introduces a more fundamental *ontological hermeneutic circle*.

A student of Heidegger, Hans-Georg Gadamer, took his teacher's concept of hermeneutics and developed it further in his book *Truth and Method* (1989). For Gadamer understanding of a text always involves a "fusions of horizons" as the lifeworlds of the writer and the reader are never isomorphic. The meaning of a text is never fixed but changes over time depending on how it is read (in which circumstances and from which questions are posed when the reading is initiated). A prominent Frankfurt-school philosopher, Jürgen Habermas, rejected the relativistic trends in Gadamer's theory, and developed a theory of universal pragmatics, of *ideal speech situations*, where two parties with similar pre-understanding are able to understand each other (1976). This led to a famous long-standing debate between Gadamer and Habermas, one that was never resolved properly.[26]

Paul Ricoeur finally brought forth a grand synthesis of the various forms of phenomenologies and hermeneutics that had appeared by the 1970s. Ricoer's project, often called *phenomenological hermeneutics,* was to show how phenomenology was already hermeneutic and vice versa:

> [Phenomenological hermeneutics is possible] by establishing, on the one hand, that beyond the critique of Husserlian idealism, phenomenology remains the unsurpassable presupposition of hermeneutics; and on the other hand, that phenomenology cannot carry out its programme of *constitution* without constituting itself in the *interpretation* of the experience of the *ego.* (Ricoeur 1981: 114)

Ricoeur claims that understanding of language is always achieved from an ontological stance, but from a historically rooted subject, and as such, the subject's being is never identical with immediate experiences. Ricoeur is known for his hermeneutic of the self, the method in which we understand ourselves, and he developed a theory of narrative to this aim. Our self-understanding is "fictive" and we interpret data from our imagination and memory at run-time, so to say. As opposed to Derrida and Foucault who see the self practically as an effect of language, Ricoeur grounds the self firmly in the body through his phenomenological stance.

When technologies are viewed from the hermeneutic standpoint, we can see the limitations of both the deterministic and the social constructivist views of technology. By viewing technologies as text, one can perform various readings that emphasise different aspects of their

[26] As mentioned in the introduction, both Gadamer and Habermas have been attacked by Derrida for their optimistic belief in the possibility of objective interpretation.

nature, such as feminist (Haraway 1991; Cowan 1983; Trescott 1979), Marxist (Marcuse 1964), post-colonial (Said 1978), semiotic (Grint & Woolgar 1997), deconstructive (Derrida 1974), cultural and media studies (Baudrillard 1988a; Kittler 1999; Hansen 2000; Fuller 2005), sociological (Hutchby 2001);[27] or psychoanalytical (Lacan) readings. Some of the problems are likely to be originating from one-sided viewpoints that omit the general picture of the technological production process.

This idea of technology as a text which should be studied through hermeneutic methods has been influential. Woolgar (1991) talks about "configuring the user" where the designer tries to constrain the future use of the artefact. This approach views the user from the perspective of the designer, but a later work (Mackay et al. 2000) criticises this view for being too simplistic: "designers configure users, but designers in turn, are configured by both users and their own organizations" (ibid: 752). Designers have to follow various organisational methods that constrain their own practices and the role of the media, the marketplace, competitors and consumption junctions[28] (Cowan 1987) cannot be underestimated. The analysis of "encoding" by the designer (Woolgar 1991) and the "decoding" by the user (Mackay et al. 2000) are important tools in the proceedings of this work. However, for the current particular topic of music technology, a more complex and integrated theoretical model is needed: one capable of analysing the structures and interaction between the heterogeneous multiplicity of actors that are at play in modern music technology. I propose using the actor-network theory as a methodology for such techno-musicological analysis.

2.3.5 Technological Momentum

The problems that emerge when people swing between technological determinism and social constructivism derive from the intense exclusive focussing on one cog of a much more complex machinery, of which the market is one element. Williams (2004) suggests after having been torn between technological determinism (endless software upgrades, tons of e-mails, new systems that one has to adapt to) and social constructivism (as a historian she is used to looking at the sociological conditions of cultural change); perhaps what really is taking place is market determinism or tyranny of the market: "To focus on the socially constructed design of any one product misses the point – the larger determinism of market-driven technology that profits from change for the sake of change" (ibid: 664) and further: "Technological change is fuelled by

[27] It is interesting in this context how Hutchby (2001) proposes the use of a theory of *affordances* as a shift of focus in the sociology of technology to go beyond the dualism of meaning residing in the designer or the user (sender or receiver). He points out that the affordances of an artefact do not change whether a person has interest in it or not. The affordances are there, objectively so, and we should pay attention to this material substratum of the artefact. It is up to the user to decide when and how they use it. However, Hutchby does not provide a systematic methodology of how to analyse the meaning of technological artefacts, their affordances, and how they are taken into different use in different cultural contexts.

[28] Cowan defines a *consumption junction* as "the place and time at which the consumer makes choices between competing technologies" (Cowan 1987: 286).

money pouring into product development from interlocking corporations, some of them with virtually unlimited resources and global reach" (ibid: 665). Although the market is a big factor, there are various other (f)actors at play, such as the ethics of the open source sector, governmental policies and non-commercial experimentations of human creativity in the fields of art, design and engineering. What Williams' comments illustrate is the complexity of the situation and how, indeed, the technology is outside the control of any one agency or institution.

Thomas Hughes uses systems theory to develop his concept of technological momentum, explaining how technology establishes a certain inertia through time. The technological system includes the technical (hardware and software) and the social (institutions, values, interest groups, social classes, and political and economic forces). The forces within the system are not symmetrical through time. Hughes shows how, initially, technical inventions tend to be socially shaped and adapted through practical reception in society, but after some time, the invention establishes itself and becomes part of the technological institutions where it tends to determine social structures:

> A technological system can be both a cause and an effect; it can shape or be shaped by society. As they grow larger and more complex, systems tend to be more shaping of society and less shaped by it. Therefore, the momentum of technological systems is a concept that can be located somewhere between the poles of technological determinism and social constructivism. (Hughes 1994: 112)

The crucial element here is that of time. Technologies change over time and establish themselves. Hughes contributes with a dynamic systems picture of the lifetime and evolution of technologies and how they can reside in the indeterminacy of social shaping at one moment, but later become rigid techno-social conventions that is almost impossible to change.[29] A good example of the initial flexibility of technologies surprised researchers at Nokia when they went to Uganda and discovered that the population there had implemented a complex money transfer system using mobile phones. The practice is called Sente:

> Joe lives in Kampala and wants to send his sister Vicky 10,000 Ugandan Shillings - about 4 Euros. He buys a pre-paid top up card for that amount but instead of topping up his own phone calls the local phone kiosk operator in Vicky's village. The phone kiosk operator uses the credit to top up his own phone, takes a commission of anywhere between 10 and 30% and passes the rest onto Vicky in cash. The kiosk operator then resells the airtime at a profit (it is after all his business). (Chipchase 2006)

[29] Hughes is adamant to show that technology is not an uncontrollable force: a system with great technological momentum can be forced to change direction through forces such as improved inventions, changed economic reality, or political policies.

The flexibility of Ugandan society makes this easily possible, whereas the technical complexity (the degree or mass of technical structures) of Finnish society is much higher, which complicates the initial implementation of such money transfer systems. In Hughes' terms, the Finnish system would be defined as inert and fixed, with a strong technological momentum. The Ugandan practice is still open and flexible but will establish itself over time and, depending on local politics, become nationalised or institutionalised in some form or another.

2.4. Actor-Network Theory: The Agency of the Nonhuman

Neither systems theory nor theory of structuration have room for the autonomy of the technical artefact. For them, the artefact gains its meaning from its *use* in the social context. In these theories, successful technical artefacts are those made by innovators who thought in social, political and economic terms about the meaning of their inventions.[30] While the systems theory focuses on the larger structure in the growth of technical systems, the structuration theory deals with the actor/systems or agency/structure relationships. One problem with these theories is that they tend to become anthropocentric in that they define the inventor, the entrepreneur, technical artefacts, etc. as naturally occurring phenomena. Here the actor-network theory, as developed by Bruno Latour (1979, 1982, 2005), Michel Callon (1986, 1997), John Law (1992, 1999), and others, provides a useful theoretical framework for analysis.

The actor-network theory has its roots in social constructivism but allows for a more complex investigation as it does not view technological artefacts as passive objects, but rather raises their status to autonomous agents in a heterogeneous network of other actors. The actor-network theory criticises the limited scope of technological determinism, claiming that it lacks consideration of what is brought together and transformed through the formation of networked systems. It also points out the weakness of seeing the human as the only possible actor in the technological spectrum. Instead of viewing technology as technical artefacts and science as knowledge, the networked view describes the relationship between the two interdependent fields, extending further into a multiplicity of other fields, with which the two interact. The problem with traditional analyses of technology vs. society has been their reductive nature and how they tend to create fundamental divisions between the technical and the social, treating each division with a different discourse. "The rule which we must respect is not to change registers when we move from the technical to the social aspects of the problem studied" (Callon 1986: 200). Callon proposes that we analyse the "actants" (named so to include both human and

[30] Hughes (1986) shows how creators of systems often have no respect for categories of professional boundaries. They create beyond a specific field. The categories of "economy", "culture", "science" and "technology" blur into something that Hughes calls the "seamless web." Hughes quotes Callon who asks why we categorise, or compartmentalise, the elements in a system or network "when these elements are permanently interacting, being associated, and being tested by the actors who innovate?" (ibid: 287).

non-human actors) through a prism of three principles: a) of ‘agnosticism’ or impartiality between actors, b) the principle of ‘generalised symmetry’ where all conflicting viewpoints are explained in the same terms, and c) we should reject all a priori distinctions between the so-called social, natural and technical worlds and use the method of “free association” to analyse the relationships between actants (ibid.). The actor-network is a heterogeneous system of actors and networks, of human and nonhuman agency:

> The actor network is reducible neither to an actor alone nor to a network. Like a network it is composed of a series of heterogeneous elements, animate and inanimate, that have been linked to one another for certain period of time... for the entities it is composed of, whether natural or social, could at any moment redefine their identity and mutual relationships in some new way and bring new elements into the network. An actor network is simultaneously an actor whose activity is networking heterogeneous elements and a network that is able to redefine and transform what it is made of. (Callon 1997: 93)

Similarly, Law (who defines the actor-network theory as a theory of agency, knowledge and the machine) proposes that we cease looking at the objects themselves (or structures and subjects in sociology) and rather “start with interaction and assume that interaction is all that there is. Then we might ask how some kinds of interactions more or less succeed in stabilising and reproducing themselves: how it is that they overcome resistance and seem to become ‘macrosocial’; how it is that they seem to generalise, scope or organisation with which we are all familiar” (Law 1992: 2). The problem with traditional technology studies is how the various schools have focused on the analysis of a single stratum, outlined its properties and functional mechanisms, and then described how the projected power of that structure affects the other field. In the actor-network theory, structure is not a noun, but a verb: it is not an object but a function. People and machines are not naturally occurring categories, but effects of processes that need to be studied in their own right. The actor-network theory criticises the focus on singular ontological categories and opens up for more functional description:

> Actor-network theory does not accept this reductionism. It says that there is no reason to assume, a priori, that either objects or people in general determine the character of social change or stability. To be sure, in particular cases, social relations may shape machines, or machine relations shape their social counterparts. But this is an empirical question, and usually matters are more complex. So, to use Langdon Winner’s (1980) phrase, artefacts may, indeed, have politics. But the character of those politics, how determinate they are, and whether it is possible to tease people and machines apart in the first instance – these are all contingent questions. (Law 1992: 4)

Law is here pointing to Winner’s 1980 text already mentioned, where he develops an argument of the political nature of technology. Winner takes the example of how the bridges into Long Island in New York were purposefully built with low headroom so buses could not enter the island, making it harder for the lower classes to enter (they would typically use public transport whereas the better off classes would use cars). Latour (1988: 36) similarly points at how the Parisian Metro tunnels were built small in order to prevent the privatisation of the Metro. But for Latour the tunnels had become an actant that prevented the possibility of change, in a manner similar to Hughes' theory of the technological momentum: "They shifted their alliance from legal or contractual ones, to stones, earth and concrete. What was easily reversible in 1900 became less and less reversible as the subway network grew" (ibid.). Grint and Woolgar take an anti-essentialist stance, claiming that Latour is here guilty of technicism, i.e., the technology has now become something objective and beyond human influence. But the weakness of this argument is that it does not take Latour's observation seriously, namely that the tunnels of the Metro have now become stabilised, they are an actor (with a network of other actors behind it) that will affect all decisions taken from now on. Practically, it is not socially changeable any longer, it has become punctualised,[31] and in this case a political fact. This does not mean that the actor-network theory is an "essentialist" theory, but rather that they have accepted how technical objects have a tendency to pointilise, to become stabilised blackboxes resistant to change.

The actor-network theory thus proposes that we look symmetrically at human and nonhuman actors as nodes in a complex network. Technical objects, or the nonhuman actors, are not mere "things" but active nodes in this network. Social behaviour does not precede or constitute technology but emerge with technology as shown with the principle of symmetry. The claim that things are agents is of course rather unorthodox in social theory, but Latour is ardent to demonstrate the agency of nonhumans through powerful arguments:

> The behavior imposed back onto the human by nonhuman delegates [I will call] *prescription.* Prescription is the moral and ethical dimension of mechanisms. In spite of the constant weeping of the moralists, no human is as relentlessly moral as a machine…. We have been able to delegate to nonhumans not only force as we have known it for centuries but also values, duties and ethics. It is because of this morality that we, humans, behave so ethically, no matter how weak and wicked we feel we are. (Latour 1992: 232)

The quote above is from an article called "Where are the Missing Masses: The Sociology of a Few Mundane Artefacts" and the title encapsulates the point well. The missing masses that Latour looks for are inscribed in the technological artefacts we use to structure, control, and police our society or simply our daily life. Technology is "programmed" by us with an agenda – it contains a script that we incorporate it with – and through this process we "delegate" functions to both humans and nonhumans that make up the network.

---

[31] A term used to denote the effect that happens when a complex system of heterogeneous actor networks become stabilised in one object. The concept is akin to "abstraction" in object oriented programming.

The actor-network theory, through the theory of punctualisation, is able to look at technological objects, socials structures, and attitudes such as how society complies with "the demands of new technology" (Burnes et al. 1988: 7; Williams 2000) and explain them as abstractions of other actor-networks.[32] Or as Grint and Woolgar point out: "who says these are the demands of the new technology, under what circumstances, how and why?" (Grint & Woolgar 1997: 18). The actor-network theory is able to analyse the agency of nonhumans and humans through the process of tracing *delegations* (when we shift an action into a nonhuman object – such as door closer, speed bump, or a word editor). In the actor-network theory there is never a "they" who are controlling "us" through some technocratic structure. All is co-dependently arising through a complex system of actors and networks that stabilise and destabilise in accordance with energy flows in the system.

2.4.1 De-scription of Technical Objects

In a rather more developed analysis than the hermeneutic one of Grint and Woolgar (1997), Akrich (1992) proposes the metaphor of the *script* as a replacement for the information theory-esque writing-reading relationship between a designer and a user. Like the film script, "technical objects define a framework of action together with the actors and the space in which they are supposed to act" (ibid: 208) and the technical artefacts "participate in building heterogeneous networks that bring together actants of all types and sizes, whether human or nonhumans." (ibid: 206). This script approach gives more improvisational role to the users as active participants in defining the meaning of technology. Akrich is careful to prevent any misunderstanding that this approach is based on technological determinism: "we cannot be satisfied methodologically with the designer's or user's point of view alone. Instead we have to go back and forth continually between the designer and the user, between the designer's projected users and the real users, between the world inscribed in the object and the world described by its displacement" (ibid: 209). However, analysing an artefact's reception and use becomes more difficult if the script has become stabilised or concretised. This happens through time where technologies (both techniques and technical objects) become "blackboxed": "Here the innovator is no longer present, and study of the ordinary user is not very useful because he or she has already taken on board the prescriptions implied in the interactions with the machine" (ibid: 211).

Through an analysis of how the electricity system introduced in the Ivory Coast changed social structures in villages (where the power of land allocation was taken from the elders and became an organisational task of the electricity provider) and how, in turn, the villagers were incorporated as state citizens through the process, Akrich shows how technology

---

[32] The roles of linguistic metaphors are abundant and there are multiple levels of explaining the metaphorical nature of our thinking as Lakoff and Johnson show so well. (Lakoff & Johnson 1999).

determines and defines actors. The designers of systems are not only designing the system itself (here the electricity network) but also which powers and roles are delegated to the nonhuman and human actors respectively. There is always a chance for resistance to the inscription of technical objects, but in practice "so long as the circumstances in which the device is used do not diverge too radically from those predicted by the designer, it is likely that the script will become a major element for interpreting interaction between the object and its users" (ibid: 216). A stabilised network of technical objects and (human and non-human) actors is established only when the script is acted out, either in conformity with the script or as a resistance to it (antiprogram). Technical artefacts are only defined by their use, and users are defined by the technical artefacts. "It is only after the event [introduction of a new technology] that causes are stabilized. And it is only after the event that we are able to say that objects do this, while human beings do that" (ibid: 222).

2.4.2 Actor Network Theory and Technological Mediation

What, then, is a tool? The extension of social skills to nonhumans. (Latour 1994: 62)

In a 1994 paper, Bruno Latour, through a criticism of materialist and social ideas of the foundation of technology, lays out four principles in order to understand how technology performs a specific mediation in our interaction with it. The aim is to get beyond subject-object dualism and rather try to understand human use of technology as a symbiotic system of multiple origins. The first principle of technological mediation is *translation.* Here the subject (the agent) is interrupted in one way or another and has to take a detour through another agent. When those agents meet, a translation occurs resulting in an outcome that neither of the agents might have preferred. Latour provides the example of the gun. A gun is an actor, a neutral one until another actor picks it up and thus changes the gun's relationship with the world. It has now become a weapon. And the actor holding the gun has now become another subject as well. When the actor carrying a gun faces a problem, the solution to the problem is likely to be different to the solutions had she not carried a gun. The gun inheres its inscription, and the carrier is enframed by this inscription. The danger here is to fall into the opposite camps of materialistic ("guns kill people") or sociological ("people kill people") versions of the human-technology relationship. It is more complex than that and the actor-network theory is able to analyse this with the analytical tool of the script.[33]

[33] Analogously, the choice of a musical software is not a neutral decision of a musician that could make her music in any of the available tools. The musician that chooses Reason is a very different musician than if she would choose to work with Pure Data. The ideologies are different, the sound is different and the affordances the tools provide for the musician are strongly determining, although in principle open for any type of expression.

The second principle is *composition* or symmetry. The actor is never a sole actor. The complex relationship of an actor with other actors create the actor-network, i.e., the actor which is him/herself already a network. “It is by mistake, or unfairness, that our headlines read, ‘Man flies,’ ‘Woman goes into space.’ Flying is a property of the whole association of entities that includes airports and planes, launch pads and ticket counters” (Latour 1994: 35). Analysis has to start before we arrive at the human actor; the idea of the acting subject is only possible as the arrival point, not the starting point. The definition of the gunman could be of a man carrying a gun, but it could also be that of a class of people normally carrying guns, or even subindividual unconscious motives in the actor. In turn, the definition of the gun could be a definition of a specific culture, a lobby of manufacturers that have strong political power, or the physical machinery itself.

The third principle is that of (reversible) *blackboxing.* Any object is always an abstraction or encapsulation of a heterogeneous multiplicity of other objects that have their own origins, traditions and history. Nothing has a sole origin but derives from various practices and material origins. A theory or an object becomes a black box when its ingredients cease to be visible or remarkable and turns into a tool that does not need to be analysed anymore. Operations that are transparent initially become black boxes when their origins disappear. A historical fact, a class in a programming language, a car engine, a social convention, a scientific theory, etc. can all be black boxes. We do not need to question everything to its roots all the time when we have black boxes, but we should be aware of their complex heterogenesis. This prompts Latour to tease the thinkers in the essentialist camp: “Yet there remain philosophers who believe there are such things as objects” (ibid: 38).

Latour’s final definition of technological mediation is *inscription*: the objects reify or materialise a program of action within the actor. Every artefact carries its own script. This is the meaning we covered in the section above on the de-scription of technical objects.

2.4.3 Shifting

There is no tool, no medium, only mediators (Latour 2005: 148)

In order to understand technological mediation as it affects us, we need to perform what semioticians call *shifting,* i.e., a shift in the frame of reference, an interpretive act where we try to dis-cover the origin of the technological object. It is a temporal, spatial and “actorial” shift as one has to go back in time, to a different place and imagine what were the reasons for an actor to act as they did and what the object now represents: “An object stands for an actor and creates an asymmetry between absent makers and occasional users” (Latour 1994: 40). But it is not a semiotic interpretation, as in a fiction; it is not that “I am here *and* elsewhere, that I am myself *and* someone else, but that an action, long past, of an actor, long disappeared, is still active here, today, on me – I live in the midst of technical delegates" (ibid: 40). Latour here derives at an understanding of technology similar to that of Bernard Stiegler to be presented in the next section, namely how technology constitutes human history through its means of being an external objectification of (human) technical reasoning:

> The whole philosophy of techniques has been preoccupied by this detour. Think of technology as concealed labour. Consider the very notion of investment: A regular course of action is suspended, a detour is initiated via several types of actants, and the return is a fresh hybrid that carries past acts into the present and permits its many makers to disappear while also remaining present. (ibid: 40)

For Latour, driving over a speed bump on the road means encountering various actants, such as the legislative body, the police, etc. "lending a policeman the permanence and obstinacy of stone" (ibid: 40). Every hour we encounter multiple absent makers of the technologies or objects that we use constantly during our waking hours. They are remote in space and time, but simultaneously active and present. In fact, the speed bump is not what it seems: "[It] is not made of matter, ultimately; it is full of engineers and chancellors and lawmakers, commingling their wills and their story lines with those of gravel, concrete, paint, and standard calculations" (ibid: 41). But this does not mean that we can restore the human agency in the technical object (technological social as presented above); there are too many actor-networks behind the object for it to be traced to some single actant's will. Neither the sociological version (which would see us, the makers, in the machines) nor the materialistic version (which would see technology as the determining structure that moulds us, the humans into its program) is sufficient in explaining the complexities of technological inscriptions. The technical mediation, the translation that we are observing in an actant (the speed bump) and the network that created it (other actants) is a process in which "society and matter exchange properties" (ibid: 41).

We find in Latour a theory of subprograms, where the subject is never a pure subject acting on inert objects, but rather a subject situated in a context of other subjects and (technological) objects, each determined by the multiplicity of heterogeneous actants that found the activity at hand. "We are an object-institution" (ibid: 45). This requires a rigid analysis of social structures, of which social practices form the objects we use:

> Objects that exist simply as objects, finished, not part of a collective life, are unknown, buried under soil. Real objects are always parts of institutions, trembling in their mixed status as mediators, mobilizing faraway lands and people, ready to become people or things, not knowing if they are composed of one or of many, of a black box counting for one or of a labyrinth concealing multitudes. And this is why the philosophy of technology cannot go very far: an object is a subject that only sociology can study – a sociology, in any case, that is prepared to deal with nonhuman as well as human actants. (ibid: 46)

The contemporary situation is not one of alienation between humans and technology, where the distance between people and technology is increased; it is a situation where objects and subjects are made simultaneously to a more intense degree than ever before, where we get a deeper intimacy – a more intricate mesh – between the subject and the object. The dualism found in older theories of philosophy of technology is thus eliminated through the tools of actants that can be nonhuman and subpersonal and/or networks (actants form networks) that are both technical and social in nature.

The actor-network theory's capacity of performing symmetrical analysis of human and non-human actors – the concepts of translation, composition, blackboxing, inscription, and shifting – becomes useful when analysing musical technologies and the practices of their use. We are not determined by technology, nor do we simply shape it through our appropriation of them – we, the users and designers of music technologies, are tangled up in a complex network of agency where tools, traditions, protocols, programming languages, designers, users, musical styles, and the market all yield influence on our actions, much of which is beyond our analytic capacity.[34] Part II of this thesis will proceed in the spirit of actor-network theory when analysing musical technologies, its designers and users.

2.5 The Human as Invented *by* and Inventor *of* Technology

Physiologically, man in the normal use of technology (or his variously extended body) is perpetually modified by it and in turn finds ever new ways of modifying his technology. Man becomes, as it were, the sex organs of the machine world. (McLuhan 1964: 46)

As covered in section 2.2.2 and further explored in section 3.2.2, Heidegger's view of technology as a conditioning element of the human lifeworld – ranging from the highest achievements in scientific thought (such as physics) to mundane bodily tasks, such as hammering – might seem, at the first glance, to be an inverted account of how things really work. It is the "social shaping of technology" view reversed. Bernard Stiegler, in his work *Technics and Time: The Fault of Epimetheus,* probes deeper than Heidegger and shows with convincing arguments how the human has always been technological and how, in fact, technics should be seen as a constitutive element in the making of the prehistoric human. For Stiegler, the technical is not merely a tool or a machine: it constitutes the *invention* of the human. When our ancestors began using the flint as a knife, it performed an act of anticipation (of its usage)

[34] Latour himself warns against the project of "applying" the actor-network theory: "Because it's not a tool, or rather, because tools are never 'mere' tools ready to be applied: they always modify the goals you had in mind. That's what *actor* means" (Latour 2005: 143). Indeed, the aim is to describe movements, objects, actors but not to explain things in a certain theoretical framework. As Latour states: "if your description needs an explanation, it's not a good description… Only bad descriptions need explanation" (ibid: 147).

and it is in this externalisation of thought (through the flint) that we find the first form of technicity. It is thus through technics (as an externalisation of the mind) that the human experience of time is made possible. Technics make the memory of the past and the anticipation of the future available to human thought.

2.5.1 Epiphylogenesis – The Exterior Evolution of Technics

As a "process of exteriorization," technics is the pursuit of life by means other than life. (Stiegler 1998: 17)

Having defined technics as the condition of the human, Stiegler proscribes that the search for its essence should not be conducted in the biological or the transcendental domains, but rather in the *process* of exteriorisation. It is in this process where the living and the non-living, the biological and the technical, are coupled by a form of inscription that is inevitably technical, or rather *exterior* to the biological human body. The human is therefore a prosthetic animal that relies on technology external to its body – a fact which renders human history the history of technology. In order to study this, Stiegler introduces the concept of *epiphylogenesis* (technical evolution) distinguished from phylogenesis (biological evolution).[35] Epiphylogenesis is the condition that makes it possible for the human to be the only life-form whose non-biological history continues through time, as humans continually leave traces through history in the form of the things they produce. Not only is memory and culture already inscribed in the tools (such as pottery and weapons), but we have also developed external systems for memory such as writing, photography, phonography and cinematography.

For Stiegler, this tracing through time (anticipating the future and remembering the past) in the form of technics is always a memory aid, an *epiphylogenesis,* but also mnemotechnics (Stiegler 2009). The human is constituted through its exteriorisation of thought, a *prosthesis* in the form of tools. Thus, epiphylogenesis is a break with genetic evolution of the human, a break which constitutes its "invention."

> *Because it is affected with anticipation, because it is nothing but anticipation, a gesture is a gesture*; and there can be no gesture without tools and artificial memory, prosthetic, outside of the body, and constitutive of its world. There is no anticipation, no time outside of this passage outside, of this putting-outside-of-self and of this alienation of the human and its memory that "exteriorization" is. *The question is the very ambiguity of the word "exteriorization"* and the hierarchy or the chronological, logical and ontological preeminence that it immediately induces: if indeed one could speak of exteriorization, this would

[35] Epiphylogenesis is not simply a philosophical concept. Richard Dawkins in his book *The Extended Phenotype* also saw tools as an independent evolutionary process that interacts with the biological evolutionary process: "[A]n animal artefact, like any other phenotypic product whose variation is influenced by a gene, can be regarded as a phenotypic tool by which that gene could potentially *lever itself* into the next generation" (Dawkins 1982: 199, my italics).

> mean the presence of a preceding interiority. Now, this interiority is nothing outside of its exteriorization: the issue is therefore neither that of interiority nor that of exteriority – but that of an originary complex in which the two terms, far from being opposed, compose with one another (and by the same token are posed, in a single stroke, in a single movement). (Stiegler 1998: 152)

Stiegler here refers to the Derridean concept of *différance,* a neologism that connotes at the same time the fact that we are always dealing with deferral (temporal) as well as differences (spatial) in the technical object, or more precisely and generally in *all* writing (technology here seen as a form of epiphylogenetic writing). Writing and memory is thus always subject to the condition of *différance*. Through *différance*, Derrida deconstructs the boundaries between the outside and the inside, of signifiers and signifieds. (Derrida 1974).[36] Stiegler uses this idea to analyse three types of memory: the genetic memory of the species; the somatic memory of the nervous system (a memory that disappears when the individual dies); and technical memory, or culture, i.e., the memory that we are born/"thrown" into and makes our history possible. The last type of memory is what Stiegler calls epiphylogenetic memory. It is in technical memory that we find the origin of the human: through the process of externalisation the "interior" is instated in a reciprocal relationship with the "exterior." The interior depends on the exterior; it becomes the "originary supplement," which is in essence a mnemotechnical prosthesis for the human being. This dependency, or reciprocal relationship, is a transductive relationship, i.e., each element needs the other to exist. The human being is thus subject to two histories or two evolutionary processes: the biological and the technological.

2.5.2 Operating Technology: Technology as an Independent Force

Stiegler works with Bernard Gille's ideas of technological evolution and outlines how it can be discontinuous, due to the limitations of its endogenous and exogenous evolution. Technology can stall because of material limitations, for example when heavier and stronger trains could not be made as the iron in the rails was to weak (endogenous limitations). The invention of hardened steel in England solved this problem, but in the late 19th century France techno-economic protectionism resulted in import duties put on steel from England, thus hindering the possible progress in the rail industry (exogenous limitations) (ibid: 33). A technical system is only able to take form when the evolution of technical objects stabilise around a point of equilibrium where the endogenous and exogenous conditions are right. It is in this condition of equilibrium that the technical system concretisises (and, as Latour would put it, pointilises and becomes a "black box").

[36] Recent cognitive science and HCI also sees writing as a process of externalising thoughts. (Dix 2008). In this thesis, we aim to show how such externalisation depends upon the tools used for the externalisation: in the case of writing, for example with a pen and a paper or a word processor, but in the case of music an acoustic instrument or more complex (and as will be argued) epistemic digital tools.

> When a set of conditions is grouped into a system, a decision to evolve takes place. In other words, there is on the one hand progress *qua* the development of the consequences of a technological invention within a stable technical system, without obligatory crises, without brutal discontinuity – a development Gille calls "technological lines" – and on the other hand, progress as destabilization of the technical system, reconstitution around a new point of equilibrium, and the birth of a new technical system. New technical systems are born with the appearance of the limits of the preceding systems, owing to which progress is *essentially discontinuous.* (ibid: 33)

Figure 2:1 A Neanderthal flute found in Slovenia with an estimated age of 45.000 years. An early technology, but through the process of externalisation, music theory has now been inscribed into a material artefact and is therefore not dependent anymore upon cultural voice-use practices.

This does not mean that we have now fallen into the trap of technological determinism. Gille resolves to talk about "loose determinism," i.e., one cannot anticipate technological evolution a priori, but neither is it a random innovation (as it often seems). It is driven by a specific logic innate to the technology: "*The system's dynamic offers the possibility of invention, and this is what is essential to the concept of technical system: the choice of possibilities in which invention consists is made in a particular space and particular time according to the play of these constraints*, which are submitted in turn to external ones" (ibid: 35). The external constraints here are those of economy, science, society and culture. The logic of an invention is therefore not that of the inventor, but a specific techno-logic that inheres in the technical object itself. The human is merely the operator, the channel through which the techno-logic proceeds. It is from this perspective, of the evolution of technics, that Stiegler is able to define technics as an "inorganic organic being." Our culture has become increasingly interwoven with this external technical evolution; we are becoming ever more "techno-logical" in the sense that we are quicker to adapt to changing technological situations and more accustomed to techne's particular logic.[37] The human is an agent *in* technology, an operator and inventor of it, but one that is necessarily

[37] For example, as Gille observes, it took one hundred and two years from the discovery of the physics of photography until photography itself was developed (1727-1829). It took fifty six years for the telephone, thirty five for radio, twelve for television, six for the uranium bomb, and five for the transistor. "This reduction in delays is a result of what Weber, Marcuse and Habermas call 'rationalization.' Its price is a totally new relation between science and technics (and politics), established by way of the economy" (ibid: 40).

bound to its techno-logic; a logic that has evolved with so much force and accelerando in the last century that the Frankfurt School has called it *instrumental rationality*, a rather derogatory term for a mind-set that calculates, orders, and places nature and people as variables into the equations of economic value.[38]

2.5.3 Techno-logic and Techno-logy

Technology is therefore defined by its own techno-logic, through a complex system of technical elements that concretisise under the right conditions. When a technology gains momentum, it establishes itself and social practices around it. But simultaneously, by becoming concrete, it constrains and closes for other possibilities.

> The concretization of technical objects, their unification, limits the number of their types: the concrete and convergent technical object is standardized. The tendency to standardization, to the production of more and more integrated types, makes industrialization possible, and not the converse: it is because there is one or another tendency in the process of technical evolution in general that industry can appear, and not because industry appears that there is industrialization. (ibid: 72)

In this context one might analyse how the simple invention of the MIDI protocol managed to create the conditions of a strong hardware and software industry in the 1980s. The MIDI protocol was created by a group of a few commercial companies with the aim that different hardware could talk to each other. This group excluded many interested parties: companies and lay people alike who wanted to have their say in these important decisions. As a protocol it is extremely simple and functional, but highly reactionary in terms of the musical tradition of the time. It does not lend itself well to microtonal music, complex rhythmic structures and other musical idioms that were increasingly being explored at the time. MIDI was designed for simple processing, eight bit numbers, cheap chips that would receive and send data, and in the end became the lowest common denominator that the companies working on the protocol could agree upon (Théberge 1997: 93). The history of MIDI goes against the social constructivist ideas that see technology as purely of sociological origin, of how *we* define technology for our own use. Technology itself has an agenda, a logic that is not sociological but contains a certain auto-nomy; a logos, a rationality that is not theoretical or scientific, but empirical and practical, i.e., the products evolve through their concretisation of technical elements and thus become blackboxes that are not questioned any more. As Latour would put it, they translate from one repertoire to a more durable one.[39]

---

[38] We are, of course, reminded of Heidegger's term *enframing* here.

[39] "This definition [of transition] does *not* imply that the direction always goes from soft bodies to hard machines, but simply that it goes from a provisional, less reliable one to a longer-lasting, more faithful one" (Latour 1992: 256).

The technical object has its own logic, but it is a logic that we can never apprehend in its totality. Whatever we invent or create through technology becomes "reality only through the technical object's potential of inventiveness, in the process of concretization characterized by the fact that the human has no longer the inventive role but that of an operator. If he or she keeps the inventor's role, it is *qua* an actor listening to cues from the object itself, reading from the text of matter" (ibid: 75). This fits clearly with Akrich's (1992) idea of technology as a script. Stiegler defines the human working with the technical ensemble as an operator, and it is through creative reading of the technology that originality is possible:

> To draw further on the metaphor, the actor is not the author – and that is why existing technical objects are never thoroughly concrete; they are never consciously conceived and realized by the human from out of this "logic," which is strictly speaking empirical, experimental, and in a sense quasi-existential (it is the object's *mode of existence*), the sense, namely that this logic is revealed only in its realization, in the experience of the object itself, or, as it were, on stage, and not at the time of conception. In this move the logic of invention becomes in essence unpredictable, as in Maunoury; and this is why "the technical object is never totally known. (Stiegler 1998: 76)

Stiegler here provides an important observation where he shows that the logic of an invention is empirical/experiential and not preconceived. Here, he is calling for a new attitude in the studies of technics, which implies admitting that the technical dynamic precedes and imposes itself upon the social dynamic. "The tasks of a knowledge allowing for the articulation of a relation between the human and the technical ensemble are those of an analysis of the new dynamic schemes and an understanding of the *necessity* of an *advance* of the industrial technical dynamic upon other social aspects. … at stake is *doing technology as one does sociology or psychology*" (ibid: 67). This is a topic we will explore in chapter 4 and further in the succeeding chapters of this thesis. How does the technological ensemble (hardware, software, socio-technological context) define musical practices, and what are the technical traces that generate these ensembles? What factors (or equally "which actors") – technical and social – are determining in this context?

2.5.4 Transductions – Techno-social Dialectics

The prosthesis is not a mere extension of the human body; it is the constitution of this body qua "human" (the quotation marks belong to the constitution) (Stiegler 1998: 152)

Stiegler shows how technics founds human existence through its biological-technological couplings. There would be no memory or culture without technics and if we want to locate the source of humanity it would be in its use of artefacts: "The question then becomes: *where* is the memory of the stereotype kept, if not *in the material trace of the stereotype in which the preexisting tool itself consists, repeated, duplicated by its "maker" and guiding the latter much more than being guided by him or her?* In this sense, the archaic cortex and equipment are codetermined in a structural coupling of a particular sort" (Stiegler 1998: 158).

To think of technology in these terms of "instrumental maeutics" means to think *transductively.* Adrian Mackenzie (2002) has studied the dynamics of transduction and provides a framework that helps to understand the process of how technology comes to constitute our being-in-the-world and cultural evolution, but he also questions what is defined as the technical and what the social:

> A transductive approach promises a more nuanced grasp of how living and non-living processes differentiate and develop. It understands the emergence of a mode of unity without presuming underlying substance or identity. Every transduction is an individuation in process. It is a way something comes to be, an ontogenesis. Importantly, transduction refers not only to a process that occurs in physical, biological or technical ensembles as they individuate. It also occurs in and as thought. Thinking can be understood as an individuation of a thinking subject, not just something that someone who thinks does. To think transductively is to mediate between different orders, to place heterogeneous realities in contact, and to become something different. (Mackenzie 2002: 18)

This process of transduction can also be referred to as a techno-social dialectics, human-nonhuman translations, or shifting. It involves transferring energies from one domain to another. A human inscribes a technological object (such as Latour's speed bumps) and the technological object in turn defines social behaviour. What Stiegler provides us with is a strong philosophical argument for seeing technics as an originary prosthesis of the human, a constitutive element of the evolution of the species. Latour, on the other hand, provides us with a powerful analytical theory to analyse *how* this happens in practice, in the actual social context of the human and its tool use. For Latour it is plainly absurd to think of either technology or society as a defining factor over the other:

> [W]e can call *sociologism* the claim that, given the competence, pre-inscription, and circumscription of human users and authors, you can read out the scripts nonhuman actors have to play; and *technologism* the symmetric claim that, given the competence and pre-inscription of nonhuman actors you can easily read out and deduce the behavior prescribed to authors and users. From now on, these absurdities will, I hope, disappear from the scene, because the actors at any point may be human or nonhuman, and the displacement (or translation, or transcription) makes impossible the easy reading out of one repertoire into the next. The bizarre idea that society might be made up of human relations is a mirror image of the no less bizarre idea that techniques might be made up of nonhuman relations. (Latour 1992: 239)

Through this analysis of transduction, or displacement, or the techno-social dialectic as we might call it, we realise that behind each human or nonhuman actor there is a network of other human and nonhuman actors, thus rendering impossible the naïve idea of trying to isolate where one domain (the technological or the social) defines or determines the other.

2.5.5 Organic Inorganic Matter and the Problem of Human Agency

According to Simondon (1980), the technical evolution is amplified after the Industrial Revolution, where, inspired by cybernetics, a "becoming-organic" tendency appears in technical objects, i.e., they evolve into becoming evolutionary. Stiegler claims technics is neither inert and static nor is it living. "*It is organized inorganic matter that transforms itself in time as living matter transforms itself in its interaction with the milieu*. In addition, it becomes the interface through which the human *qua* living matter enters into relation with the milieu" (Stiegler 1998: 49). Humans have evolved *through* technology, it is the condition for the development of human culture. Analogous to the work of Ihde, technics is here seen as prosthesis, as an extension of the human, and a necessary part of it.

> The evolution of the 'prosthesis,' not itself living, by which the human is nonetheless defined as a living being, constitutes the reality of the human's evolution, as if, with it, the history of life were to continue by means other than life: this is the paradox of a living being characterized in its forms of life by the nonliving – or by the traces that its life leaves in the nonliving. (ibid: 50)

Leroi-Gourhan defines the interior and exterior milieus of the human, (the exterior being the nature and technology – the interior being the culture and the *use* of that technology) and questions the concept of genius or inventiveness in human culture. "[K]nowing whether a technical innovation is borrowed or properly invented appears here as almost secondary, since the innovation's adoption can only take place in 'an already favorable state of the interior milieu. The adoption can be considered an almost accessory trait, the important aspect being that the group is ready, in the absence of innovation, to invent or borrow'" (ibid: 58, Stiegler here quotes Leroi-Gourhan).

If technology has its own logic, its own mechanistic teleology, we might worry about human agency in this omnipresent and determining structure of technics. What is the role of the human when technical evolution stems completely from its own structure and the human is no longer the *intentional actor* in this dynamic, but merely its *operator*? (ibid: 66). The two perspectives of technology that we will work with in this thesis are Stiegler's theory of epiphylogenesis and the actor-network theory. These might seem incompatible at first glance as epiphylogenesis practically eliminates the agency of humans (instead of the genius inventors of technology we now get mere operators of technology) but the actor-network theory, although assigning much agency to nonhuman actors, still has room for human agency and inventiveness.

Yet, on a closer inspection, we see that these two theories are merely working from different perspectives. The philosophical theory (Stiegler) views everything as technics; language, writing, primitive and hi-tech tools are all part of the externalisation of the cognitive capacity of the human. As such technics is pre-social. However, it has a history which is the evolutionary history of technics and humanity in the same stroke. The social theory (Latour) places the dividing line between objects and humans differently. It resolves around the relationship between humans and nonhumans and shows how this focal point, the relationship itself, is where social practices emerge. Just as in the philosophical theory, nonhumans, or technics, have agency and determine social practices, but behind every nonhuman is a more complex network of humans and nonhumans. The philosopher places logos outside the human, but the sociologist acknowledges that it *can* still be within. What these theories have in common is the reluctance to see technology as the Other of humanity, thus avoiding maintenance of the good old distinctions of culture/nature, human/machine, mind/matter, etc./etc.. They have shown through the theories of deconstruction and *différance* (philosophy) and dynamic systems and network theory (sociology) that these binary distinctions have ceased to function.

It is here that a new form of knowledge is necessary to understand the essence of technology, that of the techno-logist.[40] It does not involve the view that the human is alienated by technology, such a view is not caused by the machine but by the misunderstanding of its nature and essence. In order to do techno-logy, we need to understand the *condition* of the human inside the "technical ensembles." Apart from Gille, Simondon and Stiegler, some other work has already been done in this field, most notably the work of Deleuze and Guattari on the "machinic phylum" (Deleuze & Guattari, 1987: 406). Recent work by Johnston (2008) further deals with the problems of how the fields of cognitive science and AI forgot to analyse the very thing that constitutes their own practice, namely the technical ensembles themselves.

2.6 Conclusion

Is it tekhnē *that arises from* logos*, or the reverse? Or rather, is it not that* logos *and* tekhnē *are modalities of the same being-outside-oneself? (Stiegler 1998: 193)*

This chapter has presented various approaches to the problem of understanding the relationship between technology and society or culture. Those have differed according to the fields in which the authors work, but what is common to all is the immense influence technology has on our ways of being in the world as thinkers, scientists, workers, artists and people going about their daily tasks. The different viewpoints regard our relationship to technology and *how* it influences us: to what degree does technology determine society and society shape technology? Here the

[40] "Techno-logy" should here be understood in the same morphology as "psychology".

actor-network theory proposes itself as a powerful tool to analyse the agency of technological objects as nonhuman actors. It also shows how humans construct technological systems by creating ensembles; networks that eventually become actors themselves. For this thesis, the actor-network theory of technology as script, and the analysis of the way humans can subscribe to or reject that script through various programs and antiprograms are useful tools.

If we improvise with John Law's actor-network diagnosis of science, and translate it into the sphere of music, we see that music is the result of a process of "heterogeneous engineering" in which bits and pieces of the social, the technical, the conceptual and the textual/music theoretical are fitted together, and so converted (or "translated") into a set of equally heterogeneous musical products (Law 1992: 2). The musical products we have in mind here are everything from musical software, hardware, distribution media, record labels, sound systems, radio programs, magazines, night clubs, musical styles, education systems, hardware shops and magazines, etc. These are all results of the enrolment of actors who produce the musical culture.

This chapter also explored how recent Continental thought has come to think of technology as something that constitutes the human not only as something external, as some tool, but a necessary supplement or prosthesis to the human being that has shaped its evolution from the beginning. Bernard Stiegler defines technics as organised inorganic matter that exists outside of the human body but in a *transductive* relationship, which means that the relationship between humans and technology is a symbiotic one. Neither would exist without the other, prompting him to state that "[t]echnics is the pursuit of life by means other than life" (Stiegler 1998: 17). Technics move faster than human culture and constitute a specific form of temporalisation where there is delay and advance in different ways, as explained by Hughes' "technological momentum." But most important for this thesis is the analysis of how we, as humans, exteriorise our thoughts through technological systems, thus expanding our bodily and cognitive systems into the realm of technological artefacts.

> It is by freeing itself from genetic inscription that memory at once pursues the process of liberation and inscribes thereupon the mark of a rupture – on stones, walls, books, machines, madeleines, and all forms of supports, from the tattooed body itself to instrumentalized genetic memories, dis-organized, made inert [*inertifiées*] as it were, then reorganized, manipulated, stored, rationalized, and exploited by the life industries named "biotechnologies," including holographic memories that the information-processing industry is planning. (ibid: 169)

Stiegler refers here to Proust's *Remembrance of Things Past,* of how the narrator's tasting of a madeleine resulted in a transposition back in time through an involuntary memory process that made him relive various periods of his life as if the cake was the container of those memories. Post hoc, we are also reminded of the film *Memento* where the main character suffers from long-term memory dysfunction, forcing him to tattoo vital information on his body. This case exemplifies how the external environment, in this instance the protagonist's body, serves as an extension of cognitive capacity in the most brutal way.

These two foundations – actor-network theory and epiphylogenesis – establish the foundations for the analysis of musical tools later in this thesis. They provide, respectively, flexible methodologies to analyse the ontogenesis of technological artefacts as they appear and integrate into human culture on the one hand, and a philosophical understanding of technology and its relationship to human society on the other. The quote above points directly to the human body, its embodiment, cognitive externalisation and actual embeddedness in the world. It observes the cognitive function of a simple madeleine. This is where we are now heading: from the abstract discourse of technology to its phenomenological relationship with the human body.

Chapter 3

Embodiment: The Body and Technology

This chapter explores the effect digital technologies have on musical performance. A general phenomenological account is given, introducing Don Ihde's phenomenology of technological instruments. This phenomenological delineation leads to the study of embodied learning, habituation, and of the body as a space for writing. This is primarily done through the theoretical scope of two distinct, but related, theories in cognitive science: the theories of the extended mind and enactivism respectively. This chapter studies technological inscriptions in relation to embodiment.

3.1 Introduction – A New Variable in Cognitive Science: the Body

With the advent of computers and developments in psychology in the latter part of the 20th century, cognitivism became a driving force in the philosophy of mind and AI. Cognitivism is not one complete theory but a conglomeration of various theories that have in common the belief in symbolic representations, i.e., the notion that internal representations and inferential processes are characteristic workings of the mind at both high level thought processes and lower level perception. In cognitivism, studies of perception, memory, symbolic processes, attention, action planning and execution are all approached with the belief that the cognitive system consists of a world that is given to the subject through its input devices (the sensory organs) processed as symbolic data in the brain and then output through the subject's motor organs. Representation, formalism and rule-based transformation is all what is needed to simulate human intelligence. This view has therefore often rightly been called "computationalism" or "representationalism" (Putnam 1963; Fodor 1975), rendering the computer an ideal and natural metaphor for describing the workings of the mind as it has input devices, the internal machinery of computational or symbolic logic, and output devices. Consequently, as intelligence (animal or human) was seen to resemble computation so much, it became re-defined as the task of computing symbolic representations.

We find in computationalism a specific kind of dualism, not that of mind and body, but of hardware and software. Dualistic thought can be traced back to Parmenides and Plato, but more specifically, and in the form we best know it, to Descartes who saw the body as a machine (*res extensa*) and the soul as an independent matter (*res cogitans*), i.e., they were of different substances. The cognitivist approach in Artificial Intelligence – often called GOFAI (Good Old Fashioned Artificial Intelligence) – embodies this view. Here, hardware is seen as the material foundation for software, which in turn is the system of symbolic representations controlling the hardware. It claims that representation is a distinct, identifiable inner state or process, a symbol, "whose systematic or functional role is to stand in for specific features or states of affairs" (Clark 1996: 43). It is therefore the *form* of the symbol, not its semantic content, that is the basis of all cognitive processing of the world. The symbol gets its semantic meaning from the syntactic relationships between it and the other symbols in the system. This processing is necessarily confined to strictly rule-based systems. Formal rules are the mechanism that transforms one cognitive state into another.

This dualistic thinking has had its influence on the fields of computer design and human-computer interaction. For decades the main focus has been on screen-based software where the "operator" of the software uses a keyboard and a mouse for input, the software does the processing and output is typically displayed on a monitor. This is at the cost of a more holistic design, where various cognitive features such as those represented in gestalt theory, auditory cognition, and haptics can be utilised for a more integrated and embodied relationship with the tool. The field of digital music technology is still suffering from the cognitivist approach to human-machine design, although recent developments are showing more integrated understanding of the human as an embodied being whose thinking cannot be reduced to formal rules of symbolic representations.[41]

In the last decades various theories of cognition have emerged that attack the disembodied approach of cognitivism. These theories have had strong impact upon the field of artificial intelligence, psychology and human-computer interaction (e.g., Paiva 2000). The most prominent theories include *Connectionism*: where the idea is to build artificial neural networks that perform cognitive tasks through non-representational content (McClelland et al. 1986); *Enactivism*: which claims that the whole body and the environment becomes part of the cognitive function (Varela et al. 1991); *Dynamic Systems*: a non-representational theory claiming that cognition (and consciousness) arises as epiphenomena of the process of being in the world (Brooks 1991); *Situated Cognition*: a theory of knowledge acquisition as situated, being in part a product of the activity, context, and culture in which it is developed and used (Brown et al. 1989); *Situated Action*: an emphasis on the constitutional context of all action as emergence (Suchman 1987); *Activity Theory*: where the focus is on human tool use and the cultural and technological mediation of human activity (Bertelsen & Bødker 2003); *Embodied Cognition*: where thinking and acting is seen as one process, emphasising our situatedness, and claiming that our cognitive system emerges from interaction with the world (Anderson 2003); *Distributed Cognition*: cognition is seen here as an interaction between human and artefacts, and emphasis is placed on the social nature of human existence (Hutchins 1995).

---

[41] To be fair, much of this lack of multisensory interaction modalities has been due to insufficient computational power of our systems, which has made research and development difficult and expensive.

It is outside the scope of this thesis to explore how these theories, many of which overlap to a great extent, differ from traditional cognitivism in terms of representational content, symbolic processing and rule-based functionalism. However, what they all have in common is a strong emphasis on the body, the environment, culture, context, and learning as an emergent property through the agent's interaction with the environment.[42] The value of these non-symbolic theories (here used as general term) of cognition in the context of this thesis lies precisely in the emphasis on the body, environment and cultural context. In order gain an understanding of the difference in playing acoustic and digital instruments, resulting in the conception of *virtual embodiment*, a survey of some key aspects of these theories of embodied and situational cognition has to be conducted. This chapter starts with a journey into phenomenology as a response to certain dualistic problems of traditional philosophy, exploring how this philosophy has influenced cognitive science. Andy Clark's thesis of the extended mind is presented as a necessary preparation for the elucidation of the concept of *epistemic tools* that will be introduced in the next chapter. Furthermore, for the purposes of understanding musical embodiment, the enactive view by Varela, Thompson and Rosch, as developed in their book *The Embodied Mind*, is highly relevant. This chapter also explores how critical discourse, such as that of Hayles and Haraway, subscribes to the enactive view, albeit with an alternative vocabulary. These theories pave the way for the study of Dreyfus and Dreyfus' ideas of non-representational learning, and further Csikszentmihalyi's concept of *flow*, which will be important in our analysis of musical instruments and performance.

3.2 Phenomenology: A Philosophy Re-discovering the Body

3.2.1 Husserl and the Lifeworld

In the latter half of the 19th century, Franz Brentano (1838-1917) developed his theory of *intentionality*, i.e., that mental content is always "about" some object (this object can be a mental object and does not have to be a physical thing). Mental phenomena are always characterised by this "aboutness" and thus distinct from physical phenomena. A student of Brentano, Edmund Husserl (1859-1938) took up the idea of intentionality and developed a methodology of transcendental phenomenology, where the method of "bracketing" (*epoché*) is used to gain access to the real essence of things. By bracketing out the contingent properties of an intentional object, one can derive at its essence as *a priori* knowledge, i.e., the object as it appears to us before we mentally constitute them. What we bracket out from the object would be metaphysical, cultural, historical and scientific prejudice. The methodology of bracketing is therefore not an empirical technique, but rather a method of deductive a priori reasoning, which

[42] And note that we now talk about the 'agent' rather than the 'subject'. Johnston observes that the term's increased usage coincides with the break of representation in contemporary philosophy, "since a representation always implies a subject" (Johnston 2002: 485).

has been defined as the method of *phenomenological reduction*. Bracketing should be guided by what is given by pure experience and pointed at the *givenness* or *appearance* of reality. The focus shifts from what things *are* to the ways in which things are *given*. However, the idea is not only to study how things are given to consciousness, but also the subjective nature of consciousness. Husserlian phenomenology has therefore been portrayed as the science of the first person perspective, of introspection.

Husserl's idea of intentionality has been influential. As opposed to the positivist or behaviouralist trends of his time, *experience* is central to Husserl's theories; as something that is not reducible to scientific observations or quantitative data. Through the concept of lifeworld (*Lebenswelt*) Husserl develops a theory that claims that it is possible to gain access to the already given and unreflected nature of things. The lifeworld is the subject's beliefs about the world but also the (a priori) structure that makes human communication and intersubjective interpretation possible. This dichotomy between pre-scientific world and scientific world, although understandable and useful in some respects, can be problematic in Husserl. It deals with similar ideas as Kuhn (1962) with the theory of paradigm shifts or Michel Foucault's (1970) theories of epistemic eras, but for Husserl it is defined simply as the world of pre-technology. Husserl points at Galileo as a particular pivotal point in time where Galileo, the discoverer

> of physics, or physical nature, is at once a discovering and a concealing genius. He discovers mathematical nature, the methodical idea, he blazes the trail for the infinite number of physical discoveries and discoverers. [But] immediately with Galileo, then begins the surreptitious substitution of idealized nature for the prescientifically intuited nature. (Husserl 1970: 52)

Husserl thus builds up a romantic mythological construction of a pre-scientific world that is essentially different to the one that comes after Galileo. Every world-view must necessarily have its own ways of seeing the world through both conceptual and technological toolkits, and after Galileo our culture gains a new mode of understanding the world. In a more recent work – on the archaeology of knowledge – Foucault traces the changes in epistemic eras, but refrains from the romanticism of Husserl, demonstrating that every epistemic era has its own tools, both physical and theoretical, to engage with the nature of reality. For Foucault such archaeology has to be an enquiry "whose aim is to rediscover on what basis knowledge and theory became possible; within what space of order knowledge is constituted" (Foucault 1970: xxi).[43] Husserl's focus on the context of human existence is important as it was later to be picked up by many

---

[43] In a later work Foucault (1980) redefines the episteme as: "the strategic apparatus which permits of separating out from among all the statements which are possible those that will be acceptable within, I won't say a scientific theory, but a field of scientificity, and which it is possible to say are true or false. The episteme is the 'apparatus' which makes possible the separation, not of the true from the false, but of what may from what may not be characterised as scientific" (ibid: 197).

important thinkers such as Heidegger, Merleau-Ponty, and Sartre, although they tend to criticise his transcendental tendencies and lack of historicity.

3.2.2 Heidegger

A student of Husserl, Martin Heidegger, rejects the transcendental tendencies in his teacher's work, criticising his ideas of pure essences (or noema) and similarly for ignoring the importance of lived context. Heidegger points to the historicity of our condition, stressing that our activities are always situated in the world; in time and space, or rather, as developed later by Lefebvre, in historical time and place (Lefebvre 1991). We should therefore not bracket out the world but rather look at the context in which it is given to us; what kind of relations constitute the objects we confront. All interpretation is defined by the fore-knowledge that is comprised by the subject's lifeworld. Being and its relation to the world is, for Heidegger, the fundamental question of philosophy (Heidegger 1962: 24). This relationship with the world happens in a historical context where we are surrounded by "given" technologies that we are born (or rather "thrown") into. Heidegger thus rejects Husserl's transcendental or dualism, and focuses on the human existence as necessarily being part of the world and constituted by it.

Thus the focus of Heidegger's philosophy turns from the ego's cognition of the world as with Husserl, to the way we *exist* in the world as being-in-the-world (*Dasein*). The world always precedes us and we are thrown into its ordered complexity. The focus is always that of the practical, of the subject interacting with the world through historical equipment that is part of the technological totality.[44]

> As the Being of something ready-to-hand, an involvement is itself discovered only on the basis of the prior discovery of a totality of involvements…. In this totality of involvements which has been discovered beforehand, there lurks an ontological relationship with the world (Heidegger 1962: 118)

Human perception and apprehension becomes an *existential understanding*, an un-theoretical stance, not *towards* the world but *in* the world. This understanding is therefore always finite and historical, conditioned by the "thrownness" of the subject. As we will see, the value of Heidegger's work for human-machine interaction, and his principal influence on contemporary cognitive science, lies mainly in his way of analysing the human as always relating to the world through embodied practical activities, such as hammering.

[44] We have already seen how this observation translates into the idea of *epiphylogenesis* in the work of Bernard Stiegler.

3.2.2.1 The Hands and the Hammer

In Being and Time, Heidegger illustrates how the world is always given to us *before* we analyse it. He takes the example of the hammer. The hammer is characterised by its serviceability; we do not normally have a theoretical relationship towards it. There is no need to analyse the hammer, its meaning for us is revealed in our use of it. The hammer is ready-to-hand, i.e., something that we can reach out for and perform a specific task with. It is an instrument or equipment that is defined by its relationship to other instruments and it is seldom analysed on its own. However, when such equipment breaks down or malfunctions, its mode of being is no longer something *primordial* or natural for us: it becomes present-at-hand and our relation to it changes. We now become aware of the functionality of the tool, its materiality, history and purpose. It stands against other equipment and it is here that Dasein's fundamental being is defined as *care.* When things break down, our care becomes visible, a care that is obviously also manifested as a care towards other people (ibid: 102).

Heidegger's philosophy focuses more on the practical nature of human existence, rather than on the epistemological questions that philosophy has traditionally been concerned with. We are in a world full of technological objects that we use without thinking consciously about them. It is only when they malfunction that they begin to face us as present-at-hand objects and our relation to them changes to a theoretical or "scientific" relationship.

> Taken strictly, there 'is' no such thing as *an* equipment. To the Being of any equipment there always belongs a totality of equipment, in which it can be this equipment that it is. Equipment is essentially 'something in-order-to…' A totality of equipment is constituted by various ways of the 'in-order-to', such as serviceability, conductiveness, usability, manipulability. (ibid: 97)

The equipment is the means of the experience, not its object: the tool *withdraws* and the work becomes the focus or terminus of the experience. It is *through* the tool that our attention is pointed to the task at hand: "its readiness-to-hand… must, as it were, withdraw in order to be ready-to-hand quite authentically" (ibid: 99). The tool disappears, but it also discloses the world for us. It opens up new ways of experiencing and thinking about the world which, without the tool, would be impossible (ibid: 101). It is *through* the tool that the human understands the world in which the work or the results of the act appears. As we have already seen, there is a totality of equipment already at play and from which the human selects a tool to interact with the world. Reversing the Husserlian order by putting the emphasis on praxis, Heidegger is able to claim that the equipment as ready-at-hand *precedes* the equipment as present-at-hand, or the theoretical understanding of the tool as it can be analysed. Use precedes analysis.[45]

---

[45] This emphasis of the instrument as the *means* through which we experience the world is criticised by Stiegler who defines the human condition as an *instrumental condition* necessarily constituted by technics. For Stiegler, Heidegger is still guilty of a dualism between thought and skill:

Heidegger's influence upon thinkers like Bertrand Gille, Andre Simondon and Bernard Stiegler is obvious. The human is technological by nature, it is born/thrown into the world of technics which in turn constitutes the human. However, his influence is not only on Continental philosophy of technology. Heidegger's strong analysis of the embedded nature of human existence through the analysis of tool use has also become influential in contemporary cognitive science. Wheeler points out that this is not because cognitive science has become a Heideggeriean research programme, but rather that his ideas are extremely apt to explain certain empirical findings that cognitive science is currently coming up with (Wheeler 2007: 19).

3.2.3 Merleau-Ponty

The body is our general medium of having a world (Merleau-Ponty 2002: 169)

Maurice Merleau-Ponty read Husserl and Heidegger closely, but his own work focussed more on the body itself and the aesthetic and perceptual experience we gain through our body as existing in the world. The body is the condition of life. It is where action happens: "We must therefore avoid saying that our body is *in* space or *in* time. It *inhabits* space and time…. I am not in space and time, nor do I conceive space and time; I belong to them, my body combines with them and includes them" (ibid: 161). Merleau-Ponty demonstrates that our bodily experience of movement is not a type of knowledge. It is a primary and originary relation with the world that comes prior to any knowledge. The body *has* its world and it does so without having to make use of any symbolic or objectifying functions.

Merleau-Ponty thus rejects the Husserlian dualism or the dualism of Sartre (between consciousness and object) and sees mental life as thoroughly corporeal, always based on the body. There is no denying that the body can be treated as an object, for example in medical science or in the contemporary plastico-cosmetic or porn industries, but for Merleau-Ponty this

---

> Heidegger's thought is fundamentally still inscribed in the traditional opposition between *tekhnē* and *logos*. If he denounced, well after *Being and Time*, [Stiegler here referring to *The Question Concerning Technology*] and in another vocabulary, analyses of technics that are conducted in terms of the categories 'end' and 'means,' it was in order to uncover an *instrumental* conception of technics, an analysis in which he does not appear to put in question the determination itself of an instrument *as* a means. The metaphysical illusion from Plato onward that turns language into a means through which humans express themselves, rather than its being located as the site of their very constitution, is abundantly criticized by Heidegger. Yet it is the same error that induces consideration of an instrument as means. (Stiegler 1998: 205)

Stiegler claims that there is no "point in looking for the noninstrumental aspect to language; nothing of the kind exists" (ibid: 206). Heidegger sees Dasein as naturally engaging with a system of equipment in its actions of daily life, a fact that makes Stiegler claim that Dasein is essentially prosthetic: it cannot gain access to its past nor anticipate the future without consulting what is outside itself, i.e., technics (ibid: 234).

is precisely because we are presupposing the human body as being itself a subject that is in a constant dialogue with the world and other subjects. The dualistic tendencies of our language are derived from millennia of categorical separation between mind and body. An aspect of Merlau-Ponty's project is explore how this dualistic language rises from the pre-linguistic and pre-conscious level of the body. The objective body is merely an image or projection of the phenomenal body that we are: we do not have a body, we *are* one.

Various technologies can extend our body or our sense of the space we occupy in the world. Merleau-Ponty gives an example of a woman with a feathered hat who is able to walk through low doors without breaking the feather as if the feather has now become part of her body. Another example is how we do not need to calculate the width of a gate when driving car through it, as the sensation of the car has become habitual and we feel the boundaries of the car as it was our own body. But perhaps the best example he gives is how the blind man's stick becomes a prosthetic extension of the body as if he has gained a new sensory organ:

> The blind man's stick has ceased to be an object for him and is no longer perceived for itself; its point has become an area of sensitivity, extending the scope and active radius of touch and provoking a parallel to sight. In the exploration of things, the length of the stick does not enter expressly as a middle term: the blind man is rather aware of it through the position of objects than the position of objects through it. The position of things is immediately given through the extent of the reach which carries him to it, which comprises, besides the arm's reach, the stick's range of action. (Merleau-Ponty 1962: 143)

This analysis supplements Heidegger's idea of the hammer as the tool *through* which we work. The hammer and the stick disappear in use and we gain access to the world through these instruments.

> The points in space do not stand out as objective positions in relation to the objective position occupied by our body; they mark, in our vicinity, the varying range of our aims and our gestures. To get used to a hat, a car or a stick is to be transplanted into them, or conversely, to incorporate them into the bulk of our body. Habit expresses our power of dilating our being-in-the-world, or changing our existence by appropriating fresh instruments. (Merleau-Ponty 2002: 166)

It is clear that for Merleau-Ponty we incorporate the tools we interface the world with; they become part of our cognitive mechanism. We also see here a certain rejection of the cognitive science portrayal of the human as constantly building mental models of the world, models that we test through our engagement with it. Indeed, the idea here is one of bodily *habituation* in the world, from which our conscious post-reflections of it can be abstracted (when we objectify the world or, in Heidegger's terms, take a "theoretical stance" towards it). Through acting, the body "catches" or "grasps" the significance of the movements and incorporates that value for later expression. In a long but illustrative section, and one highly relevant to this thesis, Merleau-

Ponty shows how the movements of a trained musician are not those of intellectual analysis, calculated gestures or reconstruction, but more that of exerting habitual motor memory:

> The example of instrumentalists shows even better how habit has its abode neither in thought nor in the objective body, but in *the body as mediator of the world*. It is known that an experienced organist is capable of playing an organ which he does not know, which has more or fewer manuals, and stops differently arranged, compared with those on the instrument he is used to playing. He needs only an hour's practice to be ready to perform his programme. Such a short preparation rules out the supposition that new conditioned reflexes have been substituted for the existing sets, except where both form a system and the change is all-embracing, which takes us *away from the mechanistic theory*, since in that case the reactions are mediated by a comprehensive grasp of the instrument. Are we to maintain that the organist analyzes the organ, that he conjures up and retains a representation of the stops, pedals and manuals and their relation to each other in space? But during the short rehearsal preceding the concert, he does not act like a person about to draw up a plan. He sits on the seat, works the pedals, pulls out the stops, gets the measure of the instrument with his body, incorporates within himself the relevant directions and dimensions, settles into the organ as one settles into a house. *He does not learn objective spatial positions for each stop and pedal, nor does he commit them to 'memory'.* During the rehearsal, as during the performance, the stops, the pedals and manuals are given to him as nothing more than possibilities of achieving certain emotional or musical values, and their positions are simply the places through which this value appears in the world. Between the musical essence of the piece as it is shown in the score and the notes which actually sound round the organ, so *direct a relation is established that the organist's body and his instrument are merely the medium* of this relationship. Henceforth the music exists by itself and through it all the rest exists. There is here no place for any 'memory' of the position of the stops, and it is not in objective space that the organist in fact is playing. In reality his movements during rehearsal are consecratory gestures: they draw affective vectors, discover emotional sources, and create a space of expressiveness as the movements of the augur delimit the *templum*. (ibid: 167, my italics except the last word)

In many regards this view is similar to the idea of tacit knowledge developed by Michael Polanyi. Tacit knowledge is gained when we incorporate a new skill in our body "so that we come to dwell in it" (Polanyi 1966: 16), which also relates to the Wittgensteinian understanding of language's meaning as its *use,* as something we have learned through incorporating its rules (Wittgenstein 1968). Merleau-Ponty does in fact come up with similar idea of language and meaning as something that is inhabited by the body through its use. We do not think before we speak, our speech *is* our thought "A thought limited to existing for itself, independently of the constraints of speech and communication, would no sooner appear than it would sink into the unconscious, which means that it would not exist even for itself" (ibid: 206). Thought is not some internal thing that exists independently of the world or of language. The illusion of inner life appears when we think of our thoughts as something that came *before* the internal language, the thought itself. "In reality this supposed silence is alive with words, this inner life is an inner language" (ibid: 213). Here we find a view that resonates with Stiegler's view of the externalisation, or prosthesis, as a necessary element of the human. Language as technology is already an externalisation, even though we conceive of it as "inner language;" a view that will be explored in section 3.3.2.3 in Andy Clark's elaboration on cognition as something that does not simply take place "inside the head," but can extend by various means out to the body and the environment.

3.2.4 Instrumental Intentionalities: The Phenomenology of Tools

Don Ihde is arguably one of the most prominent philosophers of technology of the last few decades. His background is in Continental thought, particularly in phenomenology, but he has also contributed to the philosophy of science and scientific tools. Ihde arrived at his research project through the phenomenologically-based insight that human perception can be embodied *through* instrumentation of all kinds. Heidegger's analysis of the hammer or Merleau-Ponty's accounts of the feather in the hat or the blind man's stick become for Ihde *embodiment relations* to the world. However, Ihde rejects the distinction that traditional phenomenology has drawn between the *lifeworld* and the world of science. In fact scientific tools, such as the telescope or the microscope, are nothing but a prosthetic extensions of the human perceptual system that redefine the object of our study through an altered embodiment relation to the world. Technologies thus provide novel conditions for our thinking, and following Heidegger, Ihde reverses the idea that technology is "applied science." Technology, or *techne,* precedes science and in fact science can be seen as the "tool" or "instrument" of technology.

Ihde's focus is upon technological extensions of the human body. Like James Gibson who sees the tool as a natural extension of the human, stating that "the boundary between the animal and the environment is not fixed at the surface of the skin but can shift" (Gibson 1979: 41), Ihde explores the modalities in which the human extends its sensory and intellectual capacity. He defines various modes of relationships we establish with the world, such as embodiment relations, hermeneutic relations and alterity relations.

3.2.4.1 Embodiment relations

Examples such as drawing with a pencil or a chalk, driving a car or walking with a blind man's stick are all prosthetic relations with the world through what Ihde calls *transparency relations* or rather *embodiment relations* with the world. The better the tool, the more transparency there is. The agent "feels" the instrument or the machine, which becomes an extension of the agent's embodiment. The dentist's probe is a good example. Through a sensitive probe, the dentist is able to feel the surface of her patient's teeth better than with her finger. The sensation of the teeth is *amplified* with the probe, but simultaneously *reduced,* as other parameters, such as the warmth of the teeth or the wetness of the mouth, are not felt. In all these cases of embodiment relations, the final *terminus* of experience is the world itself as experienced through the machine interface. It is important to note here that all technology that we take into use is *non-neutral*, i.e., there is always a transformation of experience that happens through their use. This transformation is simultaneously a revealing and concealing process, which makes an awareness of the tool's transformative power upon our perception and thought an important issue in science, culture and, in our case, musical production.

The aim in the design of technologies that provide embodiment relations to the world is a *total transparency* of the instrument, for the disappearance of the instrument and for the technology to become part of the agent. The aim is to amplify the world or the body through the technology, but all amplification comes at the cost of reduction as instruments transform the experience. For Ihde, this fact is the source of both utopian and dystopian interpretations of technology.

> Instrumentation in the knowledge activities, notably science, is the gradual extension of perception into new realms. The desire is to see, but seeing is seeing through instrumentation. Negatively, the desire for pure transparency is the wish to escape the limitation of the material technology. It is a platonism returned in a new form, the desire to escape the newly extended body of technological engagement. In the wish there remains the contradiction: the user both wants and does not want the technology. The user wants what the technology gives but does not want the limits, the transformations that a technologically extended body implies. (Ihde 1990: 75)

All successful new technologies provide amplified, simplified, or ergonomically new ways of relating to the world. People are fascinated by the power of the technology and the finesse in perception or sophistication in action that it brings. But while the excitement is about the new ways of *revealing* or *performing*, what is often forgotten is what technology conceals through its use. This non-neutrality can be defined as *latent telics,* i.e., a technological trajectory that comes with technological thought where the desire for perceptual or gestural amplification transforms the world to a degree that it is not the same world anymore.

3.2.4.2 Hermeneutic Relations

Another phenomenological relation exists where the final terminus is not the world, but the machine itself as a representation of the world. This is often the case with complex engineering systems such as control boards in a factory or even music software simulating the physical studio. Ihde defines this relationship with the machine-as-representation as a *hermeneutic relation*. We need to interpret the state of the world through the interface of the machine. A good example is the atomic microscope. The microscope forces us to interpret what is happening on the "the other side" of the machine (Ihde 1979: 12). The machine becomes the "other," something that we are decoupled from and have to reflect upon. As opposed to the embodied relationship with the world through some technology, we now have a detached relationship with a machine (involving interpretation and or knowledge of its functionality), such that one can understand how it represents (as in the microscope) or simulates (as in a digital system) the world. The hermeneutic relationship also occurs when we have to interpret machine representations of the world, such as with infrared photographs. The viewer "must be able to 'read' the photo with regard to its significance and this reading, in turn, is informed by the 'exegetical expertise' which arises within the scientific community. The farther the continuum and the more extreme the variants, the more text-like the resultant reproduction becomes" (Ihde 1979: 34).

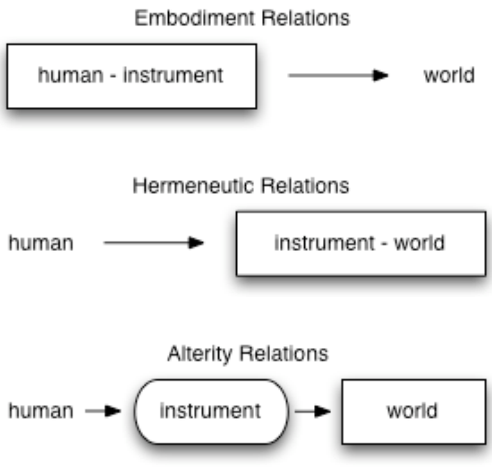

Figure 3:1 – A diagram illustrating the different human-world relationships.

The design of machines are important in both embodiment and hermeneutic relations, but the cultural element of design is stronger in hermeneutic relations. Interface design is a tradition based on cultural contingencies and traditions. For example, the screen-based interface tries to *refer* to the terminus (the task at hand or the state of the world) through an isomorphic representation in various possible forms, such as a graph, iconic symbol or auditory display. Here the aim is to provide a *representational transparency*, which of course has to be learned and understood. It only becomes transparent through repeated usage. As opposed to the direct embodied use of technology as prosthesis of the body, technology here becomes hermeneutic, a subject of cultural interpretation. It is the technology itself that becomes the focus of perception, not the world that is interacted with. In representational technologies, such as text or software we are dealing with a different mode of perception where "the referential transparency is distinctively different from technologically embodied perception. *Textual transparency is hermeneutic transparency, not perceptual transparency*" (Ihde 1990: 82). Both embodied and hermeneutic technologies have to be trained. In the case of hermeneutic technologies, the training involves some praxis incorporation (what we will call habituation) of the interface. The user or operator can thus *see* (such as the control board in a nuclear plant), *hear* (an experienced driver can hear a fault in the engine), *smell* (the baker or the cook), *feel* (Braille reader or the teleoperation of telepresence technologies) or *taste* (for example when checking the energy level of a 9V battery) the state of the world through these devices. But this is an incorporated knowledge that has to be interpreted.

3.2.4.3 Alterity Relations

In *Technology and the Lifeworld* Ihde (1990) adds another relationship to technology, which he calls "alterity relations." This is where the technology becomes the "other" or the quasi-other, something with which we interact, not in order to gain amplified access to the world, but for its own sake. Technology has become a thing on its own that has its own internal representations and structure. For Ihde, these relations can be placed on a continuum where embodiment relations are on the one end and alterity relations on the other:

> At the one extreme lie those relations that approximate technologies to a quasi-me (embodiment relations). Those technologies that I can so take into my experience that through their semi-transparency they allow the world to be made immediate thus enter into the existential relation which constitutes my self. At the other extreme of the continuum lie alterity relations in which the technology becomes quasi-other, or technology "as" other *to* which I relate. Between lies the relation with technologies that both mediate and yet also fulfil my perceptual and bodily relation with technologies, hermeneutic relations. (Ihde 1990: 107)

Contemporary human-machine interfaces can be placed on various locations on this continuous scale. They do in fact change position according to how the attention of the user oscillates from being immersed in the technology and performing through it (playing music, drawing, writing), to applying algorithms through graphical user interfaces or manipulating digital data, to allowing the software to perform on its own where the user becomes more a conductor or director than performer. Part II of this thesis analyses these different modes more thoroughly, with a special focus on digital musical instruments and their difference from acoustic or analogue musical devices.

3.2.4.4 Background Relations

Idhe gives an account of a yet another human-world relationship which he calls *background relations.* Here the actor is interacting with the world in small discrete steps, such as placing a slice of bread into a toaster, minding some other business (such as putting the coffee machine on) whilst waiting for the toast to pop up. Ihde refers to this mode of life as 'technosphere', a world where machines are automatically performing tasks for us and we only need to attend to them in very limited time periods during their active states. This is different from the three aforementioned relation modalities, where the technology or the task is focal. In background relations, the technology often performs automatically, needing only initial setup and then small maintenance tasks. Much music software has this property; I will propose later that in fact we find all these relations with the machine when working with computer music software. The interesting thing is how rapidly we oscillate between these states since a musical performance is a realtime performance where time (for example considering latency or formal structure) is of an immense importance.

3.2.4.5 Conclusion

Figure 3:1 represents the different instrumental relationships with the world. Intentionality is transformed in different ways in these modes. These intentional 'arcs' or trajectories are not pure and simple relationships that show symmetrical properties. In embodiment relations, the ideal situation is one where the instrument "withdraws" from the world and the task becomes the main focus. Here, the body is extended by transparent technology, such as with eyeglasses or a blind man's stick and we could define the interface as being part of the human, as its prosthesis. On the other hand, in hermeneutic relationships, the instrument becomes part of the focus, it is something external, something that has to be interpreted in order to understand what is being represented of the world through its representational modality. A microscope or a Braille terminal are good examples. Here the interface is part of the technological artefact and it has to be interpreted, thus hermeneutic. In alterity relations, the technology becomes the "other" and the focus on the world is secondary to the field of the artefact itself.

Although not a technological determinist, Ihde points out that there are latent *telic inclinations* that present themselves through the use of a chosen instrument. A text written with a pen will be different than a text written on a computer. Each of these technologies have different inclinations for utilisation, a different transformation. The rhythm of the instruments are different as they lend themselves for different *styles* of writing (ibid: 42). This telic inclination of the instruments provides a path that is one of the least resistance, thus influencing the writer. In the same manner, the path of least resistance in hermeneutic relationships is one of *realism* i.e., selecting the most obvious and simple interpretation coming from the machine data. The instrument provides a hidden phenomenological capacity: "I begin to accept, literally, the instrument mediated 'world' through what may be called here *instrumental realism*. The instrumentally constituted 'world' becomes the 'real' world. Not only do I forget the mundane world, but it begins to be downgraded" (ibid: 46).

3.3 The Diffusion of the Mind: A Body in the Environment

[C]ognitive science needs to put cognition back in the brain, the brain back in the body, and the body back in the world. (Wheeler 2007: 11)

As we have seen in Chapter 2, various threads in recent Continental philosophy have begun to see the human as a variable that is constructed through various technological and social practices. Feminist theory has explored these issues in length where theorists like Judith Butler claim that "[t]here is no gender identity behind the expressions of gender; ... identity is performatively constituted by the very ‘expressions’ that are said to be its results” (Butler 1990: 25). Similarly, Haraway (1991) proposes the cyborg as a solution to the problems of separations between humans and animals or humans and machines. The cyborg is a new entity that moves us beyond the constraints of gender, class and politics. In other fields, such as biology and anthropology authors have started to point out how organisms extend into their environment, such as the earthworm that uses the soil to keep water balance in its organism. The animate and the inanimate are seen as one system. The boundaries of the inside and the outside become fuzzy prompting Dawkins to talk about “extended phenotypes” (Dawkins 1982). It is outside the scope of this work to explore those interesting issues, but they derive from the same source of seeing the human as constituted by social and technological practices. As Simondon, Latour, and Mackenzie show, this constitution of the human is a highly complex process and the more technology becomes elevated in technicity, the more extensive and interleaved networks lie behind it, which makes its nature harder to grasp.

Another strand of philosophy, more inspired by biology, phenomenology and cognitive science, has begun to see the human body as a flexible construction that can be transfigured through different practices and prosthetics and one that is essentially dependent on its environment. Cognitive scientists have started using phenomenological approach in their experiments on embodiment (Gallagher & Sørensen 2006), where they expand the borders of phenomenology to include experimental (scientific) settings. This view of the mind-as-body-extended-in-the-environment can now be found in various cognitive science theories (Varela et al. 1991; Clark 1997; Hurley 1998; Haugeland 1998), in robotics (Brooks 1991; Arkin 1998), in linguistics (Regier 1996; Lakoff & Johnson 1999) and cognitive anthropology (Hutchins 1995; Suchman 1987). In this section we will look at two different but related views of human cognition: the enactive view and the theory of the extended mind, both emphasising the importance of the environment in the cognition of human agents, but from different premises.

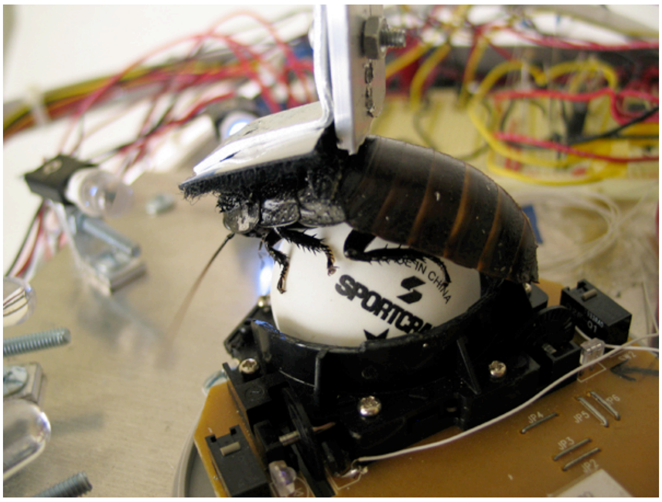

Figure 3:2. An orgcyb? This robot by artist Garnet Hertz is a humorous solution to the problem of spatial mapping of a robot. The cockroach on the ping-pong ball controls the robot through sensors sensing walls, resulting in lights shining into the cockroach’s eyes which in turn tries to avoid the walls.

3.3.1 Enactivism : Perception as Action in the Environment

3.3.1.1 Introduction

In their book *The Embodied Mind,* Varela et al. call for a pragmatic approach to the analysis of experience that could complement current trends in cognitive science. Their work can be put in context by pointing at how recent cultural (Jameson), philosophical (Nietzsche, Foucault, Derrida), psychological (Lacan, Turkle) and cognitive theories (Noe, Dennett) provide a picture of the self that is fundamentally fragmented, divided and non-unified. Varela et al. are adamant to point out that this delineation of the self recently emerging in the West has a much older philosophical foundation in Buddhist thought, particularly in the Mahayana tradition. Varela et al. derive this understanding of the self from various philosophical directions, but what colours their views is Varela's former work with Maturana on the theory of autopoiesis:

> An autopoietic machine is a machine organized (defined as a unity) as a network of processes of production (transformation and destruction) of components which: (i) through their interactions and transformations continuously regenerate and realize the network of processes (relations) that produced them; and (ii) constitute it (the machine) as a concrete unity in space in which they (the components) exist by specifying the topological domain of its realization as such a network. (Maturana & Varela 1980: 78)

Autopoiesis stresses the embeddedness of the cognitive agent as part of its environment; i.e., how the agent continually takes appropriate actions to maintain its unity through the processes of self-definition, self-maintainance, and (re)production. Living things are dynamic systems that are distinct, self-regulating, self-producing and autonomous.[46] However, although autopoietic theory implicitly depends on the idea of embodiment in its emphasis on biological processes, it does not deal with it explicitly as a main dynamic force in the cognitive development of the agent. Varela et al. are here developing their concept of "enaction," which stresses the importance of the environment as constitutive in the agent's cognition. The emphasis in Varela's later work shifts to embodiment; on how the world of lived experience is constituted through the coupling of the conscious biological organism and the external environment. Perception is a central topic here, as it is through our perceptual devices that we engage with the world. This is a view that is distinctly different from the cognitivist view of subjects as primarily symbol processing information that is passively given by the neutral senses. Enactivism shows, in a phenomenological manner, how we construct *our* (not "the") environment through active engagement with it.

---

[46] In the autopoietic theory, as interpreted here, a guitarist becomes a guitarist *when* he plays the guitar, i.e., the guitarist is defined as an autopoietic system that extends into the environment (the instrument and its sounds) and back. The vibration from the strings enter the fingers of the guitarist, from the guitar into the room and the room reflects back, conditioning the playing. A guitarist without a guitar is precisely that: a guitarist-without-a-guitar, thus not the autopoietic entity that we defined as a *guitarist.* Autopoietic ensembles establish, maintain, and destroy themselves constantly.

> We propose the term *enactive* to emphasize the growing conviction that cognition is not the representation of a pregiven world by a pregiven mind but is rather the enactment of a world and a mind on the basis of a history of the variety of actions that a being in the world performs. (Varela et al. 1991: 9)

The idea of the mind as a mirror of nature is rejected. We are physical beings that engage with the world through embodied action. Cognition depends upon the experience we have as bodies with various sensorimotor capacities who are themselves necessarily embedded in a biological, psychological and cultural context. The enactive approach to human cognition has had strong influences on various fields, such as HCI, ergonomics, education, psychology, etc. due to the interdependent relationship they grant the body and the environment; with emphasised focus on the role of perception in our engagement with the world. Perception is not a passive state but an active involvement with the world: "[it] is not something that happens to us, or in us. It is something we do" (Noe 2005). Noe here reiterates Merleau-Ponty's equation of perception and performance:

> The analysis of motor habit as an extension of existence leads ... to an analysis of perceptual habit as the coming into possession of a world. Conversely, every perceptual habit is still a motor habit and here equally the process of grasping a meaning is performed by the body. (Merleau-Ponty 1962: 153)

Perception and action are seen here as two sides of the same coin. "By using the term *action* we mean to emphasize once again that sensory and motor processes, perception and action, are fundamentally inseparable in lived cognition. Indeed, the two are not merely contingently linked in individuals; they have also evolved together" (Varela et al. 1991: 173). Knowledge is thus not knowledge *of* something but knowing *how* to do something. There is no object of knowledge. To know is to operate successfully in the enactive context in which the agent finds itself. In Maturana and Varela's own words "All doing is knowing, and all knowing is doing" (1987: 26) and "Knowing is effective action, that is, operating effectively in the domain of existence of living beings" (ibid: 29). Cognitive structures emerge from recurrent sensory-motor patterns, and these structures are non-representational, although they can be later translated into symbolic systems such as language. Recent development in neurophysiology (Metzinger & Gallese 2003) support these theories, prompting influential researchers in musicology and psychology to talk about cerebral versus corporeal intentionalities (Leman 2008: 84).

A common misunderstanding of enactivism is that it rejects entirely symbolic computation in the mind. This is not its point; it is rather that symbolic computation always *originates* in the subsymbolic domain of action-perception, as proved possible with non-symbolic artificial neural networks. But how would subsymbolic emergence and symbolic computation relate in natural system? Varela et al. provide the answer:

> these two views should be seen as complementary bottom-up and top-down approaches or that they could be pragmatically joined in some mixed mode or simply used at different levels or stages. A typical example of this move would be to describe early vision in connectionist terms, up to, say, the primary visual cortex. Then, at the level of the inferotemporal cortex, the description would be based on symbolic programs…. In our view, the most interesting relation between subsymbolic emergence and symbolic computation is one of *inclusion*, in which we see symbols as a higher-level description of properties that are ultimately embedded in an underlying distributed system…. In the inclusive view, the need for a symbolic level is acknowledged, but the possibility is left open that this level is only approximate. In other words, symbols are not taken at face value; they are seen as approximate macrolevel descriptions of operations whose governing principles reside at a subsymbolic level. (Varela et al. 1991: 100-102)

For an understanding of musicians' relationship with their acoustic or digital instruments, this perspective is important. We learn our instrument, habituate ourselves into their world, and through constant practice we become experts.[47] The instrument affords some perceptual/performative modalities that the musician incorporates. It is through an active involvement (or perceptually guided actions) with the instrument that the musician begins to understand its expressive scope. Such understanding can later become theoretical, abstract, and musicological. However, these are normally post-hoc symbolic abstractions of an embodied and tacit understanding, as we will see later in this chapter.

3.3.1.2 Enactivism in Music Technology

The issues of structural coupling between agent and environment (or musician and instrument), embodiment, haptics, perceptually guided action, etc., have all been dealt with in recent work in musical HCI, particularly in the field of NIME (New Interfaces for Musical Expression). Furthermore, a considerable amount of research has been done on the enactive role in musical perception (Pelinski 2005). However, only a nominal portion of this work actively takes an enactive approach when analysing the human-technology coupling. Some have flirted with enactivism (O'Modhrain & Essl 2005), but it is perhaps the PhD thesis of Newton Armstrong (2006) that has dealt with this topic in the depth that it deserves.

Armstrong uses enactivism as a theoretical foundation in his design of digital musical instruments. He introduces a theory based on Varela's work and recent approaches in HCI, which serves as a theoretical foundation for his technological invention; Mr. Touchy, a hardware interface for musical expression. He outlines the problems of what he calls "computer-as-it-comes," i.e., the computer with the traditional interfaces of screen, keyboard and a mouse/trackpad. A premise of his thesis is a discontent with the computer as it comes, and

---

[47] In the last section of this chapter we will explore Hubert Dreyfus' account of how such non-representational learning takes place.

his solution is to create interfaces with more tactile surfaces.[48] Armstrong explains his unease with traditional computer music software:

> the WIMP [window, icon, menu, pointing device] model of interaction leads to a mode of activity that is characterized by a sequential chain of discrete user gestures; the flow of time is effectively segmented into discrete chunks, where any action can be taken only after the prior action has been completed. There is no concurrency of actions, no possibility of operating at two or more interactional nodes simultaneously, and no potential for the cross-coupling of distinct input channels. (Armstrong 2005: 35)

He then lists five criteria for embodied activity: it has to be *situated,* as it happens in the context in which the agent interacts with the environment. Embodied activity is *timely*. The flow of activity is important and the agent will have to be able to act within realtime constraints. It is *multimodal*, where a large portion of the agent's sensorimotor capabilities are galvanised in performance. Embodied activity is *engaging*. The sense of embodiment arises when the agent is required by the task domain. The task domain presents challenges for the agent. Finally, it is an *emergent phenomenon* in the sense that the experience of embodiment happens in the process of working in a sensorimotor performance and incrementally increases over time (ibid: 9).

Contrary to Armstrong's claims, the fact is that all of these criteria can be met in a screen-based software instrument. The user can have various instruments (or input channels) open at the same time, thus a sufficient condition of an emergent performance, it can be engaging, definitively timely and of course situated. The last criteria, that of multimodality is the problem of the NIME research field, where various interfaces have been proposed and developed to the extent that they have become sophisticated musical instruments.[49]

Armstrong calls for coupled instruments where there is an explicit correlation between motor input and sonic output, and where there is a degree of resistance; a challenge built into the instrument that creates a "push and pull" tension between instrument and performer. Furthermore, Armstrong advocates the use of instrumental breakdowns, which is a strategy to be taken with a caution, as it would potentially result in a break of the performer's natural flow. The idea of breakdowns originates in Heidegger's breakdown of an equipment that switches the phenomenological mode of the tool from ready-to-hand to present-at-hand. Hamman (1999) originally picked up this idea and elaborates on the idea of "engineering breakdowns." However, Hamman is here mainly focussing on the compositional rather than the performative

---

[48] The computer-as-it-comes, defined this way, is of course a temporary condition of our technology as it was in 2005 at the time of Armstrong's writing. Already we have all kinds of alternative controllers, such as the popular Wiimote, and hardware is becoming ever more sophisticated as a sensory device. A recent paper by Fiebrink, Wang and Cook explores the laptop's various inbuilt musical interfaces (Fiebrink et al. 2007).

[49] With regards to the argument of the "computer-as-it-comes," the actual situation of users at home shows that interfaces such as the Wiimote are now becoming so popular that multimodal input devices are not an issue either.

context. Such oscillation can obviously be inspiring in the compositional situation, as if the compositional tool had some physical resistance. It could potentially be useful in the performative context as a designed resistance, but the idea of "breakdown" as a descriptive term here is far too strong. A breakdown would be the last thing the performer would want. Nevertheless, an instrument that challenges, inspires, and encompasses unknown potentiality is obviously desirable.[50]

The computer clearly allows for an embodied interaction just like the acoustic instrument. However, I am sympathetic to the general observations Armstrong makes as there are strong differences between acoustic and digital instruments. These issues will be dealt with in the second part of this thesis. The strong points of Armstrong's critique concern the typically sequential nature of operating digital instruments, the analysis of the *instrumental resistance* that we can and should build into the instrument, and the rather more complex issue of audience participation in the musical *performance*. Recent work in musical perception has demonstrated the role of mirror neurons when listening to music (Molnar-Szakacs & Overy 2006) and the obvious difference of seeing a guitarist play versus the computer musician working behind the screen becomes a proof in point here (Godøy 2003; Godøy et al. 2006). If the musician is merely sitting with a mouse, staring at the screen, the audience will have a distinctively disembodied experience of the musical performance. At the current time, as this research is still in its infancy, it is too early to draw concrete conclusions.

3.3.1.3 Enactivism Criticised from a Musical Enactive Stance

As mentioned above, the enactive view of human cognition has become influential in music and music technology design. Varela et al. mention that we often experience situations where our mind and body dissociate; where the body is performing one task but the mind is wandering somewhere else. Many musicians experience this when practicing scales or playing a piece that is not engaging. Musical performance should ideally engage and challenge the player. Now, through their Buddhist emphasis, Varela et al. also talk about mindfulness where the body and mind become one:

> We can *develop habits* in which body and mind are fully coordinated. The result is a mastery that is not only known to the individual meditator himself but that is visible to others - we easily recognize by its *precision and grace a gesture that is animated by full awareness*. We typically associate such mindfulness with the actions of an expert such as an athlete or musician. (Varela et al. 1991: 28, my italics)

[50] It has to be noted here that the *performative* nature of music limits the implementation of instrumental breakdowns. In other (and non-performative) fields, such as drawing, artists often use various tricks to create a novel interaction or functionality by misemploying the environment for the sake of creative outcomes. An example would be a draughtsman who flips the coordinates of the pen tablet in order to create the feeling of alienation in a tool whose expressive affordances she has thoroughly incorporated.

The problem here is that "developing a habit" means that we are incorporating a rote action. A habit can be as unfocused and unengaged as it can be "animated by full awareness." It seems that Varela et al. are not musicians themselves. On the next page they state that:

> As one practices, the *connection between intention and act becomes closer*, until eventually the feeling of difference between them is almost gone. One achieves a certain condition that phenomenologically feels neither purely mental nor purely physical; it is, rather, a specific kind of mind-body unity. (ibid: 29, my italics)

This is a surprising statement from the authors of the enactive view. We are here faced with a subset of the dualistic shift between mind and body; that of intention and act. Arguably, this is not how musicians learn to play scales. They do not have an abstract "intention"; a pure representation of what to do (*how* to play the scale or *how* it sounds); indeed, the scale is rather *discovered* (by incorporating it) through playing the instrument. When a guitarist has learned to play the major scale on the guitar (learning that happens by *doing*), and decides to learn/play the minor scale, she might hear the difference, but it is only by *trying* to play it that the scale-playing is learned. The problem is with the word "intention" above. Rather than defining it as a top-down process with the (imperfect) body trying to perform the (perfect) mental intention, as Varela et al. suggest, it should be understood as bottom-up process, that is heard and learned through a genuine action-perception coupling.

3.3.1.4 The Neglect of Technology as a Structural Element in Enactivism

Autopoietic agents are structurally open (they can interact with the environment and change shape, matter and location), but organisationally closed (the agents preserve themselves and continually reinstatiate and reiterate the distinction between the self and the environment). The agent extends itself through incorporating technology into the unity of the body or extends the body into the wider environment. In autopoiesis the agent can extend its systemic boundaries through incorporating elements from the environment (such as technological prosthetic devices) into its own structure. Varela et al. recognise the role modern technology plays in our thinking, in both daily life and scientific research:

> [k]nowledge has become tangibly and inextricably linked to a technology that transforms the social practices which make that very knowledge possible – artificial intelligence being the most visible example. Technology, among other things, acts as an amplifier. (Varela et al. 1991: 5)

What is lacking here is an acknowledgement of the role that computer technology has performed in generating the specific *research questions* of artificial intelligence. As Latour (1987) points out, technology plays a decisive role in how scientific knowledge is constructed. For Stiegler, this role technology plays as an external prosthesis of the human is of a vital importance, prompting him to claim that "the being of humankind is to be outside itself" (Stiegler 1998: 193). In fact, Varela et al. put forth a highly social constructivist view of technology where it is seen as a result of the rational thinking of scientists, as the *result* of thinking, thus in many ways sustaining the Platonic dualism of the interior and the exterior, of ideas versus matter, or, in logocentric terms: speech versus writing.

> science – again unlike other human practices and institutions – *incarnates its understanding* in technological artefacts. In case of cognitive science, these artefacts are ever more sophisticated thinking/acting machines, which have the potential to transform everyday life perhaps even more than the books of the philosopher, the reflections of the social theorist, or the therapeutic analyses of the psychiatrist. (Varela et al. 1991: xvii, my italics)

Even though autopoietic theory explains the "structural coupling" between the agent and the environment and how the organism "brings forth" its environment, Varela et al. seem to have a rather naïve understanding of the technological human (or the post-human), and of the ways in which our culture is technologically constituted through the transductive dialectic. They claim that technology is simply something that we make to "incarnate" our ideas but fail to engage properly with the role technology poses in our being-in-the-world and thinking through externalising functions, replacing body parts and mapping the interior. This neglect of technology as *affecting* environmental factors, as objects with an agenda, is related to an issue that has been pointed out by Johnston, namely that the lack of thorough analysis of the constitutive role of technics in both cognitive science and AI, accounts for the blurred boundary between the two fields. "In the hiatus between wanting to *produce* artificial intelligence and wanting to *understand* natural intelligence, the constitutive role of technics unaccountably drops out of sight, even though both oppositions (produce/understand, artificial/natural) presuppose it" (Johnston 2002: 477). What is needed is a proper account of technics as the hidden framework of modern cognitive science and AI.[51]

3.3.2 The Extended Mind in the Environment

3.3.2.1 Introduction

As demonstrated in the work of Varela et al., the idea of the human subject as independent of the world does not make sense. We are beings-in-the-world that become what we are through engaging with its material properties and artefacts. Our relationship with the external world forms our cultural and biological evolution. Technology (as an independent evolutionary phenomenon) is one such external factor that defines our existence as humans. This can seem

[51] Johnston himself does not manage to provide a sufficient analysis in his paper. Instead, it becomes a rather conventional, but good, account of AI and new-AI techniques. This is understandable: critical theory/philosophy and computer science are two very different discourses that do not easily conflate.

like 'the chicken and the egg' problem: does technology determine us or do we shape technology? What comes first? For Stiegler there is no question:

> Technics, which appears to be a power in the service of humanity, becomes autonomous from the instance it empowers – technics ought to be an act on the part of humanity – as a result of which it does a disservice to active humanity, that is insofar as humanity communicates, makes decisions, and assumes individual form. (Stiegler 1998: 13)

> Technology evolves *more quickly* than culture. More accurately, the temporal relation between the two is a tension in which there is both advance and delay, a tension characteristic of the drawing-out which makes up any process of temporalization. (ibid: 43)

Stiegler echoes Heidegger's claim that we are born into the world of equipment, it is "always already" there (to use a post-moderinst lingo). If technics is the constitutive element of human life, how does that work? Was the pen (technology) invented before writing (culture)? From the transductive perspective, this is a meaningless question. There is no need for the pen if there is no writing. There is no writing without writing tools. An evolutionary history of writing would presumably show us how the prehistoric human would discover some tool to carve marks onto rocks or sand, thus externalising herself onto some material surface. This "writing" is only possible through an *embodied* exploration of the tool used; it would not happen while sitting on a rock and suddenly become illuminated by an idea. The technology (the pen) here precedes the invention of writing.[52] As we have seen, this dynamic of the supplement or the prosthesis *as a problem of temporality* has been worked on in the late 20th century Continental philosophy, in particular by Derrida and Stiegler.

The focus of this section is on how the problems of externalisation or prosthesis are dealt with in another discourse, namely cognitive science. It is appropriate to look at Andy Clark's theory of the "extended mind," and how it relates to the topic of epistemic tools; of technology as an extension of the human cognitive system.

3.3.2.2 The Extended Mind

We use intelligence to structure our environment so that we can succeed with less intelligence. Our brains make the world smart so we can be dumb in peace! (Clark 1997: 180)

In the 1990s Andy Clark worked on a theory of active externalism, often called "theory of the extended mind." His main thesis is that we use our environment and other tools, such as language, to externalise our cognition in order to ease the cognitive load on the brain. Clark

---

52 The argument here being, of course, that the pen is merely a refinement of the initial writing technology.

defines this as an example of *cognitive technology* (Clark 2001). We think *in* the world: it is the *loci* of our cognition. We offload our cognition onto physical or linguistic "scaffoldings" that support our cognition and become integral parts of it. Post-it notes, notebooks, scribbling words or making diagrams, abacus or pebbles for calculating, a tied knot around the finger in order to remember something are all mnemotechnologies that we use in our mental offloading. This might seem strange at first, some form of mystical panpsychism, but Clark is clear on the fact that he is not talking about extended consciousness. "Not every cognitive process… is a conscious process" (Clark 1998: unpaginated). There is no point in identifying consciousness with cognitive processes. In any event, cognition does not take place exclusively within the skull:

> [In certain conditions, t]he human organism is linked with an external entity in a two-way interaction, creating a *coupled system* that can be seen as a cognitive system in its own right. All the components in the system play an active causal role, and they jointly govern behavior in the same sort of way that cognition usually does. If we remove the external component the system's behavioral competence will drop, just as it would if we removed part of its brain. Our thesis is that this sort of coupled process counts equally well as a cognitive process, whether or not it is wholly in the head. (Clark & Chalmers 1998: 7)

And elsewhere Clark states:

> It just *doesn't matter* whether the data are stored somewhere inside the biological organism or stored in the external world. What matters is how information is poised for retrieval and for immediate use as and when required. (Clark 2003: 69)

Clark and Chalmers introduce the *pairity principle*, namely that if a function that might otherwise be performed in the mind is now performed outside the head, in the environment, it can be defined as belonging to the cognitive process. In short, we are constantly "supersizing the mind." (Clark 2008).

The theories of external cognition are philosophical engagements with ideas that have been well explored in the last two decades of the 20th century. Nardi and Zarmer (1990) conducted research on how people frame models in terms of the tools they use and took the example of the spreadsheet. Nardi and Zarmer show how people use the tool to frame the problem; the problem is not defined and already existent in the mind of the user. Similarly, in their 1998 paper, Clark and Chalmers take the example of the Tetris computer game, where bricks fall down and the player is supposed to fit them into slots by rotating them appropriately. They maintain that it is simply faster and more effective to use our body in calculations such as these and rotate the bricks in order to *see* whether it fits, rather than thinking it through.[53] They refer to a paper by Kirsh and Maglio (1994) where the term *epistemic actions* is used for such physical gestures that effectively serve as thoughts. In their paper, Kirsh and Maglio state "Epistemic actions – physical actions that make mental computations easier, faster or more reliable – are *external* actions that an agent performs to change his or her own computational state" (ibid: 513). Using empirical experiments, they demonstrate how cognitive performance deteriorates if the agent relies solely on internal memory, as opposed to external memory support. They refer to Hitch (1978) on cognitive load, Lerdal and Jackendoff (1983) on musical composition, and Hutchins (1990) on marine navigation as a means of verification.

Kirsh and Maglio demonstrate how epistemic actions can improve cognition on three different modes of complexities: a) *space complexity*: reducing the memory involved in computation, b) *time complexity*: reducing the number of steps involved in the computation, and c) *unreliability*: reducing the probability of error in computation. We will see in the next chapter how the idea of epistemic actions translates into the focus of epistemic tools, i.e., tools that support specific ways of thinking. This is particularly relevant to digital tools as they are, on the one hand, designed from the perspective of being cognitive tools, and on the other hand, their design is of a conceptual nature; the material of digital tools is code (i.e., language) and as such they are tools for thinking created by (symbolic) thinking.

In his paper "Magic Words: How Language Augments Human Computation," Clark maintains that natural language is effectively the same type of scaffolding as the external physical representations of the "Extended Mind" paper. Language is a technology for thinking. He takes examples of how academic texts are written with external support, using notes, illustrations, graphs, quotes, etc. And further, how poetic writing is often the act of exploring words, of the interpretative "affordances" of the words as they appear on the paper in the process of writing. Speech is thus an external support for our thoughts, and of an interactive nature.[54] Clark refers to speech as "public thinking" and in a Wittgensteinian manner (Wittgenstein 1969: 6, 15-16, see quote in section 6.3.1) defines thinking *as* speaking or *as* writing respectively, when we perform those activities. Inner language (or silent speech) is

---

[53] In *The Phenomenology of Writing by Hand* Daniel Chandler (1992) deals with the topic of externalisation of thought through tool use, here specifically with the pen. He gives examples of how many writers cannot think without the pen. Furthermore, Cynthia Selfe has studied what happens when the computer becomes the chosen tool for writing and how the personal connection with the pen is lost: "Maybe I'm too far away with the computer. I mean the screen is *there*, and I am *here.* With a pencil and paper I'm touching the words. Also they look like *you* wrote them, not like the *machine* wrote them" (Selfe 1985).

[54] Things are never that simple. Speech as cognitive extension is inevitably culturally contingent, as Donal Carbaugh's (2005) *Cultures in Conversation* book shows. Carbaugh describes his cultural shock when visiting a Finnish University as a Fulbright scholar. The tempi of American and Finnish conversations are so different that it created an uncomfortable cultural gap. It shows how some cultures are more prone to think "outside of the head" as opposed to internally. This case is an example of different metacommunicative rules that define the communicative action.

similarly a scaffolding of thought. However, it requires more cognitive effort as we do not have the *external* scaffolding of language being outside-oneself. Linguistic formulations thus help us to focus, monitor and control behaviour. This second order dynamics further allows us to think about our thoughts, through the act of thinking about internal language. But as examples indicate, people prefer to externalise their thoughts by writing them down or drawing schemas in order to minimalise the cognitive load. This relates to recent studies in education, indicating that learning through conversation is a more effective way of digesting and understanding the material at hand, effectively learning through *doing*. (Kolb 1983; Baker et al. 2002)

3.3.2.3 The Discovering of Tools for Thought

In 1999 Andy Clark said he "had a ball." He read two texts by Daniel Dennett, "Things about Things" (1999b) and "Making Tools for Thinking" (1999a) that both reveal a support for the theory of the extended mind, in particular the ideas of language and tools. They do in fact show that there are "mind-tools *all the way down*" (Clark 1999: 2). In the same way that Derrida and Stiegler talk about the original technicity or the original supplement, Clark draws the conclusion that we are "natural-born cyborgs." It is only the level of technicity of our tools that have become more sophisticated and complex – more "social" as Latour would put it, as there are more networks that define the technological collective. The point is that the human has always been a tool-using animal that externalises its cognition out in the world.

In "Making Tools for Thinking," Dennett writes "*minds are composed of tools for thinking* that we not only obtain from the wider (social) world, but largely leave in the world, instead of cluttering up our brains with them" (1999a: unpaginated). In an analogous argument to Stiegler who problematises the idea of a "double origin" of the homo sapiens, i.e., one of its origin and the other of its use of language and tools,[55] Dennett simply claims that originally our tools are machines made of found objects. We are transformers; virtual machines that make more virtual machines out of our interactions with the environment. Clark interprets Dennett as proposing three questions with answers:

> Q/ What are minds made of?
> A/ Tools for thinking.
>
> Q/ Who or what uses the tools to do the thinking?
> A/ No-one, nothing. The tools-are-use.
>
> Q/ Intentionality, aboutness, content and consciousness: can all these really be

[55] Stiegler thus submits that the "cortex and equipment are differentiated *together, in one and the same movement.* The issue is that of a singular process of structural coupling in *exteriorization* that we are calling an *instrumental maieutics,* a "mirror proto-stage" in the course of which the differentiation of the cortex is determined by the tool just as much as that of the tool by the cortex: a mirror effect whereby one, looking at itself in the other, is both deformed and formed in the process" (Stiegler 1994: 158).

> brought into being by grab-bags of userless tools?
> A/ Yes. (1999a: unpaginated)

Dennett denies the traditional cognitive science of internal representations or the view of the mind as a symbol-crunching machine. Dennett's main point is the emphasis of "the skills bequeathed by the tools that build know-how tacitly into the system" (Clark 1999: 3). There are two important issues here: a) that tools transfer skills; they afford and inherit us with those skills through application, and b) that this happens *tacitly*. Tool-use does not require explicit syntactically distinct symbols, but rather some embodied *habituation* (this will be explored in the next section) of the environment. Just as Varela et al., Dennett and Clark come to the conclusion that tacit knowledge is more fundamental than internal representations:

> The trick here… is to recognize the tools as themselves a class of replicating entities whose "success" (widespread replication) or "failure" (extinction) depends on the extent to which we adopt them. Tools in general (and language in particular…) may thus be seen… as rather like viruses – incapable of reproducing on their own, dependent on a host's metabolic and reproductive systems, yet susceptible to processes of variation and differential reproduction sculpted by their success of failure at invading host organisms. (ibid: 6)

Independently of French post-structuralist thought, Dennett and Clark discover the function of technology as *epiphylogenesis*; the origin of human thinking and preservation of human memory through technical objects [see section 2.5]. Furthermore, independently of Anglo-French actor-network theory, they came to treat tools as nonhuman actors; as viruses that replicate, but whose success or failure depends upon the social subscriptions of the tools [see section 2.4]. That said, neither of them manage, or take time, to elaborate as profoundly, or in-depth, the theoretical nuances of Stiegler or Latour. Clark illustrates the way we use our environment in our thinking, how cognitive processes are realised in external objects, though he does not close the loop. It is through the analysis of how technology conditions human thought that we are able to close that feedback loop. Indeed, exploring this feedback loop between technology and thinking is one of the principal investigations of the current thesis; applied in the domain of music technology.

3.3.3 Conclusion

A forthcoming issue of the philosophical journal *Topoi* deals with the problems pointed at by Wheeler (2008); that there are certain incompatibilities between the extended mind thesis and the theory of enactivism. In the publication, Thompson and Stapleton (forthcoming) stress that the questions of the inside or outside are not the interesting or important questions to explore, but rather how the body and the nervous system is compositionally plastic, being able to extend themselves through tools and processes that go beyond the metabolically generated biological body. In another paper in this issue of Topoi, Di Paolo (forthcoming) attempts to clarify the enactive program, claiming that it should not be interpreted as the internalist account of cognition that makes Wheeler claim that the two theories are incompatible. Di Paolo attests that they have much in common, and calls for a clearer definitions of cognition, autonomy, agency and individuation from the extended mind camp. Di Paolo sees the extended mind thesis as a powerful catalyst that generates fruitful responses by the proponents of the enactive view on important topics that have traditionally been neglected within enactive theory:

> The more interesting and forward-looking themes introduced by the EM [Extended Mind] approach, and towards which enactivism must still develop, include the problems of space of technical individuation and technological networks that bootstrap the generation of cognitive identities…. Such "living" media show aspects of operational closure at collective levels without necessarily showing individuality…. Such cases put in evidence that the richness of environments as active media has so far been underplayed in the current enactive story. EM has sought to thematize this richness and the enactivist should listen. (Di Paolo, forthcoming: unpaginated)

As stated above, this thesis aims to explore this question of "technical individuation," "technological networks", and the environment as "active media" from a view of embodiment derived from enactivism and extended mind, but of technology as "inscribed" environmental artefacts, derived from epiphylogenesis and actor-network theory. This will be achieved through partly phenomenological and epistemological explorations of human tool-use and human-machine relationships. Music technology presents itself as a research field *par excellence*, as it requires human dexterity, sophisticated ergonomics, and realtime considerations in terms of time and space.

3.4 Habituation: An Embodied Relationship to Technological Artefacts

The grammar of the word "knows" is evidently closely related to that of "can", "is able to". But also closely related to that of "understands". ('Mastery' of a technique) (Wittgenstein 1968: §150)

This chapter has focused on various topics that relate to how a human being learns to operate a complex technical tool. One approach taken in HCI is to use mental models as explanatory devices. However, so much has been written about mental models in the 20^{th} century that the concept has become watered down and less useful for the current context. Typically, theories of mental models are presented from a classical cognitive science stance, where they are often seen principally as deductive reasoning tools. They are conceived of as systems of internal representations, following a symbolic view of the subject's cognitive structure (Johnson-Laird 1983). Although the origin of mental models is to be found in an early work by Kenneth Craik

(1943), it was not until in the 1980s, following the success of cognitivism and optimistic beliefs in symbolic AI, that mental models were picked up again in cognitive science and succinctly in HCI. Craik describes them here:

> If the organism carries a "small-scale model" of extended reality of its possible actions within its head, it is able to try out various alternatives, conclude which is the best of them, react to future situations before they arise, utilize the knowledge of past evens in dealing with the present and the future, and in every way react in a much fuller, safer, and more competent manner to the emergencies which face it. (Craik quoted in Johnson-Laird 1980: 73)

At first glance there seems nothing wrong with this portrayal of human agency. However, the problem we have with this account is its predilection for the top-down and symbolic perspective. Many of the works in cognitive science in the 1980s and 1990s point out the problems with this view by bringing the body and the environment into the cognitive equation. For example, Nardi and Zarmer (1990) analyse how cognitive representations of a problem arrive through an engagement with the task, and in particular affected by *which tools* are used. As we saw in section 3.3.2 human cognisers do never possess a fully constructed model of the problem; it is typically discovered through an external representation of its parameters. Nardi and Zarmer give an account of the use of the spreadsheet as an example:

> The spreadsheet thus offers good access to techniques people can use to *create* an explicit model of a particular application - but it does not in any sense match a "mental model" of an application; such models… do not exist. As the user enters data and formulas into the spreadsheet, the representation of the problem emerges *through the medium of the spreadsheet.* The representation is not "in the user's head," nor was it ever in the head, but becomes an artefact created by the user's interactions with the program. (ibid: 7)

Nardi and Zarmer claim that it is counterproductive to use mental models in the design of technological artefacts (ibid: 4). Winograd and Flores (1986) agree and maintain that the "rationalistic" tradition (or cognitivism) has been based on a naïve correspondence theory of language (ibid: 19) which materialises in how it portrays problem solving. This is the model of cognition they reject: First you have the *Task environment*. Here the problem is contextualised in terms of "states" of affairs, "actions" that are available to the problem solver to change the state and "goals," from which rational actions can be derived. The subject has an *Internal representation* of the task environment. This is a collection of "symbolic structures" where properties of the objects in the environment and their relations are represented. Then the subject has to perform a *Search*. This search space is derived from the internal representation of a vast array of potential actions. The problem solver goes through the information process of finding efficient ways to the desired goal. Finally, the subject has to *Choose*. The problem solver selects a course of action from the array of possibilities (ibid: 23). Against such representational accounts of cognition, Winograd and Flores propose a more phenomenological and non-representational account, one strongly influenced by Heidegger:

> In driving a nail with a hammer (as opposed to thinking about a hammer), I need not make use of any explicit representation of the hammer. My ability to act comes from my familiarity with *hammering*, not my knowledge of *a hammer.* (ibid: 33)

Although mental models are still used in cognitive science and HCI today, they are primarily seen as models that are built on-the-fly from knowledge of previous experience, schema segments, perception and problem solving strategies. They are internal scale-model representations of some system or object in the external reality. The models are used when taking decisions in novel circumstances, and they are "runnable" in a way that the subject can mentally rehearse the model, as if it was a computer software (Markham 1999). As such, they can be useful for contextual mock-ups of user situations and learning processes, but there is still the fundamental problem of the emphasis on the subject's symbolic representation. Although one could argue that mental models might suit to model the learning process of computer systems as opposed to, say, musical instrument or tennis playing, due to the abstract, symbolic and mental nature of such human-computer interactions, I will refrain from such generalisations and rather subscribe to an enactive and contextual model of perceptio-action.

In the argumentation of this thesis, an account is needed demonstrating how people learn to use technical tools; how they gradually come to understand the tool to such a degree that they start to think in terms of its affordances and constraints. Bourdieu's (1990) concept of *habituation* is useful here. Compared to the theory of mental models, habituation is much more concerned with the way Varela et al. (1991) talk about an agent's structural coupling with the environment, and how learning is essentially the process of adapting one's body to affordant features in the environment through action. Bourdieu develops the concept of *habitus* in order to get around the problems of dualism in social theory; that of mind and body, agency and structure, or subjectivism and objectivism. Habitus is, as Bourdieu defines it:

> systems of durable, transposable dispositions, structured structures predisposed to function as structuring structures, that is, as principles which generate and organize practices and representations that can be objectively adapted to their outcomes without presupposing a conscious aiming at ends or an express mastery of the operations necessary in order to attain them. Objectively 'regulated' and "regular" without being in any way the product of obedience rules, they can be collectively orchestrated without being the product of the organizing action of a conductor. (Bourdieu 1990: 53)

Habitus is therefore a very different picture of human engagement in the world from the one of mental models. It has much more in common with the phenomenological views presented in chapter 2; of particular relevance is Merleau-Ponty's account of the organ player [quoted in section 3.2.2]. Bourdieu proposes habitus as an alternative to the popular structuralism of the 1960s where subjects incorporate social structures (become carriers of them), and to the philosophy of the 'subject', which sees the agent as a rational self-knowing and functional agent. Habitus provides an embodied view with capacity to include tacit knowledge, unconscious actions, and distributed actions over many agents.

Habituation is the acquisition of habitus. It is enactive action that constructs an understanding of the environment, or here, an understanding and skill of the musical instrument. Bourdieu often defines habitus as "regulated improvisations" (ibid: 57). It is not a habit – as that would imply a "mechanistic vision" – but rather an "active and creative relation to the world" (Bourdieu 1992: 122). Merleau-Ponty defines habit (although I prefer the term "habituation") as a "rearrangement and renewal of the corporeal schema" (Merleau-Ponty 2002: 164) and for him, "every habit is both motor and perceptual, because it lies … between explicit perception and actual movement" (ibid: 175). Following Heidegger and Merleau-Ponty, Bourdieu is able to define tacit knowledge as embodied skill, not easily articulated in propositional form, but nevertheless a practical ability that defines the agent's actions in the environment. It is the knowledge of *knowing how,* rather than *knowing that,* as Ryle (1949) put it; a distinction that can easily be transferred to the domains of musical instrumental skill and music theoretical knowledge. It should be noted that Bourdieu argues that by rationalising habitus – in an ethnographic study for example – one has transformed embodied contextual connections into abstract information, discrete entities, and sequential instructions (Bourdieu 1977). As such, we have robbed the embodied body of its knowledge and represented it in a manner that will always be a representation, a symbolic rather than actual understanding.

There is much to be said about Bourdieu's social theory and many aspects of it would be of interest here, such as the ways he sees technology as necessarily embodied in lived practice through habitus, and how musical styles could be seen as social practices of habitus. We could also engage in critical discussions of the lack of importance he places on technology as actors in society, or of the uncritical view he has of technology as merely passive elements of social structures. However, these lines of investigations extend beyond the scope of this thesis. What this section has achieved is to show the utility of the word "habituation" to denote the *enactive* incorporation of musical skill into the human body; a word that connotes an essentially different understanding of learning and virtuosity to that of mental models.

3.5 The Body as a Space for Writing: On Socio-Cultural Inscriptions

Social and cultural process inscribe the body with meanings, and the body, which is always more than these meanings, projects its realities onto social spaces. The process is recursive: the body organizes the culture and society that inscribe this same body with meaning. (Frank 1996: 737)

3.5.1 Inscriptions and Incorporations: The Body and Embodiment

In 1999, N. Katharine Hayles published a book called *How We Became Posthuman: Virtual Bodies in Cybernetics, Literature and Informatics.* In this work, she demonstrates how the idea of the "liberal humanist self" was made extinct through various theoretical lineages of the 20th century, such as feminist theory, deconstruction, post-structurlism, post-colonialism, cybernetics and cognitive science. The amplification of computationalism endangered the liberal self through three interconnected steps: a) how "information lost its body," i.e., how classical cognitivism portrayed human cognition as disembodied, b) how the cyborg was created as a cultural artefact after the 2nd World War, and c) how the posthuman slowly began to replace the historically specific construction called the human. The posthuman appears when computation rather than humanism is taken as the ground of being; a view that allows the posthuman and intelligent machines to exist at the same ontological level. The term does not signify the "postbiological," like Hans Moravec and others are predicting, it merely signals the end of a specific conception of the human. And it is not necessarily a negative change, as in posthumanity:

> emergence replaces teleology; reflexive epistemology replaces objectivism; distributed cognition replaces autonomous will; embodiment replaces a body seen as a support system for the mind; and a dynamic partnership between humans and intelligent machines replaces the liberal humanist subject's manifest destiny to dominate and control nature. (Hayles 1999: 288)

Reverberating Clark, who defines humans as natural-born cyborgs, Hayles claims that we have always been posthuman. Her project, however, is to explain *how* and *why* information needs a body, that we are embodied and situated beings by necessity.

Hayles is interested in the body and embodiment. But what is the human body and to what extent can it be seen as constructed by ourselves? Leder (1990) draws upon the distinction between the words *Leib* and *Körper* in German that designate the lived body and inanimate or dead body, respectively. Phenomenological investigations are the domain of *Leib,* medical theories the domain of *Körper.* "These are not two different bodies. *Korper* [sic] is itself an aspect of *Leib*, one manner in which the lived body shows itself" (ibid: 6). Similarly, Foucault in his *Archaeology of Knowledge* has defined the human body as a construction of discourse systems. In yet another discourse, cybernetics defined information as independent of the body, thus effectively eliminating it. Arthur and Marilouise Kroker have written about how the body is disappearing in terms of ideology (an object of fashion), epistemology (broken Cartesian dualism), technology (prosthesis eliminate natural boundaries), and semiotically (by the use of tattoos, piercing and other floating signs) (Kroker & Kroker 1987: 20). True to form, Baudrillard simply judges the body as “superfluous” (1988). In this firestorm of body-executions, Hayles proposes a framework which does not deny the body, but rather claims that a new subjectivity has emerged, and one that allows us to think about embodiment in the age of virtuality:

> This framework comprises of two dynamically interacting polarities. The first polarity unfolds as an interplay between the body as a cultural construct and the experiences of embodiment that individual people within a culture feel and articulate. The second polarity can be understood as a dance between inscribing and incorporating practices. Since the body and embodiment, inscription and incorporation, are in constant interaction, the distinctions forming these polarities are heuristic rather than absolute. (ibid: 193)

Embodiment differs from the concept of body in that it is contextual; confined to the specifics of time, location and culture. The body is the universal, a normative construction, an idealised form that “gestures towards a Platonic reality” (ibid: 196); an entity that is constructed in the discourse of the sciences and philosophy. Embodiment and the body can coexist, but embodiment is the specific, the singular; a body with colour, gender, race, class and (cultural) environment. The second polarity – inscription and incorporation – is a more performative rather than descriptive definition, analogous to the first polarity, although the two bimodalities can be seen as “acting in complex syncopation with each other, like two sine waves moving at different frequencies with different periods of repetition” (ibid: 198). Incorporation practices are actions that are encoded into bodily memory through repetition until they become habitual. Learning to play a musical instrument is an incorporation practice. Inscription practices abstract these practices into signs, i.e., they normalise in order to create symbolic representations of bodily actions; they would be the musical theories or practices of musical performance. These practices exist on different levels, the level of bodily meaning (a tacit knowledge that is often hard to verbalise), and the level of symbolic discourse *about* actions. These practices coexist and neither of them is preferable to the other: “The body is enculturated through both kinds of practices” (ibid: 200). Both practices are cultural, contextual and form what Bourdieu called the habitus: “The body produces culture at the same time that culture produces the body” (ibid: 200). The main point here is that the body knows more than it can verbalise, a fact that has founded and supported Rodney Brooks’ design of embodied, non-representationally, distributive intelligent robots (Brooks 1991).

Unlike Bourdieu, Hayles is conscious of technology's ability to change individual patterns and social structures. Incorporated knowledge is improvisational, hard to abstract, confined to the physical circumstances, and deeply sedimented into the body. When technological changes take place they often affect how people use their bodies and their experience of space and time. Embodiment thus mediates between technology and discourse and serves as a boundary marker between the corresponding discursive systems. "In the feedback loop between technological innovations and discursive practices, incorporation is the crucial link" (Hayles 1999: 205).

In a Heideggerian manner, we might now ask: "but where have we strayed to?" Using this analysis, as we will see in the coming chapters, we are able to define musical software as bodies of inscriptions, i.e., the software is a technological entity, a digital body that has been inscribed with generalised music theory, in a way that the singular – the musician's body – will never incorporate. Musical software, as an epistemic tool, is a system of signs that operates independently of the musician's body, even if its control design seeks to imitate that of the acoustic instrument. The epistemic tool, as an abstraction of knowledge, will always tend to be generic, universal, non-contextual, normative and prescriptive in the actual circumstances of making music. Conversely, the embodied knowledge gained by incorporated practice of interacting with an acoustic instrument in the trained musical body will exhibit different being-in-the-world, different understanding of music, and different ways of improvising in real situations than possible with the epistemic tool. Of course, there are subtleties at play that are not given justice with the crude generalisation above; those will be dealt with in the next chapter.

3.5.2 The Cyborg: Technology as Literal Prosthesis in Praxis

Late twentieth-century machines have made thoroughly ambiguous the difference between natural and artificial, mind and body, self-developing and externally designed, and many other distinctions that used to apply to organisms and machines. Our machines are disturbingly lively, and we find ourselves frighteningly inert. (Haraway 1991: 152)

In the first work ever to tackle the philosophy of technology, *Grundleinen einer Philosophie der Technik* (1877), Ernst Kapp defines technics in various forms as *Organprojectionen*, as extension of the human organism: prosthesis.[56] We have already seen in the preceding chapter how technology has been defined as nonhuman agents that restructure ourselves, as the originary supplement, or as an extension of our memory. In this chapter we explored technology from a phenomenologal perspective, and surveyed Ihde's different relation modalities we have

[56] Interestingly, the Greek word *organon* refers to for both tools and organs. Musical tools are also *organon* and thus we get the scientific study called *organology* (the study of musical instruments) that we will introduce in chapter 5.

with the world. We also surveyed various theories inspired by biology, cognitive science, sociology and semiotics. All have in common a new understanding of the human as an essentially a prosthetic being, which uses technologies (such as language) to externalise their cognition. But we do not merely make use of cognitive extensions of our mind-body, we also use physical extensions, prosthetic technologies, to amplify or reduce the perceptive capacity of the body. This extended body, this biomachine has a name, the cyborg.

The cyborg is a neologism that, as most people know, is made out of two roots: cybernetics and organism, i.e., the cyborg is a cybernetic organism. The cyborg maintains an organisational equilibrium (or autopoiesis) as early cybernetic theory explains, but does so with technological extensions to the body and/or the mind. In contemporary usage, the cyber- prefix has began to stand for technology, everything digital, as in cyberspace,[57] cyberculture, or cybercafes. Etymologically "cyber" comes from the Greek *kubernetes,* a steersman/governor (whose root is the Latin *gubernare* from the Greek) of a ship, a word taken up by Norbert Weiner when he developed his system of self-governed systems. In the context of the cyborg as prosthesis, a certain passage in Aristotle's *Politics* is relevant:

> Tools may be animate as well as inanimate; for instance a ship's captain [*kubernetes*] uses a lifeless rudder, but a living man for watch; for a servant is, from the point of view of his craft [*techne*], categorized as one of its tools. (Aristotle *Politics* 1(4), 1253b in Ackrill 1987: 511)

This passage is interesting in many ways, not least for the fact that Aristotle is using the steersman [*kubernetes*] of a ship as an example (which is the root of the term "cybernetics"), but also because he is able to see animate living beings as tools.[58] The ship has animate tools, such as the servant, and inanimate ones, such as the rudder. The servant's craft is *techne* which in turn is one of the ship's tools. This argument is repeated in the *Capital* where Marx claims, with more seriousness that the "workman [is converted] into a living appendage of the machine" (Marx 1887 vol 1: section 9). It becomes clear how, throughout history, there have always been views where the human is defined an extension of the machine, a biological prosthesis of the mechanistic organism.

Donna Haraway is perhaps the best known theorist of the cyborg. She analyses the human-machine combo, the cyborg, as a politicised entity, with the aim of creating alternative views, languages and practices of technoscience, and hybrid subjects. The cyborg is not a science-fiction construction but a very real situation of the human and its relationship with

[57] In contemporary popular culture the term cyberspace was coined by William Gibson in his 1984 novel *Neuromancer*. "Cyberspace. A consensual hallucination experienced daily by billions of legitimate operators... A graphic representation of data abstracted from the banks of every computer in the human system. Unthinkable complexity. Lines of light ranged in the nonspace of the mind, clusters and constellations of data. Like city lights, receding" (Gibson 1984:51). Gibson was also instrumental in creating a genre of science fiction called cyberpunk.

[58] The fact that Aristotle is mainly talking about master-slave relationships in this chapter is ignored here.

technology. Hearing aids, glasses, artificial organs and limbs, cybergoggles or cybergloves are all famous representations of our cyborgian nature. Doctors perform operations on patients through telepresence, and people perform telesex over the net. A more simple example, the dentist's probe as analysed by Ihde, would suffice as well to illustrate this feature of our embodiment.

The cyborg is often talked about in a more metaphorical sense, perhaps akin to Andy Clark's idea of scaffolding, where we are all portrayed as cyborgs already through our use of mobile phones, search engines, and other external devices. Such human-machine relationship has, for the sake of differentiating from the more biological coupling, been termed "fyborg," or functional cyborg, in some literature (Chislenko 1995). That is arguably not a necessary distinction, as it really does not matter if the prosthetic technological artefact is inside (physically coupled) or outside the body, as Clark's extended mind thesis illustrates. We thus derive at a definition of the cyborg as both metaphorical (where technology is used as extended cognitive faculties) and bio-prosthetic (where technology actually becomes an internal functional part of the body).

In a manner similar to Hayles above, Haraway sees the body not as something natural and independent of technology but as necessarily integrated with it:

> I am defining corporealization as the interactions of humans and nonhumans in the distributed, heterogeneous workprocesses of technoscience. The nonhumans are both those made by humans, for example, machines and other tools, and those occurring independently of human manifacture. The work process result in specific material-semiotic bodies – or natural-technical objects of knowledge and practice – such as cells, molecules, genes, organisms, viruses, ecosystems, and the like. The work processes also make humans into particular kinds of subjects called scientists. (Haraway 1997: 141)

and further, "[t]he body is simultaneously a historical, natural, technical, discursive and material entity" (ibid: 209). The body is the locus where technology and discourse meets, where technological changes gain acceptance (by being taken into use) or rejection. Technology does more than affect cultural habits; it also changes our bodies as Latour points out: "even the shape of humans, our very body, is composed in large part of sociotechnical negotiations and artefacts" (Latour 1994: 64).

The cyborg, as a defining term, is new, but it can be argued that humans have always been cyborgs.[59] It is only when technology becomes so advanced as in the current situation that it becomes increasingly obvious how human life is tangled up in technological structures, and the subtle ways in which they define our very being. Various sections in chapters 5 and 7

---

[59] Donna Haraway reminds us of how concepts like the cyborg appear historically: "we are now accustomed to remembering that as objects of knowledge and as historical actors, 'race' did not always exist, 'class' has a historical genesis, and 'homosexuals' are quite junior" (Haraway 1991: 161).

explore the different modalities of bodily interaction with technology, in our case musical instruments, namely: acoustic instruments (or physical things); electronic/electric instruments (physical but mapped); digital hardware (mapped prosthesis); and software (mapped semiotics). From this understanding of the body as naturally extended – a naturally-born cyborg – we will look at how these instrumental modalities contain different expressive potential, with regards to both bodily and conceptual expression.

3.6 The Trained Musico-technological Body

Much has been written about skill acquisition in the fields of phenomenology, athletic training, musical pedagogy, military training (such as aerial warfare), dance, martial arts, and other activities that crave a dexterous and focussed mind-body unity. Surveying that literature would be sidetracking, but what is essential to this thesis is to gain an understanding of the state where the trained musical body is in a state of focus, where mind and body become one in an immersive embodied activity, a condition that has been written about elegantly as the state of *flow*.

3.6.1 An Experience Called Flow

In the early 1970s Mihaly Csikszentmihalyi, through psychological research, began work on defining an experience he called “flow” (Csikszentmihalyi 1996). It is what it says on the tin, typically an experience most humans will know through performing practised bodily actions, but could also involve mental activities such as chess or programming. Flow is also a state of mind that is emphasised in various Buddhistic traditions such as Zen, where the mind and the actions of the body are ideally one and the same: mindfulness, the state where the mind should not wander away from the domain task, whether it is performing martial arts, calligraphy, walking or breathing. Csikszentmihalyi describes nine elements that make up the state of flow:

1) Clarity of goals. The possible moves are known
2) Immediate feedback to one’s actions
3) Balance between skills and challenge
4) Action and awareness merge. Heightened awareness
5) Distractions are filtered out from consciousness
6) There is no conscious worry of failure
7) Self-consciousness disappears
8) The sense of time is distorted
9) The activity becomes autotelic (ibid: 111-113)

People attain the state of flow though skilled repetitive actions. We can imagine the musician that has practised a scale for weeks with her fingers on a fretboard and suddenly in a live setting she improvises through that scale in tune with the rest of the music, forgetting time and place, becoming one with the music. Or picture the masterful ping-pong player with his repetitive actions, not looking at the ball or his spade, but rather focussing on the whole environment; the movement of the ball, the body movements of the opponent and the sound of the ball hitting the table and the spades.

Computer systems are rarely designed for the state of bodily flow. And why should they? Flow occurs when the trained body interacts with some technology or material in a situation where the challenge is neither too high (leading to tension and/or frustration) nor too low (leading to boredom and routine). The computer excels at repetitive actions that we would not like to perform ourselves. Thus, routine tasks can be batch-programmed on a computer. It does best what humans are worst at: repetitive actions. Soetsu Yanagi describes the situation with more sophistication: "No machine can compare with a man's hands. Machinery gives speed, power, complete uniformity, and precision, but it cannot give creativity, adaptability, freedom, heterogeneity. These the machine is incapable of, hence the superiority of the hand, which no amount of rationalism can negate" (Yanagi 1972: 108). An email can be sent to 100 recipients, a text can be copied to multiple documents, a pattern can be pasted into a picture or notes repeated on a musical timeline. One of the most important *rationale* of digital tools is to automate repetition, end boredom and save time.

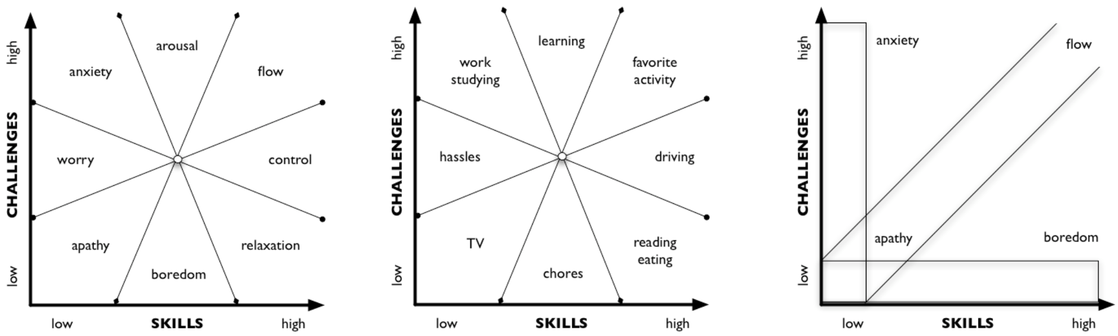

Figure 3:3. A representation of the "emotional space" of flow, as found in Csikszentmihalyi (2003)

However, flow is an important topic for HCI as ideally the tool should be as transparent as possible for the task ahead. This type of mental flow can, for example, typically be found with programmers who lose sense of time and space when coding. Much of this is due to the complexity of the programming task, where the programmer has to increase his or her capabilities for cognitive load in order to mentally represent all the variables of the system. Programming language design addresses these issues in various, yet different, ways and the design of object orientated programming or of tangible programming (Blackwell 2003) are good examples. A common practice with programmers is to have pen and paper at hand and offload unimplemented parameters onto the paper until they are needed.

As seen by Figure 3:3, flow is a concept that defines the sensation of being challenged, but still feeling in control over the situation. It typically happens in embodied activities where trained bodies perform tasks that are engaging and practised, yet not rote.

3.6.2 The Hands of the Way: An Account of Non-Symbolic Learning

In jest, this heading reverses David Sudnow's famous book title *Ways of the Hands* (2001). But Sudnow's thesis is precisely that: an inversion of daoist knowledge of mindful focus on the act that is being performed. Eastern philosophy's point of departure in all mental and physical exercises is based on a focus where mind and body becomes one, exemplified in the fact that eastern musical performance often involves meditative techniques before and during the performance. Sudnow is an ethnographer who decided to learn to play jazz piano as a mature person, a combination of facts that is quite rare, and which yielded an interesting account of the process of learning a musical instrument. His account is phenomenological, so much so that "a copy of [Merleau-Ponty's] *Phenomenology* always remains close at hand" (ibid: 131). As we have seen above, the trained musical body is a body that has incorporated musical knowledge through the interaction with an instrument. Sudnow's account is an elaborate journey as knowledge is incorporated into the body (the 'habit-body') through habituation with a musical instrument. The learning is embodied, the formal rules are blurry and vague at the start but become real through bodily practice. (Gibson 2006). The trained body is one that has practical (or tacit) knowledge, which can be translated to theoretical knowledge depending upon the vocabulary and theoretical skills of the person.[60]

> It is obvious that tacit knowledge changed to theoretical knowledge is no longer the same. Theoretical knowledge is not used to make a beautiful tone on the violin. It is knowledge about the knowledge one uses to make the tone; that is, a description of the knowledge used. The description is not for the purpose of sound production, but rather to establish relationships which can be named, ordered and compared with other experiences. (Kvifte 1989: 67)

We are faced with the problem that the enactivists had to face, namely that our learning is non-symbolic, but we still are able to think using symbols. Dennett and Clark deal well with that problem as we saw in section 3.3.2.3. And Edwin Hutchins hit the nail on its head in his observation that "[w]e must distinguish between the proposition that the architecture of cognition is symbolic and the proposition that humans are processors of symbolic structures. The latter is indisputable, the former is not" (Hutchins 1995: 369).

Another cognitive scientist, Hubert Dreyfus, has long been involved in trying to

[60] We will explore in the next chapter how the computer differs from acoustic instruments in terms of interaction. Instead of the bodily interaction, one of tacit knowledge, we now have to possess the skills of conversing in a symbolic language with the computer. It is the language of the mind rather than the body, to metaphorise this distinction, although such dualism should not be taken literally.

understand skill acquisition from the perspective of cognitive science. In their book *Mind Over Machine* (1986), Dreyfus and Dreyfus argue against the view that explicit rules form the foundation of embodied actions. One learns basic rules (although fuzzily understood) initially when embarking upon learning a new skill, but those are points of reference from which the player (of an instrument or the game of chess) proceeds in order to learn the actual task at hand. What the player learns through practicing is a long list of experiences, situations, strategies that can be applied in actual situations. This experience cannot be described in terms of a finite set of rules and is therefore not explicit but implicit (or tacit) knowledge. The argument is basically one against representation as the foundation of our cognitive skills. Knowledge is here portrayed as habituated, embodied and contextual knowledge. There does not have to be a representation of the final goal as we are able to embody such knowledge on a subsymbolic level.

In a later paper, Dreyfus (1996) becomes interested in exploring what Merleau-Ponty means by the concept of "habit-body" and his claim that a

> movement is learned when the body has understood it, that is, when it has incorporated it into its 'world', and to move one's body is to aim at things through it; it is to allow oneself to respond to their call, which is made upon it independently of any representation. (Merleau-Ponty 1962: 139)

For Dreyfus, the challenge here is to show how this can possibly happen without resorting to the use of representation and explicit rule use. He outlines five stages of skill acquisition: *stage 1 (Novice):* the instructor presents the task at hand in a "context-free" manner, thus helping the novice to recognise various objective facts and features relevant to the skill. The student acquires rules for determining actions based upon those. *Stage 2 (Advanced Beginner):* After seeing enough examples, the student learns to recognise them. She begins to cope with real situations. Rules and embodied knowledge start to merge. *Stage 3 (Competence):* Overview increases, but there is yet no sense for what is of main importance in the task, which can give the learner a feeling of exhaustion or nervousness. This increased overview also means that the learner resolves to the use of few rules/actions that she knows well. *Stage 4 (Proficient):* Knowledge is incorporated through the habituation of learning. "Proficiency seems to develop if, and only if, experience is assimilated in this atheoretical way and intuitive behavior replaces reasoned responses" (Dreyfus 1996:unpaginated). When a critical situation appears in the performance, the instrument appears suddenly anew; it becomes present-at-hand, and "to decide, [the performer] falls back on detached, rule-based determination of actions" (ibid.). *Stage 5 (Expertise):* The expert does not have to decide what to do. The knowledge is incorporated, and she can react to a vast multiplicity of situations without having to calculate possible strategies or consult a rule. She has incorporated intuitive knowledge. Dreyfus and

Dreyfus acknowledge that there is a conscious part of the mind involved in all learning. “A monitoring mind must decide when results justify reinforcement of chosen actions” (Dreyfus & Dreyfus 1986: 40) but they also claim that there are moments where all such monitoring ceases, namely when the practitioner reaches the state of *flow*, as defined by Csikszentmihalyi above. However, “from our perspective ‘flow’ is not a sixth stage of the mental activities that produce skilled behavior but rather the cessation of the monitoring activities that normally accompanies the higher levels” (ibid.).

This non-representational account of learning implies that what the learner learns through experience is not *represented* to the mind, but *presented* as ever more finegrained situation that needs increasingly refined response. “What one has learned appears in the way the world shows up; it is not represented in the mind and added on to the present experience” (Dreyfus 1996). Dreyfus sees a correlation here with feed-forward artificial neural networks where memories do not have to be stored in the brain and retrieved as symbolic data in the actual situation. Rather, the network is trained according to fitness and when a certain input is entered into the system, the network returns a solution much faster than any symbolic system would be able to. This explains the incredible speed expert chess players are able to display when playing a complex game.

Dreyfus here gives an account of how modern cognitive science explains the process of how equipment becomes “ready-to-hand” through embodied interaction with the tool, i.e., the establishment of a non-representational and non-theoretical relationship with the tool. Similar approaches are provided by the “sensori-motor couplings with the world” account by Thompson (2005), the theory of action in perception by Noe (2004), or the idea of “body-schema” as “the style that organizes the body as it functions in communion with its environment” (Gallagher 1986: 549). It should be noted that this research topic is highly dynamic and active at the time of writing, so various views for and against this way of describing human cognition, and learning in particular, are up for testing. It would be too big a task here to analyse this discourse. For now, these authors are sufficient, as references of the role non-representational habituation of skills take place when the instrumentalist incorporates the knowledge of his or her instrument.

3.7 Conclusion

This chapter has focused on the body and its relationship to the technological artefacts it perpetually uses in all aspects of life. Through a phenomenological approach the attention was directed away from more abstract theorising about technology and pointed to the human body and the human experience as a being-in-the-world – a world full of technological equipment that constitutes its being. Ihde’s account of technology as prosthesis was presented, demonstrating how tools give us different relationships, such as embodiment or hermeneutic, with the world. Two important and highly relevant theoretical models were introduced: enactivism and the theory of the extended mind, both of which will have descriptive value later in this thesis. These theories show how the human body is consistently defined by its use of environmental props, either as an autopoietic system that through enactive performance builds its world, or as a body whose cognition takes place to a large extent outside the body, making use of supplementary tools in the environment. Various theories of embodied learning were explored, demonstrating how knowledge can be attained through methods of incorporations or inscriptions. Skill acquisition was succinctly defined as a non-representational activity of the brain, where bodily incorporation through habituation yields expertise that can result in the state of flow, a state of mindfulness where mental focus and bodily actions unite in a single action.

As exemplified with excellence by Stiegler (and in another discourse by Dawkins (1982)), we humans have been defined by our tool-use since the origin of our species. The tool and the body became one in this modality of prosthetic extensions of the body. For us, the world is a different world with a tool in the hand, and the tool is a different tool in our hands (as Latour shows). However, as Marx observes in the *Capital*, with machines things change. The human becomes a different subject, often transformed into an operator rather than creator, a follower of instructions rather than its generator. “The difference between tool and machine is that in the case of a tool, man is the motive power, while the motive power of a machine is something different from man” (Marx 1887 volume 1: section 1). Although Marx talks here about what drives the machine or the tool, he is already making a distinction between a tool and a machine with reference to the human body.

Marx’s machines were mechanistic and industrial. With the advent of the computer our tools enter yet another stage, where our machines becomes symbolic. Semiotic systems are now used to communicate with the machines and not physical gears, levers and couplings. Instead of incorporated knowledge used to drive tools, we now turn to the language of general inscriptions. Our tool-use is characterised by this transition from the tacit knowledge of the body into the abstracted symbolic realm of the mind. The tools have transformed to mechanical machines to information machines, and we experience a shift that could be defined as the *inception of mapping*. In mapping, our relationship with the environment occurs through symbolic couplings. Humanity has therefore gone through three stages in their development of tools: tools proper (flints, hammers, flutes, guitars), machines (mills, cars, mechanical music machines, pianolas), and automated, informational, or cybernetic machines (computers). The next chapter will explore what mappings and embodiment in modern machines mean to human cognition, supporting the definition of the *epistemic tool*.

Chapter 4

Epistemic Tools: Thinking Through Technology

Having established the complex and symbolic nature of digital technologies, it is relevant to analyse how systems of knowledge and representations are inscribed into technological artefacts. After a study of ethnocomputing and an excursion into externalist thinking in HCI, this chapter introduces the concept of epistemic tools*. Epistemic tools are defined as technologies that influence our thinking due to intrinsic mechanisms that can potentially be defined as cognitive. Here digital technologies are seen as cultural machines (affecting culture through influence on their users) and the chapter ends by discussing the relevance of a new research field called software criticism.*

4.1 Introduction

At the beginning of history philosophy separates tekhnē *from* epistēmē*, a distinction that had not yet been made in Homeric times. (Stiegler 1998: 1)*

After a thorough excursion into the philosophy of technology [chapter 2] and the relationship between technology and the body [chapter 3], we are now prepared to start looking at the unique qualities of the computer as a new type of tool, an epistemic tool, whose nature of interaction and design is essentially different from the analogue tools we are used to. This chapter is an elaboration of the theory of epistemic tools. It will demonstrate how the embodied relationships of the semiotic machine differs to those of analogue tools.

Plato is always a useful point of reference. In *Gorgias* (450b) Socrates distinguishes between two types of *techne*: one that consists mainly of physical work in crafts and requires little amount of language (such as wood-carving,[61] painting or sculpture). The other type is the *techne* of the word, where language is involved (such as arithmetic, logistics or astronomy).[62] In our context we could translate these types into the incorporated knowledge habituated by the body through instrumental practise, versus the more symbolic relationship we have with the digital musical tools of the computer. The focus of this chapter is the second category of *techne,* i.e., the craft that is linguistic, logical, calculative, not based on physical motor memory, as studied in the previous chapter, but rather the activities of the mind, perhaps with the help of physical

[61] Mitchell shows that initially the stem *techn-* originates from the Greek word for "wood-work" or "carpentry" (Mitchell 1994: 117).

[62] What is of another nature is *poiesis*, the craft [*techne*] of poetry by virtue of divine inspiration (*Ion* 533d). As opposed to craftsmen or astrologists, the poets can neither describe their craft nor teach it.

devices to augment the cognitive function, such as in astronomy or arithmetics. Of a particular interest is the comparison of computers, as semiotic machines in which we build musical tools, with analogue technologies such as acoustic instruments.

In a later work, *Philebus*, Plato divides the arts (*techne*) into those arts akin to music (imprecise) and those akin to carpentry (precise). He nominates the mechanical crafts as primary to artistic bodily crafts. This is because mechanical arts can be measured more precisely and are quantifiable. Bodily crafts, such as music, are not as precise: "in music when it adjusts its concords not by measurement but by lucky shots of a practised finger – in the whole of music, flute playing and lyre playing alike, for this latter hunts for the proper length of each string as it gives its note, making a shot for the note, and attaining a most unreliable result with a large element of uncertainty" (*Philebus* 56a). Plato distinguishes between *techne* and *episteme* (a distinction that is still represented in contemporary scientific discourse in the form of the dichotomy between technology and science), but in early Plato, these two terms were closely related and inter-connected. In later Plato, as seen in *Philebus*, the distinction between the categories strengthen to a certain degree. Here some of the mechanical arts become pure knowledge (*episteme*) as they are about numbers and quantifiable logic (arithmetics versus carpentry) whereas other and secondary arts are more bodily, imprecise – thus less epistemic. [63]

This thesis defines the modern day digital musical instruments as a class of *epistemic tools*. Hard-line distinctions should not be drawn between analogue and digital, as many physical tools are epistemic in their nature (such as the astrolabe used in navigation or the abacus used in calculation), and many digital tools have been designed for embodied and non-symbolic interactions (although they are always built with symbolic languages and not "knowledgeable matter"). Our question of epistemic tools could be posed thus: if much of our thinking happens "in the wild" (Hutchins 1995), external to our body, as a sociocultural process that uses technology as external scaffolding of our cognitive processes, and if our learning is largely a process of incorporating knowledge in a non-symbolic way through enactive relationship with our tools and environment, how do computers, as necessarily symbolic devices, enable, produce, maintain, support, augment and limit our cognitive processes? In what sense do epistemic tools define our cognitive horizon and to what extent do they constrain our expression due to the low bandwidth they pose in terms of embodied interaction?

This chapter thus explores the question of how digital technology affects our creativity and the nature of our relationship with that technology. We have already considered the dialectic structure of technoculture, and how technology can serve as a programme for action,

---

[63] However, it is only with Aristotle that we start to get a more modern distinction between praxis and knowledge (the one that is epitomised in our education systems), although he recognises the implicit knowledge of all embodied skills.

which we either accept or reject (or re-interpret) [chapter 2]. We also looked at how technology defines the body and effectively extends human cognition into the environment. Therefore, from the context of the creative tool and aesthetics, it becomes important to investigate how the external tool influences the cognitive process (here processes of aesthetic or compositional decisions). Next on the agenda is to inquire into the nature of designing systems whose interaction is symbolic, thus intrinsically cultural. Having introduced ecological psychology and its ideas of affordances and constraints which will be important later in the thesis; the problems of culture, with regards to the design and reception of computer systems are investigated. A short history of HCI is presented in order to elucidate how these ideas have been dealt with in that particular branch of research. Then, finally, we are ready to explore the concept of epistemic tools and its relevance to music technology in general and musical instruments in particular.

4.2 Ecological Psychology

In Heideggerian language, we are "thrown" into the world of equipment. As beings-in-the world the world appears to us as a reservoir, or an assemblage, of objects that we *relate to* in various ways. Things or ideas that are invisible at one stage can become of vital importance in another. This relationship between the human and the world has been studied in psychology under the terms of *ecological psychology.* The ecological approach to cognition was developed in the 1960s by the psychologist James J. Gibson who pointed his attention to the way the environment affects the human. The HCI research field has incorporated many important concepts from his work, in particular *affordance* and *ecological constraint.* The former is more commonly used within HCI, but in the context of musical interfaces, the latter might be of a higher importance.

4.2.1 Affordances

In order to identify how various technologies affect the user, it is useful to resolve this using the two partly inverting terms affordances and constraints. Gibson initially defined affordances as: "The affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill" (Gibson 1979: 127). In this definition, affordances are the properties of the *relationship* between the environment and the agent (human or animal). The relationship consists of a mapping between the *properties* of the environment to the *potential actions* of the agent. An instrument like the violin affords certain actions to the human that it does not afford to a dolphin for example. Donald Norman, working in the field of HCI, introduced the idea of *perceived affordances* (Norman 1999), meaning the properties that the agent perceives as possible actions upon an object. This is a narrower definition of affordances, as Gibson claimed affordances exist independently of the agent's perception of them, a view supported by Gaver (1991) who talks about perceptible, hidden and false affordances. Norman's view has been influential in the field of HCI as the design of any system can be seen to be largely about providing affordances to its user.

Affordances have also been defined as entirely subjective. Vera and Simon define affordances as "carefully and simply encoded internal representations of complex configurations of external objects, the encodings capturing the functional significance of the object" (Vera & Simon 1993). Costall (1995) extends this definition of affordance to also signify a social construction, something that people learn from each other in every culture. The affordances of a physical object in one culture will thus have different affordances in another. However, to get around the relativism of the social constructivist argument, he proposes the "canonical affordance" (Costall 1997) of an object as derived from its name (one hammers with a hammer, wrenches with a wrench, and thus "computes" with a computer).

Ian Hutchby (2001) disagrees. For him, affordances are important precisely because they are not culture specific. They can provide objectivity; constituting a common platform that exists beyond all cultural difference. Hutchby claims that, as opposed to the views of Grint and Woolgar (who propose the reading metaphor of technology [see section 2.3.4]), technology exists in a real, physical world that is not reducible to textual interpretation: "Different technologies possess different affordances, and these affordances *constrain the ways that they can possibly be 'written' or 'read'*" (ibid: 447). A musical instrument thus affords certain ways of playing, but at the same time allows for a cultural reading of its expressive scope: we can see how the affordances of certain instruments change when they are used in different cultures.[64]

The concept of affordance used in this thesis is a compromise between the objective and cultural stances. From the perspective of phenomenological enactivism, it is questionable that we could ever talk about an affordance that is not culturally grounded. (For example, a chair might afford sitting to a member of the medieval Japanese society, but in reality that person would have been much more likely to sit on the floor). More importantly, as opposed to Vera and Simon, we will not see affordances as representational structures of the object in the subject's mind, but rather as potential applications of the object derived from an embodied *relationship* with the object in the sense discussed in the last sections of the 3rd chapter.

---

64 This is exemplified by Swift (1990) who shows how the violin (a European invention) is "read" differently by performers of Indian music. Furthermore, cultural conventions and languages can affect people's capabilities of using an instrument. For example, Kevin Austin (in a cec-conference mailing list entry on October 11th, 2006) observes how the embouchure of a brass player is affected by her native language. (As most English people do not have the retroflex /r/, they are not able to generate certain sounds on the brass instrument that, say, Scottish or Spanish performers can.

4.2.2 Constraints

In her book *The Creative Mind,* Maggie Boden defines constraints as one of the fundamental sources for creativity: "far from being the antithesis of creativity, constraints on thinking are what make it possible." And she continues: "Constraints map out a territory of structural possibilities which can then be explored, and perhaps transformed to give another one" (Boden 1990: 95). For Boden, the continuity of constraints in our culture gives rise to the possibility of an idea to be recognised as creative. All culture is founded on constraints – they are the rule-sets that maintain the dynamic unity of the culture. The constraints can be implicitly understood and explicitly formulated. The physical constraints of the violin defines its expressive scope and the theoretical constraints of a compositional system (such as the 12 tone system) define the compositional scope. Furthermore, the way the violinist plays the composition is highly dependent on cultural constraints. Norman's elucidation of physical, logical and cultural constraints are important here. *Physical* constraints are constraints of what can be performed with the physical environment (such as mouse cursor not exiting the screen). *Logical* constraints are the ones where human logic informs us inductively about the environment, and they are related to cognitive modelling. *Cultural* constraints are conventions shared by a cultural group. "A convention is a cultural constraint, one that has evolved over time. Conventions are not arbitrary: they evolve, they require a community of practice" (Norman 1999).

Patricia Stokes defines constraints as *task* constraints on the one hand and *subject* constraints on the other. The choice of materials and processes from a specific focus of the subject define the overall *goal* constraint (Stokes 2006). Marcus Pearce and Geraint Wiggins refer to the work of John Sloboda and define the working constraints of a composer into three categories: *stylistic constraints* – where the composer is working within the limits of a specific genre or style; *internal constraints* – the logical possibilities of how the piece can progress according to the rule set that has been implicitly or explicitly set; *external constraints* – the need to be sure that the piece is physically possible for a performer (Pearce & Wiggins 2002).

In the context of this research we will provide an alternative model of constraints, one that focuses on the philosophical-technological relationship between the human, and the tool, and the social context in which they exist. In this model *subjective constraints* refer to the expressive limitations that face the thinking, creative, performing human. This person is constrained by the technology at hand and the cultural values adhered to. *Objective constraints* are the physical limitations of the environment or physical tools, such as not being able to play very high pitched notes on a double bass. These constraints are also to be found in the virtual world, where software is designed with a careful detail to performative and expressive limitations. *Cultural constraints* are the conditions in which both technology and ideas exist. Technology is defined by culture but retrospectively defines it as well. Artists are defined by the constraints of the culture in which they exist. A musician is conditioned by all of these constraints to differing degrees. Often the musician has problems of breaking the boundaries of the expressive scope of the instrument, and these problems are partly due to the long training he or she has had in the particular musical culture which defines the expressive or imaginative constraints in the player. Here the musician might reject the script of the instrument and reinterpret it according to his antiprogram [see section 2.4.1]. An example is the "table top guitar" where an (electric) guitar is used in a way it was not designed for. Cultural constraints appear, for example, as styles in music where originality is often defined as the capability of expanding or transforming the particular style (Boden 1990). We will explore further in later chapters how the objective constraints of digital and acoustic instruments affect musical expression.

4.3 Computers as Representation of Thought

We have already [chapter 2] elaborated on the thesis that technologies are non-neutral. They organise, select and focus the task and the environment (or the material of the work) through various transformations. These are what Don Ihde calls *amplification-reduction* transformations (Ihde 1979: 56) [section 3.2.4]. By selecting a technology, one is choosing a background theory, a specific world-view, and thus potential outcomes. This selection involves what kind of perceptual resolution we want to engage with the world, what type of embodiment and theoretical inscriptions. Technologies conceal and reveal. Technological artefacts necessarily amplify certain possibilities of experience whilst simultaneously reducing that very experience.

Throughout the history of computing, computers have not served as extensions of our body (as embodiment relation) but rather extensions of our mind (involving a hermeneutic relation with the world). When "mind" is used in this context, we are observing that the computational technologies are creating a language, a system with which to think. It changes many of our common habits of interpreting (or being in) the world. The aim is not to introduce yet another dichotomy between the body and the mind, but rather to point out the change in activities that are needed to perform the tasks at hand. A quick example might clarify this point. When clocks became digital (and time was represented as numbers instead of rotating arms), a certain prominent understanding of time as linear flow yielded for an understanding of time as a succession of moments.

> The person who awaits the train, who once could glance at his watch and *see* that it was yet ten minutes until arrival time by *seeing* the relation between the pointers and the span, now sees only the number and must infer or calculate the span. This is to say that the mental operation for telling the time changes, even if unnoticeably, with the digital clock. (Ihde 1983: 38)

Instead of the sensation of continuous flow we get discrete moments. Instead of the embodied and analogue continuity of rotating arms we now get a numerical display where we have to use our cognitive faculties differently than with the analogue clock. The cognitive load is increased as we need to subtract, say 48 from 60, in order to know the clock is 12 minutes to six. Five might even be displayed as 17, which requires an extra operation of subtracting 12 from the hour. This example is illustrative. It reveals how our interactions with the computer become increasingly symbolic, discrete, and hermeneutic. It is not our body but the mind that primarily interacts with the computer – through engaging with the symbolic structure of the software. The interaction becomes a mental habituation, so to speak, one that requires the user to have a primarily disembodied knowledge of the system, although it should be emphasised that there is also embodiment at stake when working with software. This example illustrates what I will define as *virtual embodiment* [section 7.2], due to the nature of mapping, consistency in conventions, and relativity of representational scales in the computer as medium.[65]

When dealing with technological artefacts that are representational to the degree computer software proves to be, we are inevitably transposed into the realm of text, of hermeneutics, of *culture*. It is therefore appropriate to explore how computer technology affects culture in various ways and how the field of HCI has evolved to an understanding of these issues. Through the introduction of ethnocomputing and a short historical account of HCI we will arrive at the concept of epistemic tools.

4.3.1 Ethnocomputing – Computers as Cultural Systems

Like the machine, the human of the industrial age is dependent on the technical system, and serves it rather than making it serve itself; the human is the "assistant", the auxiliary, the helper, indeed the means of technics qua system. (Stiegler 1998: 24)

Apart from the general *software studies* research programme [presented in section 4.5.2 below], another important research, called ethnocomputing, is being conducted on the cultural conditions of computer science. Ethnocomputing is defined as "the study of computational phenomena within a culture" (Tedre 2002: 43). It attempts to bring knowledge from social constructivism or science and technology studies [section 2.3] into the study of computing. The main areas of ethnocomputing are defined as: "(a) data structures: organized structures and

[65] This will be explained later, but arbitrary mapping refers to how how a human gesture (movement with intention) is always arbitrarily mapped to a specific action when using tangible user interfaces with the computer. By "consistency in conventions" I mean facts such as the File menu normally being at the top left in most software on most operating systems, no matter how they differ otherwise. And finally, by "representational scaling" it should be conveyed that objects that are interacted with in one mode (with the mouse for example) can be represented in different mode by a keypress or the click of the mouse. If the objects are graphical representations of some data structure, those representations could be rotated, zoomed into, inversed, normalised, or subject to any algorithmic manipulations at the will of the designer.

models that are used to represent information; (b) algorithms: ways of manipulating organized information; and (c) physical and linguistic realizations of data structures and algorithms: devices, tools, games, art, or other kinds of realizations of computational processes" (Tedre & Eglash 2008). The focus of ethnocomputing until now has been largely on how computational concepts affect cultural practices, but less on how cultural practices influence the designs of computer based systems. This research field is still in its infancy, and below I will make my own contributions to what I understand as ethnocomputing.

Anyone who speaks more than one language, in particular if those languages are of different linguistic families, knows how differently languages portray the world. A language is a world-view. This observation is the basis of a theory called the Sapir-Whorf hypothesis which states that different languages portray both space and time differently.[66] The Sapir-Whorf hypothesis claims that grammatical categories of the language determine the speaker's behaviour: "We cut nature up, organize it into concepts, and ascribe significances as we do, largely because we are parties to an agreement to organize it in this way – an agreement that holds throughout our speech community and is codified in the patterns of our language" (Whorf 1940: 213). Nietzsche has a similar stance where he argues that physics is an interpretation of the world and not its explanation. The sciences reject phenomenological experience for objective language and they "do this by means of the pale, cool, gray, conceptual nets which they threw over the colourful confusion of sense, the rabble of the senses" (Nietzsche 2004: §14). Rather more recently Bowker and Star (2000) have argued that there are certain problems in the design of computer systems that relate to the act of classification. When we classify and categorise our external world in order to represent it through the machinery of information technology, we are inevitably reducing the world. Such reduction is inevitably contingent and messy, with an abundance of ethical and political implications. Human action is displaced into representation and thus creating strata of complexities and interdependencies that limit the agent.

This nature of categorisation and abstraction of human knowledge (both know-that and know-how) and actions is the essence of computer software, a fact that is very apparent in the realm of musical software tools. Software is, as Bowker and Star point out, a "contingent" and "messy" classification that has diverse implications and, in our case, aesthetic, cultural, and culture-political effects. As Latour (1987) so elegantly demonstrates, objects establish themselves as black boxes through repeated use and in the process their origins disappear. The object is naturalised[67] through heavy use, a fact that makes Bowker and Star observe: "[t]he more naturalized the object becomes, the more unquestioning the relationship of the community

[66] This hypothesis has many followers and critics. A notable criticism was voiced by Stephen Pinker (1994) who, in a Chomskyan manner, argues for a universal underlying structure of language.
[67] Or "concretisised" or "pointilised" in actor-network theory lingo.

to it; the more invisible the contingent and historical circumstances of its birth, the more it sinks into the community's routinely forgotten memory" (Bowker & Star 2000: 299).

Translating this argument into our context, we observe that musical patterns and musical styles are blackboxed in software. Most people do not know why the standards, implementations, patterns, or solutions in musical software are there. At times they obviously limit musical expression, but at other such obviousness is concealed by the rhetoric, streamlined functionality and slick interface design of the software tool. Here we encounter yet another difference between the worlds of acoustic and digital instrument making: the acoustic instrument maker can freely re-invent or improve the instrument at any time with new materials, changing structures or adding/deleting features, whereas the software instrument maker is restricted by a huge infrastructure of operating systems, programming languages, protocols and interface limitations. This is a factor that involves social and technical elements well described by Bowker and Star: "Systems of classification (and of standardization) form a juncture of social organization, moral order, and layers of technical integration. Each subsystem inherits, increasingly as it scales up, the inertia of the installed base of systems that have come before" (ibid: 33).

Winner (1980) has demonstrated how technological artefacts have politics, but it is clear that they also have aesthetics, morals (Latour 1992) and style. A possible turn here is to the Marxian analysis of the machine and the way Marx defined technology as "frozen labour." Modern information technologies thus:

> embed and inscribe work in ways that are important ... [but] are often difficult to see. Where they are used to make decisions, or to represent decision-making processes, such technologies also act to embed and reify those decisions. The arguments, decisions, uncertainties, and processual nature of decision making are hidden away inside a piece of technology or in a complex representation. Thus, values, opinions, and rhetoric are frozen into codes, electronic thresholds, and computer applications. Extending Marx, then, we can say that in many ways software is frozen organizational and policy discourse. (Bowker & Star 2000: 135)

Much computer music is inevitably mechanical. The engine that typically drives the music is not a human, but a machine. The human is merely an operator of a machine that performs routine tasks, repeating and automating laborious and boring rote movements.[68] The digital

[68] It should be stressed that the intention is not to pass value judgements on the machine. Rather more interesting would be to investigate how the machine has inspired humans through the ages: arguments can be made that speed (increased with heightened technology) in transport or communication affects musical style. We see how music changed in Renaissance with the rhythms of horse wagons or how the style of noise influenced by contemporary information overload. One could also explore how machines in factories have influenced music and dance (tap dance is a good example (Pacey 1999) and another would be the 1920s New York Can-Can dance that drew inspiration from – and in many ways resembled – the assembly belt). Also, as already mentioned, the beatboxer is perhaps the ideal simulation of the technical object (the drum machine).

music software, where music is inscribed, organised, and played back from a mixture of data structures and algorithms, is essentially an engine. The underlying core of the system is organised in blocks of repetitious routines. As such, it is ideal for repetitions, perfect timing, standard dynamic of notes, etc.[69] What the digital system does not ideally lend itself to are tempo changes, variable dynamics, metre changes, flexible durations, (accelerando, crescendo, etc.) i.e., what the human excels at when performing music. These temporal and dynamic changes in the human derive typically from an emotional engagement with the music performed, a fact that is much expressed in classical music education. Needless to say, the machine is not (yet) able to experience emotions and it has proven a difficult task to simulate them (Juslin & Sloboda 2001). We therefore have very different ideals: the machine (time perfect, repetitious, logical, perfect, normative) and the human (flexible, fallible, instinctive, emotional, individual).

This portrayal of the digital musical systems as mechanical machines illustrates how the computer provides strong affordances for certain musical styles, but also how, conversely, it has created those very styles through people's musical explorations of computer music systems. It is important to note that it is not merely the internal mechanisms of musical software that are machinic, but also, in many cases, the interaction we have with the instrument. The experience of playing the acoustic instrument is that of a flow, as one undifferentiated and qualitative process of actions proceeding through time. Conversely, in screen-based digital musical instruments, the experience tends to be that of juxtaposed and quantitative moments in time where communicative actions are performed sequentially.[70]

As we have repeatedly seen in this thesis, human thought and action is constituted and supported by external technologies and the relationship is one of dynamic transduction through time where each side conditions the other. It has become clear that the way musical tools are designed serve some cultures and not others. A beautiful example of cultural differences in terms of how musical time is represented, is reported by Philip Glass, who had been studying Indian music with Ravi Shankar (and I quote a rather long passage as I find it important for an understanding of the issues we are dealing with here):

> I would explain the difference between the use of rhythm in Western and Indian music in the following way: in Western music we divide time – as if you were to take a length of time and slice it the way you slice a loaf of bread. In Indian music (and all the non-Western music with which I'm familiar), you take small units, or 'beats', and string them together to make up a larger time values.

[69] James McCartney, the author of SuperCollider, describes the effects of Patterns (lazy event streams) in his software: "with things like Patterns it is easier to use them to make rigid sounding music vs. more fluid timing, which I think is unfortunate, but that is the nature of computers" (McCartney 2008, personal communication).

[70] It should be stressed here that this applies to screen-based digital systems due to their unifocal nature. Many digital musical instruments, such as Waisvisz's *Hands* are multidimensional and multifocal.

> This was brought home to me quite powerfully while working with Ravi and Alla Rakha in the recording studio. There we were with the musicians sitting around waiting for me to notate the music to be recorded… [Ravi] would sing the music to me, and I would write it down, part by part… The problem came when I placed bar lines in the music as we normally do in Western music. This created unwarranted accents. When the music was played back, Alla Rakha caught the error right away. No matter where I placed the bar line (thereby 'dividing' the music in the regular Western style), he would catch me.
>
> 'All the notes are equal,' he kept piping at me…
>
> Finally in desperation, I dropped the bar lines altogether. And there, before my eyes, I would see what Alla Rakha had been trying to tell me. Instead of distinct groupings of eight notes, a steady stream of rhythmic pulses stood revealed. (Schwarz 1996: 115)

It is clear from this example that our language or notation systems for classifying the world or representing its structures differ between cultures. What the quotation also illustrates is that certain cultures – or as Glass phrased it: "all the non-Western music with which I'm familiar" – have problems with the Western rationality or Euclidian methods in representing space and time in music. This should have major implications for the designers of musical software that have to take accounts for different musical practices, not only in terms of other cultures but also subcultures within their own. Lucy Suchman raises some concerns regarding the directions of how organisational power flows in a recent publication:

> One obvious concern here is the asymmetrical directionality of those flows, insofar as those in the hyperdeveloped countries maintain a disproportionate hold over the distribution of at least computer-based information technologies, and remain largely ignorant of activities elsewhere. Those in the so-called developing countries, meanwhile, are inundated with the products, processes, and propaganda generated in the commercial, educational, and governmental centers of the high-tech North. (Suchman 2002: 139)

The commercial musical software we know today (all made in the West) perform abstractions and generalisations from the perspective of certain popular styles of Western music. There are innumerable styles of Western music that are not well supported or incorporated into musical software but more seriously there are whole continents whose musical traditions have been neglected. This trend is maintained by market laws and product design considerations of who the real buyers of the software are (young, white, middle class males). Deleuze and Guattari define the functions of the capitalistic machine thus: "There is a two-fold movement of decoding or deterritorializing flows on the one hand, and their violent and artificial reterritorialization on the other" (Deleuze & Guattari 1984: 34). Which we read like this: the decoding or deterritorialisation of embodied practices is abstracted into software (which is the result of a conditioned and market oriented hermeneutic reading of musical cultures on behalf of systems designers). This software is then marketed globally, thus violently imposing the artificial abstractions of a disembodied (and imagined) musical culture onto real musical cultures.

Howard Rheingold in *Tools for Thought,* picks up McLuhan's mantra of "the medium is the message" (McLuhan 1964) and asks what happens when our whole environment has become the medium?[71] Albeit an amusing thought at first, we have to ask what transpires in other cultures, or their "messages," when they start to live in a media environment entirely designed, implemented and produced for Western (or perhaps West-Coast) culture? For Ziauddin Sardar this is an enslavement, yet another colonialisation by the West that now, "[w]hen mental and cultural territories are exhausted, moves on to conquer the reality of Other people" (Sardar 2000: 734). Sardar observes how the Western culture's view of the body fundamentally differs from Asiatic cultures, which prompts him to state that it now "seeks liberation from the body by dissolving it into the machine" (Sardar 2000: 749). Even with an awareness of different cultural practices (that would have consequences in software from the level of musical patterns, interaction design, iconic design and general interface design) it would be difficult for Western software houses to work this out from their specific cultural standpoints as they would be likely to misinterpret the cultures they would be designing for, potentially falling into orientalistic traps (Said 1978).

> In the postmodern world where things have systematically become monuments, nature has been transformed into 'reserve', knowledge is giving way to information and data, it is only a matter of time before Other people and their cultures become 'models', so many zeros and ones in cyberspace, exotic examples for scholars, voyeurs and other interested parties to load on their machine and look at. (Sardar 2000: 736)

Above, we entered the realm of history, culture, and colonialism through the spectacles of Ziauddin Sardar. Things have changed since the time of Sardar's writing: India and China now have very strong software industries but we have yet to see musical software coming from those cultures that represent the indigenous musical traditions. There are many reasons for this lack of products.[72] One reason is that the software industry of these countries have until the mid 2000s been focused on supporting Western software houses that offshore their work to cheaper production environments. Another reason is that the young musicians in these countries that start to work in electronic music most typically "subscribe" to the musical styles, culture (and thus the tools) of where this music originates. They typically copy styles that are popular in the

[71] An observation that is reflected in Peter Weibel's last stage in the archaeology of media which he calls "The World as Interface" (Weibel 1996). In his account there is a logic in the evolution of the media in which, from the invention of photography, there is an increasing automation of production and an intensified separation between the content and the physical vehicle that mediates it.

[72] And there is a solution for it as well. Open source audio programming languages are typically built without musical references. This is by intention (McCartney 2002; Puckette 1996). It is therefore possible to support various musical traditions in these environments. Of course, we are still conditioned by the machinic nature of software as analysed earlier in this chapter.

West, although there are certain differences in emphasis, as exemplified by Japanese noise for instance. Yet another reason is the strong emphasis on tradition in the musics of these countries, which is an attitude that does not frequently lead to experimentation (due to the underlying compositional and performative philosophy). The emphasis of a tradition's continuity was extorted out of the Western tradition with modernism, resulting in a uniquely positive attitude towards self-criticism, change, and progress in the art world. The experimental stance inculcated by modernism explains the ease and naturalness of investigations and experiments in the arts, such as Western contemporary music's explorations of genetic algorithms, artificial intelligence and robotic performance as compositional and performative devices.

Definitions of musical creativity thus differ from culture to culture, within sub-cultures of a culture, or within the same culture over time. Leman (1999) shows how ideas of creativity have changed in Western culture over the last two centuries from the Romantic view of creativity as inspiration and ingeniousness to the more modern Rationalistic view of creativity as something that is in abundance and which we can manufacture with the right technical means: "The technological environment implies that creativity can be controlled and guided by rational thought. Machines become useful as extensions of musical creativity, just like electronic calculators are useful in taking over some parts of mathematical reasoning" (ibid: 288). Leman here points at how the romantic idea of creativity has yielded for a more productionistic and less metaphysical conception.

But the adoption of technological work processes comes at a cost: the composers and musicians that use technology in their work have to learn a highly complex set of skills from acoustics, psychoacoustics, electronics, software control, audio synthesis and often computational modes of creativity. This habituation into a culture of technology by musicians unavoidably affects their thinking processes, and they learn to think in terms of the machines. And the musical results of this thinking are obviously a good proof of this claim, as an excursion into the world of modern nightclubs would clearly show. These are issues that will be dealt with further and more in depth in Part II.

4.3.2 A Short Journey through HCI : The Shaping of our Understanding of Tools

The field of computing that has dealt most extensively with the problems of cognition and embodiment is human-computer interaction (HCI). In short, it deals with the problem of how to formalise human cognition and agency in order to devise more intuitive and ergonomic interfaces in computational systems. HCI is a transdisciplinary field that combines psychology, ergonomics, computer science, design and to a certain extent sociology. As a research programme it involves the creation of methodologies and processes to design interaction and interfaces, but also ways to evaluate those through psychological experiments and user tests. Chapter 6 describes how the ixiQuarks have been created as experiments in human-machine interaction, and evaluations performed in order to explore the research questions of this thesis will be reported on. Much good work has been written on the methodological and evaluative techniques used in HCI (Sharp 1994; Sharp et al. 2002; Jacko & Sears 2003; Shneiderman & Plaisant 2004; Rogers 2004), and it is not necessary to reproduce that here. However, what is of interest to us is a short philosophical tracing of the ideas that have changed in the course of HCI as a scientific study as it relates to the concept of epistemic tools. In the following subsections we will follow the transitional phases of HCI from Cognitivist HCI, to Phenomenology, Ethnomethodology, Experiential Design, Technomethodology, and finally to Semiotic Engineering.

4.3.2.1 Cognitivist HCI

Chapter 3 provided a short view of cognitivism as one that historically ignored the role of the body and the environment in cognition. Computer design originated at the same time as the flourishing of cognitivism. These two fields went hand in hand and the analysis of computer technology at the time played an active role in forming scientific thinking, providing important metaphors for cognitive science. Again, we are reminded of Latour's claim that science and technology cannot be separated – thus technoscience – but what also comes to mind is Heidegger's insistence on technology preceding science (which is also a fundamental stance in Stiegler's idea of epiphylogenesis). In the early days of the computer, the model of interaction between the human and the computer was based upon a view of the human as a symbol processing machine, not so different from the machine that the human was interacting with. The communication between the two therefore should be as logical and straight-forward as possible, a fact that we can see in the design of the command line interface of early UNIX systems.

But how did we get to a situation that reduces human agency to mere symbolic operations? To answer this question we would have to revolve back into the history of mechanics and formal logic. Maggie Boden has written an exemplary overview of this history (Boden 2006) but it suffices to mention here that the industrial revolution and progress in formal logic enabled Charles Babbage as early as 1822 to build an analogue computer. Formal logic is a system allowing the expression of complex propositions through the use of simple set of symbols. By removing the semantic meaning of the symbols and concentrating on their syntactic structure, formal logic is therefore easily mechanised. Babbage's assistant, Ada Lovelace,[73] realised that the machine could be programmed and thus able to perform infinite amount of calculations that had not been thought of when the machine was designed.

[73] Not only was she the first programmer (with a programming language named after her), but she also had interesting ideas of computers composing music: "Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the [Analytical] Engine might compose and elaborate scientific pieces of music of any degree of complexity or extent" (quoted in Roads 1996: 882).

Philosophical theories such as logical positivism (the Viennese school), the early work of Wittgenstein, together with the work of Russell and Whitehead helped to create an understanding of language as an independent logical symbolic system with an indexical referentiality to the world. In the 2nd World War, Turing developed ways to use formal logic as abstract machines that would later be implemented in computers. This made possible "the emergence of a new level of analysis, independent of physics yet mechanistic in spirit… a science of structure and function divorced from material substance" (Pylyshyn 1986: 68 quoted in Clark 2001: 11). This was a view stating that if "the mind is a sort of computer, we begin to see how… there could be non-arbitrary content-relations among causally related thoughts" (Fodor 1987: 19).

The fact that the human was seen from this strongly disembodied perspective determined the ideas for implementation of human-computer interaction. If the user had a proper mental model[74] of the functioning of the computer, the conversational interface – the command line – would be an ideal channel to interact with the computer. This, together with expensive yet primitive hardware and operating systems coloured the initial stage of *commercial computing* until Xerox Alto and Apple began to research graphical user interfaces based on research on Gestalt theory.[75]

4.3.2.2 Phenomenology

Winograd and Flores' book *Understanding Computers and Cognition* (1986) is often listed as the seminal text criticising the cognitivist/rationalist approach to computing. It has had strong influence on various succeeding work, such as Coyne (1995), Bannon and Bødker (1991) and Dourish (2001). In their study, Winograd and Flores stress the importance of embodiment, of the difference between subjective and objective knowing, and the material conditions that produce mental states and the cultural and historical context in which the act of cognition takes place. They are adamant about illustrating how technology is essentially a social practice, situated in the everyday activity of humans. In fact, the revolutionary aspect of their work is how their focus has more in common with biology, hermeneutics and phenomenology than traditional computer science as it was at the time. They claim that through designing tools, we are determining alternative ways of being in the world and that such design is essentially an act of ontological enquiry, a transformation of tradition:

---

[74] See criticism of this view in section 3.4.

[75] I stress the fact that this was the case in computing in general. It has to be noted that there were many interesting experiments that saw the potentiality of the computer as *virtually embodied*, as an augmentation of the mind. There were a few black horses in the field at the time and in the section on epistemic tools below we will get to know the ideas of Vannevar Bush (with his Memex system), Doug Engelbart (and the augmentation workshop), Ted Nelson (Xanadu) and Alan Kay (with his Dynabook), all introduced in section 4.4.2.

> Computers, like every technology, are a vehicle for the transformation of tradition. We cannot even be fully aware of the transformation that is taking place: as carriers of tradition we cannot be objective observers of it. Our continuing work toward revealing is at the same time a source of concealment… we can let our awareness of the potentials for transformation guide our actions in creating and applying technology. In ontological designing, we are doing more than asking what can be built. We are engaging in philosophical discourse about the self – about what we can do and what we can be. (Winograd & Flores 1986: 179)

Although Winograd and Flores' work on phenomenology has been influential, their reliance on speech act theory (such as that of Austin and Searle in the 1960s) as a solution for building better technology has been criticised. By categorising the task domain rigidly and then proposing (speech) acts as the way the user interacts with the system, they are in fact determining, limiting and imposing structures (social, aesthetic and political) upon the user of the software. Suchman is critical of this endeavour and precedes the arguments by Bowker and Star (2000) by pointing out that:

> [t]he picture of the basic conversation for action unifies and mathematizes the phenomena it represents. It works by transforming a set of colloquial expressions into a formal system of categorization that relies upon organization members' willingness to reformulate their actions in its (now technical) vocabulary. (Suchman 1994: 185)

For theorists like Susanne Bødker, Winograd and Flores engage in activities that are not only of social pertinence, but also political. Through system design one organises social realities and activities in offices, homes, and production studios around the world. McCarthy and Wright (2004) write that Winograd and Flores "not only pre-constitute categories for discussing organizational life, they also impose those categories as ontological reality on practice." (ibid: 33). We thus get a picture of how the field of HCI comes to an understanding of the cognitive implications tools can have through the political and social aspects of what Winograd & Flores call ontological design. From this realisation of the political nature of usability-design comes the tradition of *participatory design*, where the user is involved in the design of the system from very early stages. It also involves an understanding of the limited knowledge a system designer is able to have from a detached standpoint. Involvement with the actual situation is necessary in order to grasp the functionality that the system has to perform. This tradition is particularly strong in the Scandinavian countries and has proponents in Bødker (1990, 1996), Ehn (1987, 1988), Button and Sharrock (1996) and Kensing (2003).

4.3.2.3 Ethnomethodology

Although the work of Winograd and Flores has been successful in generating an understanding of embodiment and situatedness in the design of computer systems, the problems they encountered have been addressed recently with influences from anthropology and sociology. In anthropology, ethnography is described as the encounter between actors of different social or cultural contexts. The Heideggerian ideas of fore-knowledge and Gadamer's description of the hermeneutic circle [section 2.3.4] thus filter into the socio-academic practice. Here, the attempt to describe a culture from a neutral standpoint is acknowledged as an impossible task due to the hermeneutic or interpretative baggage we carry with us. Social practices are not translatable between cultures or sub-cultures and therefore the attempt to design software as universal engines for human activities is likely to be futile. The ethnographer thus observes and interprets the area of study with an awareness of her cultural pre-judgements. The explanatory framework will always inform both the observation and the interpretation, such that all questions posed and all answers derived will be culturally conditioned.

Another approach aware of the ethnographic circularity of interpretation, is *ethnomethodology*. Developed in the 1960s by Harold Garfinkel, the object of this methodology is to study people in their real settings and how social order is maintained. Latour has performed ethnomethodologically influenced studies in scientific laboratories (Latour 1987), but he parts ways with the main strand of ethnomethodology by creating the complex terminology of the actor-network theory. Ethnomethodology studies the field and tries to describe the social structures by using the very the language of the field in its commonsense parlour of concrete social circumstances. It focuses on the origin and usage of the conceptual categories of social practice. This "shifts the emphasis away from the production of sociological accounts and theories of social doings to an emphasis upon the description of the accountable practices involved in the production of naturally organized phenomena" (Button 2000: 325).

Researchers in the field of HCI have used ethnomethodology extensively for understanding the situation of end-users and it has spawned various other approaches. Lucy Suchman's work (1987) did partly emphasise the importance of studying the context in which the system would be used. The situated action theory (ibid.) emphasises the contextual knowledge of actors in their environment and the common sense or tacit knowledge they use to produce, analyse and collaborate with other actors in actual working situations. The participatory design movement (Bødker et al. 1987; Greenbaum & Kyng 1991) has taken these ideas further and emphasised the role that the users of the system should have in its design. Important research has focused on computer supported cooperative work (CSCW), aiming for a situation where the computer is not just designed as a single-user device, but one that represents accurately what typically happens when people work together in a shared space (Dourish 1992; Fitzpatrick 1998). Conversation analysis (Sachs 1992) is yet another popular method to get an understanding of users' needs, but more importantly it is an active involvement in the workspace and the designer should ideally become a part of the work setting in order to understand the flow of activities that take place.

4.3.2.4 The Experiential Turn

In recent years the importance of emotions in technological design has been emphasised by various authors such as Norman (2004), Jordan (2000), Sharp et al. (2002) and McCarthy and Wright (2004). McCarthy and Wright propose a turn from analysis to practice. They wish to create a space for the individual and felt experience; of technology as something that is lived through. Taking a pragmatist approach inspired mainly by John Dewey and Michael Bakhtin, they outline four important threads of experience: *The Sensual Thread*: the way we experience things through our sense organs; the concrete, visceral character of the situation in which we find ourselves. *The Emotional Thread*: the emotional quality of experience as understanding or sense-making process. *The Compositional Thread*: a focus on the whole of experience and its relationship to the individual parts that make up the experience. *The Spatio-Temporal Thread:* an analysis of an emotional experience that can change our sense of space and time. Our language about experience is pervaded by metaphors of space and time. McCarthy and Wright question why HCI has largely ignored the social theory that is concerned with the production of the subject or the person.[76] They ask if it could be, that “social theoretic insertions and the rejection of information processing notwithstanding, HCI and CSCW (Computer Supported Cooperative Work) still hankers after a rationalism that would be challenged by a turn to the person and then unavoidably to desire, value and emotion?" (ibid: 190). Insisting that it is possible to bring the language of identity and difference to the practice of sociocultural analysis of technological use, they propose that to shift:

> the focus of conceptualizing human-computer interaction from the representation of observable technologies of accountability, contextualized cognition, and external consciousness to the emotional-volitional play of observable and unobservable, cognition and emotion, internal and external puts the weight of each particular moment at the center and brings into focus the variety of experiences that people engage in. Shifting the focus from one pre-constituted structure or another – mind or society – to the interplay that constitutes both enables us to see participation as well as non-participation, multiple identificatory possibilities, lags in participation, and conflictual moments of identification. (ibid: 190)

McCarthy and Wright use pragmatic theory to emphasise the situated creativity of action, the openness of experience and sensory engagement of real end-user situations. Every experience is also aesthetical and ethical, of which we have answerability. This experience is intersubjective and forces us to relate to other people such as the designers of the tools or other users, whilst the

[76] McCarthy and Wright continue that tradition within HCI by choosing to ignore the work of various scholars that have done precisely what they are proposing, i.e., to look at how society and technology construct the subject through various impositions (such as Foucault, Butler, Turkle). However, they are right in that these philosophical or sociological theories have not made it properly into HCI or computer studies as an important topic of research.

holistic nature of the experience makes us acknowledge the relational and multi-layered quality of life. This emotional-volitional character of experience requires us to avoid the reduction of it into conceptual categories that downgrades the lived experience in order to gain clear formal structure. McCarthy and Wright point out that we cannot design experience; we can design *for* an experience. Experience is as much about what the user brings to the situation as what is provided in the situation as material for experience. They then offer six processes to take into account when we design or analyse technological artefacts: *Anticipation:* we anticipate our future actions, but this planning is always originating in the flow of experience. *Connecting:* refers to the pre-linguistic and immediate character of the concrete situation. A judgement is made without much conscious thought. *Interpreting:* the narrative structure of the experience. It involves the relevant agents and possible actions to take. Its scope is the remembrance of the past and the projection into the future through the awareness of the experiential momentum. *Reflecting:* we constantly make judgements as the experience unfolds. *Appropriating:* we make the experience our own by reflecting how it relates to our historicity, sense of self and projection into the future. *Recounting:* the act of making an account of the experience for ourselves or to other people through speech or writing (ibid: 124).

The focus on designing experience has been an important factor in the design and analysis of the ixiQuarks. It is a perspective that is much more inclusive and holistic than the narrow and abstracted approaches in early HCI. As we will see in sections 5.2 and 6.7, it is an important part of our surveys and user interviews. It provides a conceptual framework to analyse the user experience in relation to other questions such as the understanding of the designer, the reading of the technology as script, the interpretation and acceptance or rejection of its programme, etc. In the current context the experiential design theory is useful in user studies as a framework for analysing the emotional and embodied relationship people have with their tools, but the theory itself is lacking in coverage of embodiment, cognitive functions, and the agency of technology.

4.3.2.5 Technomethodology

In their first paper on "technomethodology" Button and Dourish (1996) state that the main contribution of ethnomethodology has been to show how systems designed for a specific work often fails to support it, or even worse, they do not allow people to go about their work in the way that they have done hitherto or would prefer to do. Ethnomethodological studies of the workplace have been helpful in analysing the workplace situation, but they also pose particular problems. The paradox of systems design is that "the introduction of technology designed to support 'large-scale' activities while fundamentally transforming the 'small-scale' detail of action can systematically undermine exactly the detailed features of working practice through which the 'large-scale' activity is, in fact, accomplished" (Button & Dourish 1996). The paradox (and here it is the paradox of technomethodology) is that if the workplace setting and the wishes of the people are taken into account and implemented in the system, the result is a system that is always *bound to transform* the nature of the work through the heightened technological infrastructure. This results in new desires and new feature requests, but these typically appear long after the system designers have left the site.

Technomethodology thus focuses on how to incorporate the findings and insights from ethnomethodology into the process of design. It does so neither by merely consulting the ethnomethodologist for how to design the system, nor does it simply use the ethnomethodological accounts for the design. It goes further and implements the system itself from the perspective of ethnomethodology, thus generating a "hybrid" (as Garfinkel has often termed it when ethnomethodology is adapted by other disciplines) between the sociological method and technological production:

> On the one hand, it deals directly with the generally operative social processes which are the currency of ethnomethodology; elements such as situatedness, practical action and representation, achievement and mechanism, phenomena of order, and accountability. On the other hand, it deals with the fundamental, almost implicit, aspects of system design-generalisation and abstraction, configuration, data and process, fixedness and mutability. (ibid: 4)

Technomethodology thus forces ethnomethodology to go beyond the emphasis on the particular, the situational, the unique, in order to create the general, universal and applicable concrete knowledge needed in system design. It is aware of the different abstraction modes in sociology and software engineering: the sociologist tends to have *analytic* abstractions (categories and descriptions of actions), whereas the software engineer will have more *generative* abstractions (not only characterising the system's behaviour but also giving rise to it). Technomethodology combines the "accountability" (description) of ethnomethodology with the "abstraction" (prescription) of system design. All computer systems are effectively built on abstractions and which become performative or executable through the process of running them. It is a method that does not simply analyse practice but also is aware that it "invents its future."

With their definition of technomethodology, Dourish and Button attempt more than merely pointing at the paradoxes of system design and work-place studies. After showing how human action as "accountable" by the ethnomethodologist they claim that what computational abstractions (as in software) "do *not* share with the abstractions of everyday activities is the observable-reportable nature of everyday action which is at the heart of ethnomethodological investigation" (Dourish & Button 1998: 16).[77] They propose a method of system design which

[77] The claim that everyday action is "observable-reportable" is rather problematic. Any discussion with an instrumentalist, a studio producer, a listener of music, or a dancer shows that the embodied act of making or listening to music is neither objectively observable nor fully reportable. Tacit knowledge is not easily conceptualised. As we saw in the section on hermeneutics, any attempt to represent the world will always

enables the opening of the abstractions, the exploration of the black box: "like social interactions in everyday life, we would like our interactions with computational systems to be organised in terms of abstractions that are supported by their own unfolding, rather than opaque (and brittle) black boxes" (ibid: 16).

The *Open Implementation Model* of Kiczales (1996) is seen by Dourish and Button as a possible solution. In this model, the system offers an account of its own activity. This is a well-meaning and appropriate approach, but with an abundance of practical problems, encapsulated in the basic question: what shall be shown to the user and how shall it be represented? One approach is to conceptually or graphically represent the workings of the program, but this would lead to yet another level of abstraction that would have to be learned and decoded by the user. Another approach would be to simply reveal the source code of the application. This is the approach taken in Open Source software such as SuperCollider and ixiQuarks (where the user can always call up the source),[78] but it requires a strong knowledge of programming if the user is to understand the workings of the program. Often a person with that strong knowledge of programming hardly needs to see the code. She would understand intuitively what software engineering processes would lie beneath.

Although the approach taken by Dourish and Button in their technomethodology is useful in system design, their belief in the "accountability of systems" i.e., that the system can be accountable on its activities just like a human is based on a premature understanding of the semiotic nature of language and symbol systems. A system account of its inner workings might only become yet another layer of complexity which the user has to understand. As useful as such system accounts can be in some cases (such as the case provided by Dourish and Button of providing a more detailed representational structure of the process of copying files between remote computers), my stance is that some processes are better left blackboxed. Instead, we should try to understand and acknowledge the difference between the physical world and the world of epistemic tools and embrace those differences.

4.3.2.6 Semiotic Engineering

Yet another field relevant to the design of human-machine interfaces is semiotics, the study of signs. If software designers intend to convey a certain meaning with the software, a work pattern or a specific view on how to deal with a problem, they have to consider how this is best presented on the interface level and what kind of language (graphical, textual, or mixture of

---

happen from a pre-judgment, from a specific world-view that is different to the one that observes or interprets.

[78] Of course we are faced with the problem of the strata of abstractions when defining source code. The source code available in ixiQuarks and SuperCollider is genuine SuperCollider code. If a lower level is desired, the user would have to enter the level of C/C++/Objective C code layer in which SuperCollider is written.

both) is appropriate. Various authors have explored the relevance of semiotics in the design of digital systems (see Nadin 1988; Andersen 1992; Brandt 1993; Nake 1994; Iazzetta 1996; Andersen 2001a; Andersen & May 2001b; De Souza 2005) where the common understanding is that the act of designing a user interface is an act of communication and therefore essentially semiotic. Such design comes about in culture and often acts as a bridge between scientific and humanistic praxes (Nadin 1988: 269).

Modern semiotics originate primarily from Saussurian linguistics (1983) and Peircian semiotics (1955). From Saussure's theory of signs they derive the arbitrary relation between the signifier and the signified (the reference and the referant). From Peirce's theory of signs come the definitions of icons (direct mapping between sign and its object, such as in a diagram or a picture), indexes (dynamical relations between object and interpretant, not based on convention), and symbols (a relationship between object and sign based on pure convention, thus strongly cultural). Furthermore, from Jakobson they derive a communication model which is a six-fold structure of *sender – message – receiver* where *context*, *channel* and *code* are important factors (Jakobson 1960: 350). This model is used in the analysis of interfaces as semiotic machines in section 6.3.

Andersen's project is to establish a discipline of computer semiotics analysing the origin and application of meaning in computer systems. Andersen (1992) outlines four modes of seeing signs as used in computer systems. The sign as *system* points to the social structure of signs and the difficulty of changing social conventions. The sign as *knowledge* shows how an individual can acquire information from and use sign systems. The sign as *behaviour* points to the importance of the sign in social communication and collaborative work. And finally the sign as *art(ifacts)* stands for the generative creativity of sign use, how users are able to present ever new meanings through the use of signs. Andersen (2001b) further points to the important role semiotics can have in making HCI more coherent and situating them in the context of other media. It becomes necessary to understand how design on computers relate to and remediate symbolic systems from other media (see Bolter & Grusin 1999) and study properly the unique characteristics of the computer as a communication medium.

In her project of semiotic engineering, de Souza uses semiotics to show the principles, materials, processes, effects and the possibility of creating a meaningful discourse on computer systems. As such, the project investigates the epistemological conditions and methodological commitments that affect the creation of a computer based artefact. Signs become the main focus of designers according to this method. De Souza uses concepts derived from semiotics such as metacommunication, semiosis, signification and communication in her work with HCI analysis. These concepts prove useful when generating HCI research questions and methods.

The concept of metacommunication derived from semiotics is used by de Souza to illustrate how the user of a system engages with its designer on a communicative level. Here the system becomes the "designers deputy," i.e., a communicative agent that represents the designer at interaction time. This entails a shift in perspective: the developer is injected into the HCI picture, such that the system now includes the developer, the communicative system itself, and the user. The focus is not merely on usage of the system but on its communicative meaning. Another shift is that whereby seeing the system as the locus of problem-solving is transformed to seeing the system as a communication between the designer and the user. Furthermore, the designers and the users belong to the same ontological category, as interlocutors in computer-mediated communication.

De Souza explores how signs can stand for something else. This is established through the process of semiosis, using the above mentioned theories by Saussure and Peirce attempt to explain how this happens. Meaning is here seen as a dynamic entity, one that can change according to context. The establishing of meaning is therefore a process of signification where the some content is assigned to certain expressions. Signification is a highly culturally conditioned activity which further emphasises the importance of the discipline of ethnocomputing [see section 4.3.1]. Communication in the context of semiotic engineering is defined as: "the process through which, for a variety of purposes, sign producers (i.e., signification system users in this specific role) choose to express intended meanings by exploring the possibilities of existing signification systems or, occasionally, by resorting to non-systematized signs, which they invent or use in unpredictable ways" (de Souza 2005: 26). It is important to note here, in the context of communication systems, that although a symbol can be applied with different signification in the different computer systems, its function in a system has a rigid meaning. Computer systems are not flexible like human culture in terms of changing or adapting its meaning of a symbol. Their meaning is codified in their design and this is a factor that users have to take into account. Computer mediation therefore imposes strong limitations in terms of semiosis.[79]

The shortcomings of this methodology can be summarised with reference to the circle semiotic engineering draws around designers and users as the main actors that communicate in an interactive system. They close the circle and pointilise the users and the designers as authorities of actions. This focussed circle is too narrow for the current context; a more fruitful approach is to operate with the actor-network theoretical concept of a complex web that reaches out to all kinds of networks of technological production, social and technical protocols, conventions and cultures of technology usage. Semiotic engineering can thus be helpful in the design and analysis of technology, particularly within the defined limits of semiotic design in

[79] This influences de Souza to make a distinction between human meanings (of designers and users) and computer meanings (of coded programs). For the computer, the meaning of a "symbol" is just the encapsulation of a state, an object or class of objects that are encoded into another symbol system that translates to the physical state of the hardware. Computers do not allow for unlimited semiosis or floating signifiers.

computational media, but like many of the theories surveyed in this section, it can not serve as a methodology to be used exclusively in this context of epistemic tools.

4.4. The Epistemic Tool

The mind is inherently embodied. Thought is mostly unconscious. Abstract concepts are largely metaphorical. These are the three major findings of cognitive science. (Lakoff & Johnson 1999: 32)

The implications of the digital computer to the design of musical instruments is a key focus of this thesis. The design of digital tools is a fundamentally new activity in human history and it differs immensely from the design of analogue tools. For example with regards to how we learn acoustic instruments through an enactive and embodied practice. Here digital instruments pose a difficulty. They do, of course, provide mapped couplings between bodily gesture and resulting sound, and what we might call *virtual embodiment* [section 7.2], but their nature is essentially symbolic. Although we interact in an embodied manner with the computer (moving our mouse in a 2D space, touching screens or swinging Wiimotes) the interaction always takes place through symbolic channels of varied bandwidths. The interaction is primarily with words, menus, icons, graphical representations and not physical mechanisms. This symbolic communication is based on an abstraction which is inherently an approximation, a representation, a construction of the task/the world/the music. It is therefore in the hands of the software designer to decide what the tool affords in music-theory terms. The nature of constraints in acoustic instruments are of a different kind: they have naturally inbuilt affordances, but those are mechanistic and physical, not musico-theoretical (or symbolic).

From the above, and with a slight exaggeration in order to prove a point, we can assert that the *body* of the acoustic instrument is physical, whereas the primary body of the digital musical instrument is intrinsically theoretical.[80] It is this symbolic foundation of digital music systems that calls for the conceptualisation and definition of the epistemic tool. Whenever we map or interpret data with some technologies, i.e., informatics, we are using tools that should be considered as having an epistemic dimension. Epistemic tools are the result of some particular designers' categorisation of the world, selected representation and system building. Such a design is a process of selection and omission, abstracted as general system. This necessarily implies the cultural foundation of the epistemic tool and its political, ethical and aesthetic pertinence.

In all crafts, the awareness of the relationship between tool use and aesthetics has long been acknowledged (Pye 1968; McCullough 1996). The practised hand uses highly designed

[80] If the digital musical instrument makes use of tangible user interfaces, those are normally arbitrary peripherals that could be swapped out or changed at any time.

and sophisticated tools that result in a unique aesthetic, historically rooted within a specific tradition of workmanship. The same applies to digital tools, although their material-symbolic nature might be harder to scrutinise than their physical counterparts. All artworks originate from a specific technique, and this very technique is a result of a specific context where aesthetics, materials, ergonomics and cultural situation merge. Real life is dynamic, characterised of fuzzy borders, tacit knowledge, distributed roles and networks. It is flexible, adaptive, and emergent. If a node in the network disappears the flow is quickly adjusted to that lack.

Computer systems are of a different nature. They are abstractions, interfaces or black boxes to designed activities. For the system designer it is therefore important to work with the users of the software (as we find in participatory design) and analyse the context in which it will be used (the focus of ethnomethodology). They should also be aware of the fact that the system itself will change the practice with its introduction in the work flow (as analysed by technomethodology). It is precisely this abstract nature of system design that constitutes the epistemic nature of digital tools. The tool is designed, its affordances and constraints are outlined, and the user's actions are predicted and delineated into the interface and interaction design of the tool. The designer decides with clear rational arguments what is revealed and what is concealed when the system is used. When computers are seen as mediation, the role of the system designer is to direct the channel of energy form the physical interface to the work (or the *terminus*) through the symbolic engine of the epistemic tool. It is this activity of blackboxing, of creating abstractions of activities where bodily movements *and* thoughts are presented as discrete chunks in time that makes these tools highly complex and non-transparent.

For Latour, there is a symmetrical equality between the human and the nonhuman (such as the technological object). By applying a new technology, the user is subject to a process called translation, where his goals and the goals of the technology are transformed through displacement, drift, invention, mediation – the creation of a link that did not exist before. "Which of them, then, the gun or the citizen, is the *actor* in this situation? *Someone else* (a citizen-gun, a gun-citizen)… You are a different person with a gun in your hand" (Latour 1993: 179). Latour introduces the idea of *symmetry* where the actors both change their being by coming into relationship with each other: "This translation is wholly symmetrical. You are different with a gun in your hand; the gun is different with you holding it. You are another subject because you hold the gun; the gun is another object because it has entered into a relationship with you" (ibid.). This is the situation musicians finds themselves in. The symmetry is at work: the tool is different in the hand of the musician, but he is not the same musician anymore. Although this applies to acoustic instruments as well[81] it is of a different nature with

[81] The choice between a Gibson jazz guitar or Ibanez heavy metal guitar will strongly affect the guitarist on many levels. So does the composer's choice of instrument to write for.

digital instruments as there is more *music* inscribed in the body of the software than in the acoustic instrument. Consider the following quote:

> Software design is the act of *determining the user's experience* with a piece of software. It has nothing to do with how the code works inside, or how big or small the code is. The designer's task is to specify *completely and unambiguously* the user's whole experience. (Liddle 1996: 30, my italics)

Here it is stated that the systems designer is designing *completely* and *unambiguously* what the user will experience when working with the software. Whilst I hope this is not absolutely true (the goal in ixiQuarks is to aim at expressive openings within the constrained environment of each instrument) Liddle's point is that the user's experience is designed from the self-conscious, top-down, and encapsulated overview of the system designer. The designer of the acoustic instrument is not designing the musical experience per se, but the tactile, ergonomic, and inspirational (music catalyst) experience. In order to understand this point better we have to delve into the nature of epistemic tools.

4.4.1 Instrumental Knowledge – Material Epistemologies

In his book *Thing Knowledge*, Davis Baird demonstrates how material objects have a different epistemological status to statements of language. Technological objects can have a function which the user is able understand, but could not describe in a language as "[t]he material products are constitutive of scientific knowledge in a manner different from theory, and not simply 'instrumental to' theory" (Baird 2004: 1). In a manner akin to Heidegger, Baird shows that technology can precede science, and afford scientific discoveries through their physical structure and functionality. The point here is not what precedes what, but rather that the instrument becomes an *expression* in itself, an externalisation of knowledge in a form that is not symbolic but material.[82]

> Knowledge can be expressed in many ways. Theories express knowledge through the descriptive and argumentative functions of language. Instruments express knowledge both through the representational possibilities that materials offer and through the instrumental functions they deploy. (Baird 2004: 131)

Baird is aware of Latour's account of blackboxing. When the black box works, its origins are forgotten and "paradoxically, the more science and technology succeed, the more opaque and obscure they become" (Latour 1999: 304). Baird, agrees with Latour on the nature of scientific blackboxing, but sees another and perhaps more active function of the black box itself. While talking about a particular instrument, called Spectromet, Baird says "[t]he knowledge used in this context is tacit in the sense that those using the instrument (typically) could not articulate

[82] Sections 7.4 and 7.5 explore how many programmers see their software as artistic expression, thus blurring the line between the system designer and the user of the software.

the understanding of spectrochemestry they deploy in doing so. Nonetheless, they can use it. This spectrochemical knowledge has become detatched. It has gone inside – inside the instrument – and can now tacitly serve other technical and scientific purposes" (ibid: 163). It is clear how the blackboxed instrument now contains the knowledge of its inventors which means that the users of the instrument do not need to have a deep knowledge about its internal functions. If we assume that both the designers of the instrument and its users have an understanding of it, this understanding is very different and attained from distinct origins. The former creates the instrument from the conceptual understanding, the initial idea, whereas the latter gains operational knowledge that emerges through use (or habituation) and not from an abstract understanding of the internal functionality.

In his book, Baird provides an analysis of three different modes how technological artefacts perform epistemological work, but he states that this list is not exhaustive. Instruments can embody knowledge as material mode of representation (model knowledge); as material modes of effective action (working knowledge); and as material mode of effective action that synthesises representation and action (encapsulated knowledge). These are three forms of material epistemologies that we will now explore.

4.4.1.1 Model Knowledge, Working Knowledge and Encapsulated Knowledge

For Baird, technological artefacts are *models* of thought and as such they can perform epistemological work alongside scientific theories, often preceding them. An example is the development of the water wheel, where the practical building of a model was more successful than theoretical calculations. Another example is of James Watson and Francis Crick discovering the double helix (the DNA) through manipulation of physical models of atoms. It was the manhandling of the physical model, experimenting with structural combinations of the model, that led to the discovery.[83]

> The ability to manipulate material models by hand, so to speak, is important because it provides a different entry point for our cognitive apparatus. Conceptual manipulation provides one entry, material manipulation a second *independent* entry. By admitting that models, and instruments more generally, have epistemic status, that they are knowledge, we enlarge our ability to bring our cognitive apparatus to bear on the world. (ibid: 40)

Model knowledge is representational. It is a symbolic structure that embodies a material (or social) theory. The model can function alongside the theoretical contributions to science and

---

[83] This fact is realised in contemporary design and perhaps epitomised in the *Sketching in Hardware* conferences that are currently in their 3rd year. Here the doctrine is that hands-on knowledge and manipulation of physical stuff and hardware yields design results that are different from design done on paper or, particularly on computers. see: http://www.sketching07.com/ and http://www.sketching08.com/

technology. Its basis is empirical and theories can be proved or refuted by empirical evidence. As material objects they lend themselves to physical manipulations of the models, often generating ideas that would not have been conceived of through non-external thinking alone (the model here being, of course, the ideal candidate for Clark's tools for thought).

Working knowledge is a form of material knowledge where the instrument embodies a "know-how" (although Baird states that it is not his intention to anthromorphise the object). The instrument bears material agency, a way to perform correctly according to a set of definitions. As such, it relates to the idea of tacit knowledge, as non-verbal "know-how" but that relation only extends to a certain degree:

> Subjectively, a person who has "a working knowledge" has knowledge sufficient to do something. Objectively, a device that bears working knowledge works regularly. It presents a phenomenon, which might be used to accomplish something. This form of material knowledge, in contrast to model knowledge, is not representational, but rather appeals to pragmatist notions of knowledge as effective action. (ibid: 45)

Moreover, a working knowledge in physical things does not have to have deep theoretical basis as its fundament is material:

> With the air pump, we are admitted into the laboratory of nature herself. While we are not guided by human theorizing, it does take technological ingenuity to unlock this laboratory. We know material agency – nature herself – not through our words but through our crafts. We make instruments where material agency is both in our control and engaged, "working." We *make* this working knowledge of material agency. (ibid: 48)

Finally, there are certain instruments that *encapsulate knowledge* by functioning as a hybrid of a model and a working knowledge. Measuring instruments are a good example.

> At a fundamental level, measurement requires a phenomenon – the working knowledge.... Here is the signal generated by interacting with the specimen. But it is a field of possibilities, typically – but not always – understood theoretically, that drives the choice of signal generated and the transformations that are made to the signal as it is rendered "a measurement." These choices, then, encapsulate in the material form of the instrument a representation of this field of possibilities – model knowledge. When the integration of both forms of material knowledge is done seamlessly, the instrument appears to extract information from nature. (ibid: 68)

In a Heideggerian manner Baird shows that instruments are not simply "instantiation of ideas." Materials and ideas are both necessary in science and materials do not behave like ideas. This is the point of Baird's instrumental epistemology. "[A]n artefact bears knowledge when it successfully accomplishes a function" (ibid: 122). Theoretical scientists are "concept smiths" whereas people who build instruments are "function smiths" (ibid: 123). They develop, replace, expand, and connect new instrumental functions from given functions. The functions of a measuring instrument are semiotic, they are built according to our theoretical understanding of the domain to be measured.[84] As such, measuring instruments do not extract information from the world: they generate a signal that has to be transformed into a suitable form and only then can the data be understood as information about the world. We thus witness one of the functions of the epistemic tool as defining, outlining and conditioning the world into a form that can be incorporated into our general practices.

4.4.1.2 Instrumental Knowledge in Musical Instruments

Looking at Baird's material epistemology, it is obvious how the inventors of acoustic instruments have benefited from the model and working knowledge of objects. The string "knows" its pitch and timbre according to its length and tension, and the wood "knows" how it resonates. The same happens with electronic instruments. Pinch and Trocco (2002) show how the Moog synthesizer got its special sonic characteristics without a solid theoretical understanding of why it sounded as it did. One example was how the oscillators would not lock together in terms of pitch, resulting in a pleasant choral effect. Another was that the circuits were deliberately overdriven, producing modulations and distortion of the sound. A Moog engineer, Jim Scott, explains

> It was something like vacuum tubes, in that the circuitry would not suddenly go into clipping, it would distort gracefully… Also, the circuitry was inherently wide band…It passed frequencies beyond the audio range… And we're getting into guesswork here, but the feeling is that there were things that happened up in the ultrasonic range that can cause inner modulation and distortions, [this] reflects back and can be heard in the audible range. (ibid: 235)

The heart of the matter is that although the electronic technology is different from the physicality of acoustic instruments, it is still a system with strong physical properties of valves, oscillators, wires, transistors and conductors. The *electronic instrument* is a physical system like the acoustic instrument, but with an interface that is designed and decoupled (the keyboard instruments are an exception in this case in the acoustic instruments), such that any gesture could be mapped to any parameter.

From our analysis of the nature of system design in section 4.3.1, we can now crudely generalise a core difference between the digital and the acoustic: In the *acoustic instrument,* the sound-generation theory is given to us for free by nature. We had complex instruments in the form of lutes, flutes, and organs long before we ever had a solid theoretical understanding of

---

[84] The relationship we have with such instruments would be hermeneutic in Don Ihde's terminology. The difference between Baird's and Ihde's projects can be defined as Baird focusing on the epistemology of material objects themselves versus Ihde's phenomenology of our embodied relationship with the world *through* material objects.

sound. The physics of wood, strings and vibrating membranes were there to be *explored* or *discovered* and not to be invented. However, *using* the acoustic instrument we need to have understanding of music, music theory, and the tradition in which we are playing. As the interaction with the instrument is primarily embodied, we have to train our body to be in rhythm with the instrument, i.e., incorporate the musical knowledge *in sync* (or mutual resonance) with the particular instrument of choice. As opposed to the generic explicitness of the digital instrument, the acoustic instrument contains an infinite scope for exploration as its material character contains various ways for instrumental entropy, or "chaotic," non-linear behavior that cannot be mapped and does not have to be the same on the same type (brand and model) of instruments.

The *digital instrument*, however, is first and foremost theoretical. In order to *make* the instrument, we need to know precisely the programming language, DSP theory, synthesis theory, generative algorithms, music theory and hopefully utilise knowledge of HCI. To *use* the instrument, on the other hand, we learn the tool through working with it and habituate ourselves with its functions, gradually building an understanding of its functionality. As users we do not need to know as much about synthesis or music theory, as the black box is intended to be used through its simple and user-friendly interface and the music is often programmed into its functional logic.[85] Ergonomically the interaction happens primarily through a symbolic channel, which gradually teaches the user to operate with technical terms (such as "low-pass or the highly rational "beats-per-minute"), but this happens from the habituation of the model (what I call the epistemic tool). The predefined quality of the digital instruments means that its functionality can be exhaustively described in a user manual: all is explicit and nothing is supposed to be hidden.

---

[85] Baird quotes Richard Feynman who said "What I cannot build I don't understand." (Baird 2004: 114). Interestingly Feynman was quoted with the same quote by a participant in a survey we performed with users of audio programming environments (who resist the blackboxing of commercial musical software). The idea was that in order to understand one's own music one has to build the instrument or the compositional logic oneself.

4.4.2 Epistemic Tools as Symbolic Extensions of the Mind

Metaphorically [computers] let us get a hold of our ideas. Concepts become things....Our use of computers ought not to be so much for automating tasks as for abstracting craft. (McCullough 1996: 81)

4.4.2.1 Four Pioneers of Epistemic Tools

Let us commence this section with a half a century old quote by Licklider:

> Man-computer symbiosis is a subclass of man-machine systems. There are many man-machine systems. At present, however, there are no man-computer symbioses... The hope is that, in not too many years, human brains and computing machines will be coupled together very tightly, and that the resulting partnership will think as no human brain has ever thought and process data in a way not approached by the information-handling machines we know today. (Licklider 1960)

Does this sound like a science fiction? Although it is unclear what Licklider meant by "not too many years," it can be argued that we have entered this state of symbiosis long time ago. In Andy Clark's terms, we "offload" our cognition and communication onto our computers that perform some of our cognitive or communicative tasks like searching for a file, finding information online that we might already know, write music, or send email to a colleague at the next table. Licklider discovered that most of his time "thinking" as a scientist was spent in "getting into position to think" (ibid.) where calculations had to be done, graphs made and data compared. Then, finally it was possible to "think" about the topic. Licklider here uses the verb "think" in a different sense from Clark as he excludes the activities of making graphs or calculating as thinking. For Licklider, thinking is a more abstract process that does not have to deal with the material scaffolding of the thought. Once again we meet the disembodied cognitivist view of the mind. Licklider does not acknowledge that the graphs themselves, the process of making them and the way they appear – what software was chosen to make them? – can have strong effects upon the thinking process. A specific software might take the scientist upon a path of thinking that another software could not. However, what Licklider is showing is that much of our cognitive activities (although not "pure thinking") can be automated and "offshored" onto a machine.

System designers have focussed on this idea of cognitive prosthesis of the human, using computer technologies, from the beginning of computational technologies. One could go back to Charles Babbage and Ada Lovelace, to find examples of how they define their machines as being of (or assisting) cognitive nature. Rather more recently, one influential pioneer was Vannevar Bush, who wrote a paper in 1945 about a system he called *Memex*. It was a hypertextual system that would combine all scientific knowledge. Bush proposed a system of microfilms where text could refer to other texts through a mechanism that we now know as hyperlinks. The Memex was "a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory" (Bush 1945).

Roughly more than a decade later Douglas Engelbart (the inventor of the computer mouse) was working on electronic computers and he came up with a similar idea, although now it was called "augmentation workshop." Engelbart was conscious of how every tool affords certain thinking and how computers would yield certain mental structures in their users. A good example is how the Arabic numbers as a new medium and tool for mathematical thinking simplified certain operations (especially mental calculation) compared to the older, and now less used, Roman numeric system.

> We did note that the co-evolution of the human and tool systems is the driving force of the kind of intellectual bootstrappings we set out to do. When people adopt powerful new tools, a subsequent co-evolution of the human systems is inevitable. Very seldom do you come in and just automate something and find that your change has had no other effect. The goal of augmenting rather than automating is being able to stimulate and design this co-evolution. (Engelbart quoted in Rheingold 2000: 327)

This quote reads much like technomethodology's acknowledgement of how the system designed for a specific work practice is always doomed to change the work itself through its introduction. Engelbart's main focus was to study how the machine could augment and enforce our external symbol manipulation habits, to automate them, and design a machine so clever that it might come up with combinations that we might not even be able to think of.[86]

Two other pioneers are important to mention here: Ted Nelson and Alan Kay. In 1960, Nelson designed a system called Xanadu which, if implemented, would work like an improved version of the World Wide Web. Xanadu is a multimedia platform that would improve human thought through transparent links, support of a variety of media types and seamless integration between documents and media. Kay's 1970s Dynabook is also a hugely important vision of a world where humans can engage with digital devices as they do with the notebooks they carry around with them. As opposed to the notebook, the Dynabook would be programmable, able to present and work with all media types and become a second platform for human cognition. Similarly to Licklider, Kay does not see thinking as something that happens outside the head, but "[a]lthough thinking goes on in one's head, external media serve to materialize thoughts and, through feedback, to augment the actual paths the thinking follows" (Kay 1977: 31). This is important for Kay, but further to the genuine novelty of this medium: "Methods discovered in one medium provide metaphors which contribute new ways to think about notions in other media" (ibid.). With modern laptops, integrated media browsers and compositional

[86] For a good survey on Engelbart's engineering implementation and ideas of human augmentation as compared to Stiegler's thesis of epiphylogenesis, see Barnet (2006).

environments, we have reached a stage where all these things are possible. But just like Engelbart and Nelson, Kay does not think that we have done a good job in designing the workings of our media. Some of the reasons for this discontent will be explored in the next chapters on the inventions, innovations and nature of digital musical instruments.

It has been important is to demonstrate that the idea of epistemic tools, of artefacts that exist external to our body but assist with our thinking, is an old idea that has now epitomised in contemporary music technologies, albeit not without problems. However, the pioneers mentioned above were not particularly concerned with the cultural problems, perhaps as, all being part of the same (and arguably narrow, technical sector of already highly segregated) North-American culture, they are not able to perceive how their systems impose cultural conventions on onto other cultures. The have therefore neglected the investigation into the ethical, social, aesthetic and political significations, particularly in the context of other cultures in the world, of their system design.

4.4.2.2 Process Enclosing Software in Process Engendering Machines

Having defined the computer as an epistemic tool, as an augmentation of the mind, and a prosthetic extension of our body, a question presents itself: what does this mean for designers and users of computer systems? The answer is long and the implications are wide, but for start it demands from the designers an understanding of fundamental role they play in the user's work practice. Their ideas of music, musical culture, and musical working practices are inevitably inscribed into the software. It demands an understanding from the users, that the software they choose to work with conditions their thoughts, defines their musical ideas, and work practices. It means that they have to be critical to the ways of the software, scrutinise the underlying assumptions the software signifies and thus consciously accept or reject the script that the software contains:

> Software is not just a device with which the user interacts; it is also the generator of a space in which the user lives. Software design is like architecture: When an architect designs a home or an office building, a structure is being specified. More significantly, though, the patterns of life for its inhabitants are being shaped. People are thought of as *inhabitants* rather than *users* of buildings. (Winograd 1996: xvii)

If the primary tools, such as hammers, extend the body and prescribe certain processes through their affordances, industrial machines automate those processes with energy that is not derived from the human body. The machines are repetitive which results in normalised output. The machines perform human tasks in an ideal way. Cybernetic machines (or computers) do not only automate human work processes; they prescribe human thinking through their symbolic structures to an elevated degree. The human response to the analogue machine was to emphasise human eccentricity, uniqueness and specialist craft. This observation reverberates the common thought that the invention of photography was simultaneously the invention of impressionist painting (Levinson 1997). The cybernetic machine streamlines and automatises human creativity as it becomes a tool for creativity. But since it is a tool for thinking, an implementor of our ideas, we can program our resistance to that streamlining using the very same machine. The resistance to the machine is done in the machine. And thus we get programming languages, creative environments, biologically influenced intelligence and support for human input into the machine all in order to expand its limitations. Mitcham describes what has been called machinic enclosure on these pages, using examples from engineering:

> The steam engine and internal combustion engine are no longer simply objects after the manner of hammers or saws; they have become containers for processes – and in a much more specific way than a cooking pot or dye vat, in which quite different processes can take place depending on what is placed in them and on external conditions (temperature, etc.). Gasoline placed in an internal combustion engine does something quite specific, and it does this only under conditions established by the engine. (Mitchell 2004: 168)

Mitchell refers to them respectively as *process engendering* and *process enclosing* machines. The hammer engenders action; affords creative use of it, whereas the machine is predefined, enclosed and generic. Whilst the computer itself (the perfect meta-machine) is an engendering machine that can be infinitively programmed, its software is not. Computer software is always an enclosing machine. It provides a clear and outlined expressive constraints that acoustic instruments (here seen as process engendering tools) do not have. This is represented in the following table:

instrument type	physical nature	expressive character
acoustic	defined constraints	openings
digital	open and mappable	closure

Table 4:1 shows a rough characterisation of the acoustic vs. the digitial in terms of nature and expression. The physical nature of the acoustic instrument (such as wood) is more clearly defined than the open and mappable nature of digital instruments (hardware).

As outlined above, Baird states in his materialist epistemology that things "bear" knowledge. The theories we come up with "serve instrumentally in the articulation and justification of knowledge borne by things" (Baird 2002: 13). This is a reference to how things can contain theoretical knowledge before they are conceptualised or understood as such. Baird's definition of such incorporated knowledge is that "an artefact bears knowledge when it successfully accomplishes a function" (ibid: 15). He provides an example of how Smeaton's 1751 physical waterwheel provided a better way to determine efficiency than Parent's theoretical analysis of it

(Baird 2004: 29). In the same vain, it is obvious how the violin bore knowledge of acoustics before Fourier discovered the Fourier series or Helmholz wrote his theories of timbre.

Thus we encounter yet another difference with the computer as an epistemic tool. Computer software does not "bear" any knowledge apart from what has been inscribed in it by humans. It is an entirely simulated system that has been designed from the level of microtransistors to machine language; to operating systems; to programming languages; to graphical user interfaces. It does not contain much hidden affordances or mysteries. The computer mediates: in itself it is never the object of engagement as that object is virtual. The digital instrument maker therefore gets nothing for free, unlike the maker of the acoustic instrument who receives the gift of sonic timbre from the physical properties of the materials he works with. Instead of presenting, the computer *represents*. The computer system models a compositional or timbre-specific system. Without the prior musical language or the knowledge of sound physics, there would be no computational music system. It affords primarily a top-down design process, as opposed to the experimental bottom-up process known to designers working with physical materials.

4.4.2.3 Defining the Epistemic Tool

Considering the above, it is therefore possible to define epistemic tools as instruments (*organon*) whose origins and use are primarily symbolic. The concept of epistemic tools is not intended to define exclusively computer-based tools; it includes all tools that have theoretical origins and typically serve as props for symbolic offloading during our cognitive process. Good examples are the abacus and the astrolabe as symbolic but analogue machines. The chessboard with its black and white pieces is also an epistemic tool (although in this case one might want to swap "tool" with "props"). Epistemic tools are abstractions inscribed in machinery, typically, but not necessarily, serving as cognitive extensions. A theory is therefore not an epistemic tool. Furthermore, it should be made clear that our interaction with epistemic tools is not entirely disembodied as we control them through various interface mechanisms, both analogue and digital, as the astrolabe in Figure 4:1 exemplifies.

Figure 4:1. The astrolabe is an analogue epistemic tool, encapsulating a theory of the world and representing its states

The concept of epistemic tools is related to Norman's cognitive artefacts (1991: 17) and Clark's cognitive technology (2001). It also relates to Mithen's (1998, 2000) and Sterilny's (2004) material and epistemic artefacts. It subscribes to and incorporates critical parts of those definitions.[87] However, unlike them, it emphasises the way technology *changes* the cognitive structure of the user and explores the importance of the highly integrated and complex network of thought systems that constitute any technological object. Furthermore, cognitive artefacts or technologies, as defined by Norman and Clark, are *representational* devices, able to extend our cognitive capacity. Epistemic tools, on the other hand, do not necessarily have to perform any representational function to the user, they can function just as non-representationally as a simple hammer. In the context of Baird's instrumental epistemology [section 4.4.1], the epistemic tool can encapsulate both model knowledge and working knowledge: it is both a symbolic structure that embodies theoretical contents, and an instrument that contains a certain know-how of its intended actions.

Two distinct digital musical systems serve as examples: The drum machine (a step sequencer) represents musical patterns through time (typically designed for 4/4th rhythms and 16th part notes). The user organises the music by turning on or off buttons that are spatially laid out representing the temporal order. It is a clear representational device. Another example is the digital drum (with a physical stick and a rubber membrane) that implements a physical model algorithm[88] of a real drum. As opposed to the drum machine, the drum pad and the drum stick do not have any representational qualities. However, the software part of the digital drum is designed from a particular designer's conceptualisation of the physical world, i.e., what is considered relevant in the simulation of a physical drum, therefore writing off important features and emphasising others, in a way generifying the real instrument. The concept of epistemic tools is therefore more sociological than the cognitive science origins of the cognitive artefacts concept. An epistemic tool does therefore not have to be a cognitive artefact (as defined by Norman) although it surely can be.[89]

Norman's cognitive artefacts, Andy Clark's cognitive technologies or *wideware* (Clark 2001: 141) and Sterilny's epistemic artefacts all focus on the external objects as props in the cognitive process. The mind extends into the environment. Philip Brey (2005) has a different focus which goes further in establishing the cognitive role of external devices. Instead of the coupled systems of Clark and Chalmers, Brey talks about *hybrid cognitive systems* that are part

---

[87] Not everything though. For example the critical question in the hypothesis of extended mind with regards to memory storage and whether it is equivalent to store information in the brain or in a notebook needs further exploration. Such questions are outside the scope of this thesis.

[88] Physical Model Synthesis is a synthesis type used in digital instruments that builds a mathematical model of the instruments' physical structure. Its main concepts are those of the exciter (the drumstick) and the resonator (the skin and body of the drum).

[89] Because the computer is the machine *par excellance* for creating epistemic tools and because most applications made with this machine need a human readable interface (one which is necessarily symbolic and metaphorical), the epistemic tool often serves as a cognitive artefact.

human and part artificial “in which two semi-autonomous information-processing systems cooperate in performing cognitive tasks” (ibid: 392). For Brey, cognition does not have to happen in a mind, as

> [e]ven the user becomes the object of epistemic actions by computer systems, when the system is programmed to ask input from the user in order to perform assigned information processing tasks. In conclusion, then, the computer is a special cognitive artefact that is different from others in that it is capable of autonomously performing cognitive tasks and is able to engage in symbiotic relationships with humans to create hybrid cognitive systems. (ibid: 392)

Brey’s definition of the cognitive is one of algorithmic procedure. The computer is ideal here. However, he claims that when the computer is used for drawing, listening to music or games, its primary function is not cognitive anymore and as such does not qualify as “cognitive artefact.” He proposes a distinction between simulation device and cognitive device. A strong issue will not be taken here with Brey’s definition, apart from stating that it is not of much use in the current context, as it neglects the theoretical implicaions of the object’s design, and this institutes the importance of a new concept that encapsulates the areas of our focus.

In other words, it is important to stress that the rationale for the definition of epistemic tools in this dissertation is not to provide yet another account of how our thought is augmented and even conditioned by external objects, but rather to show how our technological objects are inherently impregnated by systems of thoughts of cultural origins, and in the case of the computer, a system necessarily defined from top-down symbolic instructions. None of the definitions above manage or care to encompass this emphasis on the transductive process that the techno-cultural process of software design implies. Cognitive scientists seem prone to neglect culture, history and the complexity of human-machine relationships in their work, and this has been addressed with the concept of epistemic tools.

4.5 Software: A Cultural Machine

In this section I explore the question of software from the perspective of culture. Such a perspective is never a unified standpoint, but involves sociological and political observations, but perhaps most intriguingly the introduction of a branch of study called software studies or software criticism. Here, software is acknowledged to be a cultural product with a cultural influence, an artefact that can be studied with methods like those used in media studies. Software is not merely a cultural product, but also a cultural machine, an aesthetic machine (or “desire machine” as in Guattari 1993: 26), one that produces culture and is therefore an important factor in the feedback loop of cultural practices. The question of *who* creates software becomes apposite and therefore also an unfolding of relevant strategies of resistance.

4.5.1 The Explicitness of Epistemic Technologies

When digital systems are taken into use, we loose much material properties and knowledge that are given to use in the physical world. Those properties have to be deliberately designed into the system, deliberately excluded, or most likely, not paid attention to, as the rationalistic thinking of abstracting human ergonomic processes filters out what is not considered important. Furthermore, *what* people find important differs. A good example of how simple material processes suddenly become problematic when software systems are introduced, is given by Rosalind Williams where she describes changes within the Massachusetts Institute of Technology:

> Consider the apparently simple project of drawing up class lists in order to allow electronic distribution of class materials to students. The project became less straightforward when it revealed that some professors would allow a few students or colleagues to sit in on a class on an informal basis. Were they officially students, and should class materials be distributed to them? A memo described the dilemma well: "As we move from a mostly analog world in the classroom to an increasingly digital world, some aspects of the physical world need to become more explicit. By becoming more explicit, we often shine a light on inconsistencies between practices, behaviors, and policies." The memo concluded with this question: "When a Faculty member's 'view' of the classroom comes into conflict with explicit MIT policy, or de facto MIT rules, what is the MIT philosophy for resolving such disputes?" (Williams 2000: 650)

We see here how things that would sort themselves out naturally and easily depending on number of seats or number of extra materials to bring into the classroom, now seems quite a complex question. Here technology has intruded the fuzzy sphere, where humans are happy to be flexible and introduced the need for an explicit policy; a policy that might be against many lecturers' private policies or adhere to others'. Williams points out how "technology" and "culture" are often contrasting practices, even within MIT, the dome of modern technology. Culture is seen here as a set of values that is in conflict with technological change. Mackenzie observes: "Technologies are not neutral servants of whatever social or political order chooses to adopt them. Their adoption and operation often involves changes to that order – changes that are not automatic consequences of new technology but must themselves be engineered, often in the face of conflict and resistance" (Mackenzie 1996: 14). But why is this so, and what is wrong with technology? Why can it not adapt as humans do? What is the nature of technology that makes it so rigid and inflexible? A quote by Clark enters perpendicularly into the argument where he proposes:

> a rather profound shift in how we think about mind and cognition – a shift I characterize as the transition from models of representation as mirroring or encoding to models of representation as control (Clark 1995). The idea here is that the brain should not be seen as primarily a locus of inner *descriptions* of

> external states of affairs; rather, it should be seen as a locus of inner *structures* that act as operators upon the world via their role in determining actions. (Clark 1996: 47)

We close the circle here. Software does precisely what the brain was supposed to do according to cognitivism. It is a system of descriptions, of symbolic data that categorises and orders the world – "decodes and deterritorialises" according to Deleuze and Guattari – without the flexibility of adapting to it. Software is a machine that keeps representation of the world as objects in memory and is able to mobilise those representations (the data) through methods of agency. It is primarily descriptive and thus prescriptive in its use. Even if the core of software would be adaptive and emergent (through the use of neural networks for example) the input into the machine will have to be defined preliminarily and such definitions are always already normative classifications or deterritorialisations of the actual situation. There are therefore no hidden mysteries in software, it does what we have told it to do. Its only properties are those properties and variables we consciously inscribe it with and those typically contain references to lower level simulations of processes that have also been designed by us.

4.5.2 Software Studies

A whole dissertation could be written about software studies; i.e., the new critical discipline that has started to operate in the same spirit as film studies, media studies or gender studies.[90] Here, software is seen as a cultural object that can be subjected to critical cultural analysis, not only as an object but also as a cultural or social process. It is outside the boundaries of this thesis to explore those interesting questions; the focus here is on the techno-cultural and the phenomenological relationship we have with technology in general, but software in particular. However, as this field is relevant, providing important insights into the cultural mechanisms of software, a review of software studies is appropriate.

When studying software from the cultural perspective, it is striking how cultural objects, practices and behaviours are becoming increasingly software like. Paradigms and concepts such as object orientation, functions, memory assignments and encapsulation find their way into culture for example in poetry, industrial design or architecture. This is perhaps most obvious in music, where we hear musical styles that originate from the mechanisms in which we program computers. Loops, strong timing, and the aura of virtuality or synthesis are all highly noticeable phenomena. "if/then" and "while/do" are control structures used in abstract machines that are increasingly found in cultural practices who in turn are seen as abstract machines (Deleuze & Guattari 1988: 141).

[90] There are already a few writers doing work in the field (see Kittler 1995; Fuller 2003; Cramer 2005; Mackenzie 2006; Fuller 2008; Manovich 2008). Software studies should be distinguished from writings on software art, which is a related, but more specific field. Software art has been relatively well-represented in critical texts on the digital arts.

Philosophy of technology, in particular the research field called science and technology studies (STS) has a wider coverage than software studies. It includes the study of hardware on its own and non-digital technologies. An interesting difference presents itself in the comparison between the two fields, namely that the study of hardware or non-digital technologies often are studies of objects whose design rationale cannot be accessed. They are black boxes. The study of software however, especially if open source, allows us to access the conscious thoughts of the designer – even if the code is closed, the rationale is more obvious as the thought is represented in the interface, the interaction patterns, and the classes of libraries that are used in the creation. Software studies are thus a more hermeneutic field than technology studies with its origins in philosophy, cultural studies and media studies.

One of the main protagonists in software studies is Lev Manovich. In his book *The Language of New Media* (2001), Manovich analyses the cultural implications of software and talks about "cultural interfaces" (2001: 69) as the interface in which the computer allows us to interface with cultural data, i.e., media of various types presented in the computer through the various multi media techniques. In this early work, Manovich also defines the term software studies as a branch within media studies that originates in the work of Luhan and Innis, but one whose existence calls for a new terminology and conceptual understanding that media theory per definition cannot provide.

> To understand the logic of new media we need to turn to computer science. It is there that we may expect to find the new terms, categories and operations that characterize media that became programmable. From media studies, we move to something which can be called software studies; from media theory — to software theory. (Manovich 2001: 48)

Mackenzie (2005: 6) points out that the study of code shows that code itself is never a static identity and needs a discourse of its own outside computer science. Code is an "unstable volatile material" (ibid.) that has to be studied in its own terms and in its actual *function*. Code is not a dead literary text, but text that becomes alive through its execution by a computer. For Mackenzie it is therefore important that we do not treat code as it was a natural or fixed entity, we have to explore how it is written, what it says, and in what situations it is run.

This optimistic belief in the helping hand of computer science has also been re-evaluated by Manovich himself (Manovich 2008: 5). In retrospect, he observes that computer science is itself a natural product of culture and will therefore have to be understood as a cultural phenomenon, and not something that can be taken *into* cultural discourse. Hayles (1999, 2005), Haraway (1991, 1997) and Turkle (1984) already started working on this in the 1980s and 1990s from the perspective of psychology, cultural studies and media studies, and this work is carried out more specifically by Fuller, Cramer, Mackenzie and Manovich with their recent publications that mark the foundation of the field of software studies.

Manovich's recent publications point at the role software plays in creating, distributing and consuming culture. For him, cultural software is:

> *a subset of application software* which enables creation, publishing, accessing, sharing, and remixing images, moving image sequences, 3D designs, texts, maps, interactive elements, as well as various combinations of these elements such as web sites, 2D designs, motion graphics, video games, commercial and artistic interactive installations, etc. (Manovich 2008: 11)

Software culture is therefore, reciprocally, "a culture where the production, distribution, and reception of most content – and increasingly, experiences - is mediated by software" (ibid: 19). Manovich explores what has happened to media "after software," i.e., how the media change when they are created, distributed and consumed through software. His account is empirical and it shows software use in context, but we need to look elsewhere to get a more philosophical understanding of the fundamental role software plays in people's cognitive and aesthetic functions, and how its mechanisms affect or influence cultural processes.

Florian Cramer is one of the first researchers dealing with software studies. He is interested in how, through software execution, words become flesh. For Cramer (2005), software history is always a cultural history; it is not a history of technology, but should rather be seen as a conceptual and cultural narrative. Cramer defines software as a cultural practice made up of:

> (a) algorithms, (b) possibly, but not necessarily in conjunction with imaginary or actual machines, (c) human interaction in a broad sense of any cultural appropriation and use, and (d) speculative imagination. (Cramer 2005: 124)

Cramer explores our obsession with software, the origins of this obsession of words having a life on their own through various cultural practices such magic, Kabbalah, music, and experimental poetry. He calls this "imaginative computation," discovering that computation and its imaginary are rich with contradictions, and impregnated with metaphysical and ontological speculation.

Matthew Fuller has dealt with the signification of software in various publications (2003, 2005, 2006, 2008). Recently Fuller (2006) has explored how art methodologies, i.e., the artists' self-conscious understanding and unpacking of their own practices, are able to break up the homogenisation of the computer as a universal device (that tries to "do all things for all people" Norman 2000: 10) through the artists' non-conventional use of the computer. Fuller further points out that the computer is not a singular object anymore, as normally it interacts with a multiplicity of servers, external software, networks and various forms of social, cultural and commercial systems (ibid: 2). Software has become a glue that connects a heterogeneity of cultural practices. The task at hand is to understand software and its role in our culture. He succinctly introduces the concept of interrogability, i.e., a quality that allows systems to evaluate themselves. Such reflexivity implemented in software can help users to understand their relationship with the software as it opens up for its being.

> Interrogability is a quality that provides a way for software to make an account of itself, to allow the user to enter into an engagement with the epistemological and reality-forming dimensions of an application or piece of code. This is a quality shared to varying extents by many activities involving the mixing and manifestation of cognition and materiality, but can also be understood as key to much art practice. (Fuller 2006: 22)

Fuller points out that code is not something that exists outside culture, made by beings unaffected by culture: "Instead, what is suggested by such work is not some digital equivalent of the translation of the digital holy books into the vernacular, but rather that all code is already a vernacular, already heavily cultured" (ibid: 28).

Interrogability, a "desublimation of code" (ibid: 27), or the opening up of code, is a process of materialising the media, of opening up its conditions and origins. Code is made sensual as well as analytically understood. Art methodologies are therefore useful in rejecting the homogenisation of digital practices and open up for infinite pathways out of the technological enclosure that Norman warns about. For Fuller, however, the solution is not necessarily a world full of gadgets (as in ubiquitous computing) but an opening into the conditions and sensuality of software.

4.5.3 Livecoding

Livecoding is the artistic practice of performing with computers (music, visuals, robots, etc.) by programming in realtime, typically projecting the code with overhead projectors for the audience to watch. Livecoding is the ideal example of how culture resists the technological script and rewrites it in sub-cultural contexts, such as the livecoding performance. It is an obvious response to the complaints audience made after the immediate novelty of "laptop performances" in the late 90s wore out. People demand to see the relationship between the musician's gesture and the sound, yielding a twofold response: opening up for embodied expression (as exemplified by NIME) on the one hand, and opening the gate to the musician's stream of thoughts by projecting his/her symbolic thinking in the idiom of programming language, on the other.

Livecoding has been dealt with in excellent texts by many of the protagonists in the field (Collins et al. 2003, McLean 2004; Sorensen 2005; Ward et al. 2003) and there is no reason to expand on the topic here. Suffice to say that it has become a cultural practice defined by the technological affordances of hardware and programming languages, by the structure of algorithmic thinking, and in turn it projects those constraints onto the audience who has to keep up with the, in many cases, hieroglyphic symbols and commands of livecoding programming languages.[91] Furthermore, livecoding is covered extensively in section 7.4, where users of various audio programming language talk about their practices.

4.5.4 Strategies of Resistance

Digital music software are epistemic tools. They inscribe a musical language, a theoretical outlook on which elements are important in music, how musical creativity takes place and which steps are involved in the musical process both in composing and performing. As Latour and Akrich show (Akrich & Latour 1992), there are complex networks at work in which both human and non-human actors define how the software is scripted and presented to the users and what readings of the technology the users are able to perform. Technical objects, such as software, can thus define actors and relationships between actors. The object delegates prescriptions to its users through user manuals, interface design and system affordances. It is also inscribed in a larger context which is the culture of *using* this tool, and its status in the general music community. The software enrols musicians into production. It describes the world of music and its language through the interface mechanisms of the tool. Actor-network theory talks about antiprograms and de-inscriptions when users reject the tool and resist its inscriptions. The script of a technology is written from a specific world-view, a program of actions, by the designer. When users do not share that world-view, they have an antiprogram with regards to the technology. If such a user chooses (or, as often is the case, is forced) to use the technology, she will react with a de-inscription of the technology, a re-interpretation of it through the rejection of its script (ibid: 259).[92] This rejection of the intended use of the technological artefact through the making of a new meaning of it typically yields unforeseeable social patterns. A good example is the popular use of text messaging which was not intended to be used as a social communication channel between people but rather an alert system from service provider to customer (Trosby 2004).

Suchman (2002) defines as *local improvisations* the forms in which things (devices, technical systems or organisational forms) are taken into work according to specific social circumstances.

---

[91] In these written words, on Saturday night December 13th, 2008, the author is sitting in a hotel room in Asturias, Spain. On the screen is a live screen-cast of Alex McLean live-coding from his sofa in London on a stream that is projected as part of the Pixel festival in Bergen, Norway. The audience (in Bergen and online) is able to enter the screencast through chatting. Curious to test the medium, I asked McLean, through the chat channel, to change the delay time of one of the instruments and increase its amplitude. This request was approved, McLean responded with a change in his code, resulting in a sense of interactive performance where audience members are able to affect the musical performance through feedback and comments.

[92] The antiprogram or consumer resistance to products are related to recent studies on how *non-users* also participate in defining the life of an artefact. Resistance in itself can yield socio-technical changes. Non-users can be further defined into the "resisters," the "rejecters," the "excluded," and the "expelled." (Kline 2003; Wyatt 2003).

> Here I want to underscore the extent to which this is the case not only for the export of solutions from the developed to the developing world, but from any site of technology production to other sites of technologies-in-use. The greater the distance – geographical, economic, cultural, experiential – the greater the need for reworking is likely to be. (Suchman 2002: 139)

Suchman concludes that we need alternative readings of what "resistance" means in the cultural context. There are strong politics at work, for example when technologies are introduced in the developing countries, where the change being proposed are often "solutions" to problems that have been wrongly defined, often causing further (and now technocratic) complex problems. The question is *who* performs the analysis of the problem, who designs the change in social practices, how is the new technology presented, and which are the possibilities to adapt and modify the technology to local needs. In short, we need to question the meaning of these terms: the owner, the designer, the imposer, the user, the consumer, the winners, and the losers.

In terms of the culture of musical software there are various strategies of resistance. Cracking software is one way to assault the power of the commercial software houses. Cracking is a reverse-engineering process that can take a long time, time which the cracker could easily spend by working a few hours, buy the software and then proceed to make music. However, the aim of the cracker is not to save money, but to attack the commercial, and closed-source, companies and in the process gaining respect within the community of other hackers and musicians alike (Turkle 1984). Another interesting strategy is deliberate misuse. Binary files can be corrupted, disks scratched, and software modified to the degree that it does not function as intended. This malfunctioning of software and hardware is here seen as a new platform with new affordances and constraints in which to express oneself. This platform is not generic and universal anymore: it becomes a personal and unique tool of the experimental user. In terms of aesthetic responses to the culture of software, we see genres of that deliberately work with the malfunctioning of technology such as the glitch or the noise movements (Cascone 2000; Sangild 2002, 2004; Goriunova & Shulgin 2008; Moradi 2009).

Perhaps most interesting and positive (in terms of generativity) of the resistance strategies is the open source movement. Here programmers from all over the world collaborate on projects that are not the sole design of a single person or commercial company.[93] As opposed to the binary files one gets when buying commercial software, open source software contains all the source files that makes the project transparent, open, malleable and adaptable to various situations. The onus will then be up to the local, non-Western musician to adapt the software to his or her needs, and not on a group of designers in the West trying to grasp the subtleties of all

[93] Although each open source project defines its own rules on what code should be included and what excluded, and who has the powers of rejecting or accepting code into the core of the project. This is necessary for the project to have longevety, and arguably to keep it consistent and elegant (yes, code is an aesthetic object).

existing traditions. Open source software tends to be more *expressively open* for this reason. When there are large numbers of designers working on the same code base, each with their ideas about music theory, creativity and performance, they have to settle on the lowest common denominator where the platform is general enough for all users. In section 7.4 and 7.5 few such environments are introduced through interviews with their users and creators. We will see how they serve as open engines for the creation of constrained compositional environments or instruments. It will also become clear in what sense they become extensions of the musicians' mind through it becoming the language in which they think.

4.6 Beyond Technesis to Techno-Practice

This chapter has questioned the epistemic nature of tools, describing how the technological system conditions our thinking and performance. Our practical and empirical explorations in part II of this thesis will deal with this question of the encounter with technology, how we incorporate it and, finally, how we are able to abstract our knowledge of it in the terms of its discourse. Deleuze (2005) deals with this problem of first encounter and the difference between (embodied) sense and (conceptual) recognition. "Something in the world forces us to think. This something is an object not of recognition but of a fundamental *encounter*…. In this sense it is opposed to recognition. In recognition the sensible is not at all that which can only be sensed, but that which bears directly upon the senses in an object which can be recalled, imagined or conceived" (ibid: 139). What Deleuze is exploring is the difference between that which can be understood rationally and that which can only be sensed, but later understood. This object "forces [us] to pose a problem" (ibid: 140) in order to understand it. After exploring the nature of both acoustic and digital instruments, their interface modalities, ontogenesis and taxonomies, we will introduce empirical findings derived from two online surveys and one interview survey we conducted on musical instruments in general and the ixiQuarks environment in particular.

In the preceding chapters we have explored various contemporary theories of technology and the body that originate in different scientific discourses with distinct philosophical foundations, but what they have in common is an understanding of the human as essentially interdependent upon the techno-environment in which it finds itself (or is "thrown" into). The cultural critic of technology and science, N. Katharine Hayles summarises elegantly how these theories portray cognition as something that:

> does not issue from the mind alone but extends beyond the neocortex into the lower brain, the limbic system, the central nervous system, and the peripheral nervous system. Cognition also reaches out into the techno-environment, dissolving the boundary between inside and outside into fluid assemblages that incorporate technical artefacts into the human cognitive system, not just as metaphors but as working parts of everyday thoughts and actions. (Hayles in Hansen 2000: vi)

We have defined technology as constitutive, as prosthesis, as the original externalisation of the human, its condition, and we have seen this happening in various distinct and unrelated discourses. In *Embodying Technesis,* Mark Hansen warns about a particular danger in our attempts to understand technology and its meaning for humans. Hansen coins the word *technesis* for the "putting-into-discourse" of technology, where modern cultural studies (post-structuralist in particular) incorporate technology into its discourse as if in order to dissolve its materiality and turn it into a pure discourse. Cultural theory tends to perform a "reductive strategy that allows for a progressive assimilation of technology to thought" (Hansen 2000: 4) and maintaining the modern philosophical priority given to thinking as opposed to embodied existence. This is a concern for Haraway as well in her cyborg manifesto, where she states: "'Textualization' of everything in poststructuralist, postmodernist theory has been damned by Marxists and socialist feminists for its utopian disregard for the lived relations of domination that ground the 'play' of arbitrary reading" (Haraway 1991: 152). The danger is that we treat technology as textuality, as a structural centre of our culture and embodied being, but forget about its real and material exteriority.

Sidestepping the question of whether Hansen's danger really poses any serious peril, I believe that it is possible to refrain from such *technesis* by engaging with different discourses and actual musical practices and by giving a thorough philosophical, sociological, and historical perspective on contemporary thinking about technology. Moreover, our emphasis has been on embodiment, on non-linguistic or pre-conscious experience that per definition only becomes discursive post hoc. Hayles' incorporation/inscription model examines this dynamic clearly (see section 3.5.1). However, the aim of this thesis is not merely theoretical; it is also to understand technology in praxis, its influence on creative thought, and how it constitutes an environment in which modern musicians work. Also to be explored is how musicians adopt or adapt to technology through various cultural and technological mechanisms. Indeed, we will treat technology as essentially external materiality, as something that we *encounter* afresh, but adapt to and learn to think in. From this perspective, a technological object belongs to a symbolic system like language, and as an analogy to language, it is something that we learn through contextualised embodied practice – though habituation.

4.7 Summary and Conclusion

The two preceding chapters were instrumental in laying the ground for our definition of epistemic tools. The nature of technology had to be understood as an evolutionary force alongside human culture and theoretical tools introduced with which technological artefacts can be analysed [chapter 2]. The focus was then pointed at the relationship between the body and technical objects (both acoustic and digital) and key theories, such as enactivism and the extended mind, were introduced. These theories are crucial elements in the analysis of the distinction between digital and acoustic instruments [chapter 3]. This chapter presented a definition of epistemic tools as physical objects, whose main function is, as symbolic systems, to extend human cognition. Through the work of Baird I established the possibility of "thing knowledge;" of how physical structures can bear knowledge that is not yet abstracted into theoretical forms or, alternatively, providing a different or supplementary account of the theory due to its mechanistic or physical properties.

The digital instrument was succinctly defined as a theoretical construction where all knowledge of its properties, as a sound or pattern generating tool, has to be consciously and symbolically inscribed into the instrument in the form of written code (which we can also call "script" that will be acted out, or even "code" that will be decoded in its use). Although one could create sophisticated tangible hardware as interfaces to the epistemic tool, the interaction will always be primarily symbolic, and taking the form of representational patterns. This symbolic relationship with the instrument shifts the learning process from the nature of embodied training to disembodied habituation of the symbolic structure, the habitual model of the software. It also means that, metaphorically speaking, much of the "music" resides in the tool rather than in the fingers of the musician. Conversely, with acoustic instruments the "science of sound" is given to us through the instrument's physical properties. Acoustic instruments do not encode musical languages to the same degree as digital instruments (although some do more than others, the piano being a good example). This means that in most traditions, during performance the musical language is stored in the musician's fingers and not at a conscious symbolic level (Sudnow 2001; Gibson 2006).

Implied in the theoretical provisions of this thesis is the understanding that *techne* and *episteme* are not to be separated, that they are in fact often inherent in the same cultural objects, and therefore influence our activities on both phenomenological and epistemological levels. The traditional division of science (*episteme*) and technology (*techne*) originating from early Greek philosophy is rejected. This is done with reference to 20th century Continental philosophy (such as Heidegger, Derrida and Stiegler), and contemporary cognitive science (such as Varela, Clark and Dennett). The account given above of digital versus acoustic instruments has been rather generic and detached from actual technological contexts. As promised above, the aim is not to be guilty of *technesis* (the purely abstract account of technology where it becomes swallowed up in discourse) but engage, on a practical level, with the subtleties that we invariably stumble into when analysing musical technologies. The next part of this thesis will keep this promise; the expedition into the field of musical instruments is launched.

PART II

Chapter 5

A Theory of Digital Musical Instruments

A comprehensive analysis of digital instruments is provided in this chapter. After the introduction of an extensive survey on the phenomenology of musical instruments, a comparison in the design, use and reception of acoustic, electronic and digital instruments is presented. Three selected instruments (the saxophone, the MiniMoog and the reacTable) are analysed and explored through actor-network theoretical (Latour) and genealogically (Foucault & Nietzsche) inspired analysis. This analysis paves the ground for a taxonomical analysis of digital musical instruments with emphasis on bodily gestures or expressivity. Finally, a dimension space for the analysis of digital musical instruments as epistemic tools *is presented.*

5.1 Introduction

Even simple physical instruments seem to hold more mystery in their bodies than the most elaborate computer programs. (Evens 2005: 190)

This chapter starts by outlining the results of a survey conducted with users of creative digital systems.[94] As these systems are often used as meta-tools for creating other systems, the users are by necessity creators and inventors of artefacts that range from generative musical systems, instruments, to music software. The survey focuses extensively on the phenomenology of digital and acoustic musical instruments, and through comparisons of them a descriptive theory of digital musical instruments is developed. Due to the methodological strategy of this research to investigate the digital as compared with the acoustic, the findings of the survey are instrumental for the subsequent arguments in this chapter, but also relevant to other parts of the thesis.

Referencing Stiegler's theories of the epiphylogenesis [chapter 2] as the evolutionary history of technics, we explore the question of historicising cultural artefacts (or designed objects) through a phylogenetic (evolutionary history) analysis [section 5.3]. Having found ephiphylogenesis a revealing overview, but not sufficiently specific theory for a solid understanding of digital instruments, the ontogenesis (history of structural change – of origins) of three different types of musical instruments is explored: the saxophone, the MiniMoog synthesizer and the reacTable [section 5.4]. These instruments are chosen because of their recent origins, resulting in a relatively complete documentation of their history. The saxophone

[94] Here we mean both environments such as SuperCollider, Pure Data, ChucK, CSound, Max/MSP, Impromptu, etc. where the user is the creator of both music and tools through the same process, and commercial software products that provide a more "subscriptive" policies of use.

is arguably the latest invented major acoustic instrument (150 years ago), the Moog synthesizer one of the most famous and interesting examples of electronic instruments, and the reacTable is being born in realtime right in front of our eyes, and serves as a good example for the analysis of a digital musical instrument.

The genealogical or ontogenetic analysis of musical instruments is revealing, but it is necessarily a study of details, of the singular. In order to get a more general picture I resolve to the creation of taxonomies as they appear to us as both designers and users of these tools. The taxonomies identify similarities and differences to acoustic instruments, but also demonstrate the complexity and impenetrability in the understanding of digital technologies. The topics of metaphors, embodiment, mapping, etc. are explored. The chapter introduces a dimension space for the analysis of epistemic tools which can assist in future design, composition and analysis of digital musical instruments. This dimension space makes explicit the epistemic properties of digital musical instruments and is used in chapter 6 in the analysis of the ixiQuarks.

Technology has influenced music since humans started building musical instruments. Various sections of this thesis have pointed to the fact that technology can never be pure or independent of society. It is always social, political, economical, psychological and historical – and this becomes crystal clear when analysing the history of musical instruments. Throughout history, new musical instruments have inspired new musical techniques, performance styles and genres. The arrival of the pianoforte changed the musical landscape through the 18th and 19th centuries in Europe. For some, especially harpsichord enthusiasts, it was an intrusion that had many negative and irrevocable consequences (DeNora 1995). But the creation of new instruments is always a dialectic process that happens in a living culture. Changes in cultural structures, such as new social forms, technological advances, or architectural improvements, invariably yield new developments in instrument making. For example, when the pianoforte was being developed, new demands were coming from composers in relation to the dynamics of the instrument, but *also* from organisers of social events who were increasing the size of musical events (Campbell 2004). With larger and more populous concert halls the instruments had to be louder. This requirement for louder instruments meant that ergonomic considerations were often sacrificed at the cost of more thunderous performance tools (Jordà 2005: 169; Livingston 2000). Modernity was not only obsessed with amplitude: the rationalisation of the instruments through improved engineering techniques made the instruments more reliable and perfect in pitch, timbre, and control. However, this came at the price of sacrificing important "fuzzy" characteristics.[95] For example, the European flute was developed with linked-keys and valve

[95] "Fuzzy" here is meant to signify elements that are tacit in the design of the instrument. It is no coincidence that traditional composition notation does not have a language to write for those characteristics of the instrument.

mechanisms instead of fingers pressing holes. This made the instrument more uniform and tones were cleaner, but the dexterous finger control over the hole was lost, a feature that allowed for vibrato and slight pitch changes in the note. Instead of analogue continuous control over the hole, it became binary: either open or closed, on or off (Ahrens 1996).

At this moment in time, we witness an extraordinary boom in the research and development of digital musical instruments. This explosion is partly caused by advances in tangible user interfaces and new programming paradigms that provide new affordances for musicians, composers, designers and programmers to explore. It is also driven by the curiosity and aesthetic demands of musicians who, in a Vareseian manner, demand new instruments for new musical ideas. A plethora of audio programming languages and environments now exist that allow musicians to develop their own instruments or compositional systems.[96] These environments work seamlessly with new sensors and sensor/controller interfaces and a majority of them are open source and free. This makes the design and implementation of novel controllers relatively simple – a transformation of the situation as it was merely a decade ago. On the net there is an abundance of information (languages, protocols, interfaces, and schematic diagrams) needed for the design of such instruments, resulting in people altruistically (which, in the open source culture, results in various personal advantages) posting their inventions up on blogs, wikis, and video sharing websites in order to spread their inventiveness. It is an enthusiastic culture of sharing information, code, instrument design, and, of course, music.

Nevertheless, as noted in the recent *Sound and Music Computing Roadmap* (Serra et al. 2007), unlike the Theremin in the 1920s, there are very few new musical instruments that have become popular with the general public. In fact, the new musical instruments we find are not even digital: consider the turntable, the electric string instruments, etc. In the aforementioned research document, the weakness of digital interfaces is identified as sound control, mapping, ergonomics and interface design; all topics that are defined in this current thesis as the elements of digital musical instruments that define them as epistemic tools. It is therefore appropriate to investigate why, or indeed *whether*, this really is a weakness.

5.2 The Acoustic, The Digital and the Body: A Survey of Musical Instruments

5.2.1 Introduction

The explosion in research and development in musical tools (where most of the interesting research comes from academia and DIY people – as opposed to the commercial sector) is also a response to the unsatisfactory feeling the audience of computer music performances tend to experience as a result of the lack of perceptual causality between the bodily gestures of the

[96] The process is transductive, where ideas resulting in cultural forms are shifted between actors.

musician and the resulting sound. This lack of causality is not to be found in acoustic instruments; artists are realising that it is not just a matter of "getting used" to the new mode of disembodied performance of the "laptop performance." Fundamental to musical performance is the audience's understanding of the performer's activities, a sharing of the same "ergonomic" or activity space. Recent research in brain science on mirror neurons and their role in music and musical performance (Molnar-Szakacs & Overy 2006; Godøy 2003; Leman 2008); in sociology, of thinking as collective activity (Hutchins 1995); and in biology of enactivism (Varela et al. 1991) all lead to a deeper understanding of what constitutes embodied performance. This development certainly relates to the realisation in cognitive science of the central role of the body and the environment in human cognition.

In a nutshell, the situation is this: after decades of computer music software focusing on the production environment (of composing or arranging musical patterns), there is now a high demand and interest in the computer as a platform for the design of expressive musical instruments. Instead of prepared "projects" or "patches," the desire is for instruments that are intuitive, responsive and good for improvisation. Instruments that provide the potential for expression, depth and sophistication, encouraging the musician to strive for mastery. In its current form, improvisation with computers is typically characterised by long, extensive parts and slow formal changes. Improvisation with acoustic instruments, on the other hand, can be fast, unpredictable and highly varied in form, even within the same piece/performance. It is a clear example of how technological affordances (in this case limitations for fast response time) have conditioned artistic expression and resulted in the diverse musical styles. The demand is now for digital instruments that have expressive depth, ease of use, and intuitive flexibility (all criteria that support the design of the ixiQuarks). Furthermore, echoing Jordà's arguments (Jordà 2005: 201), a musical instrument should necessarily be able to perform "bad" music, i.e., designers should refrain from the desire to inscribe so much music theory and automation into the instrument that whatever the performer does, the music sounds good.

5.2.2 The Aim of the Survey – The Research Questions

In the light of the differences between acoustic and digital instruments we (Enrike Hurtado Mendieta and myself) decided to conduct a survey[97] with practitioners in the field – both acoustic and digital musicians (which are categories that typically overlap in the case of the digital musician). The survey is specifically concerned with people's *experience* of the differences in playing acoustic and digital instruments. The approach was phenomenological and qualitative: the interest was how musicians or composers describe their musical practice, and the relationship they have with their instruments, whether acoustic or digital. By

[97] Information about where to find the survey can be found in Appendix III

deliberation the "digital instrument" (such as a sequencer software, a graphical dataflow language, a textual programming language or a tangible sensor interface coupled to a sound engine) was not defined a priori, as the interest was in how people themselves define the digital, the acoustic, and the relationship between the two. It is intriguing to explore how people rate the distinctive affordances and constraints of these instruments and whether there is a difference in the way people critically respond to their design, if people relate differently to the inventors and manufacturers of these two types of instruments. Another line of enquiry questioned if musical education and the practicing of an acoustic instrument result in a different critical relationship with the digital instrument. How does instrumental practice change the ideas of embodiment and does it affect the view of the qualitative properties of the computer-based tool? Finally, the survey investigated how people related to the chaotic or "non-deterministic" nature of their instruments (if they saw it as a limitation or a creative potential) and whether they felt that such "quality" could be designed *into* digital instruments in any sensible ways.

5.2.3 The Participants

The survey tried to reach as wide an audience as possible: from the acoustic instrumentalist that has never touched a computer to the livecoder that does not play a traditional instrument. More specifically, the survey was interested in learning from people with a critical relationship to their tools (whether acoustic or digital); people who modify their instruments or build their own, with the aim of expressing themselves differently musically. Of special interest was to learn how people that have used ixi software experience it in relation to the questions mentioned above.

The results of the survey came precisely from the group that it was aimed for. The majority of the participants actively work with one or more of the audio programming environments mentioned in the survey. Very few replies were from people that use exclusively commercial software such as Logic, Reaktor, Cubase or ProTools, although insights from them would have been very welcome.

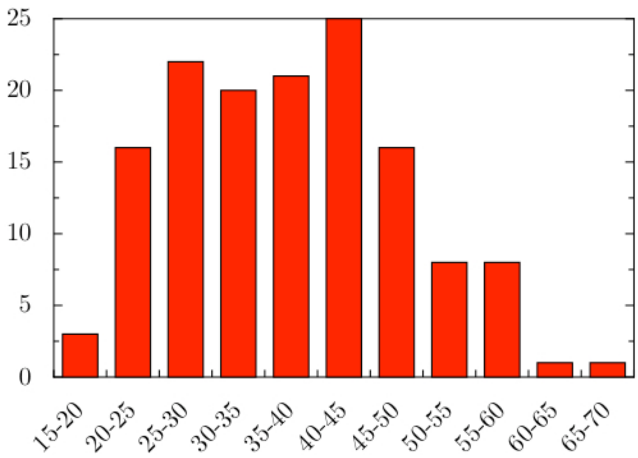

Figure 5:1. The age distribution of the survey participants. The Y axis shows number of participants; X axis the age-range

The survey received 250 replies, mainly from Western Europe and North America, but a considerable amount also from South America and Asia. There were only 9 female participants, but it is outside the scope of this research to explore the reason behind this fact. Of interest was the age of the participants and how long they have been playing music. It was surprising how relatively high the mean age was (37 years), distributed as shown in Figure 5:1.

5.2.4 The Questionnaire

The survey was a qualitative survey where the main focus was on people's description *of* and relationship *with* their musical tools. It was divided it into six areas:

1) **Personal Details**: a set of demographic questions on gender, age, profession, nationality, institutional affiliation, etc.;

2) **Musical Background**: questions on how the participants define themselves in relation to the survey (musician, composer, designer, engineer, artist, other), how long they have played music, musical education, computer use in music and musical genre (if applicable);

3) **Acoustic Instruments**: questions about people's relationship with their instrument. (Which instrument, how long they have played it, etc.). The survey asked whether people found their instrument lacking in functionality; if they thought the instrument has "unstable" or "non-deterministic" behaviour; and if so, how they related to that. Also explored was how well the participants knew the history of their instrument and which factors affected its design. Would it be beneficial if the human body was different?

4) **Digital Instruments**: which operating system people use and why; what hardware (computer, soundcard, controllers, sensors); what music software; and whether they have tried or use regularly the following audio programming environments: Pure Data, SuperCollider, ChucK, CSound, Max/MSP, Plogue Bidule, Aura, Open Sound World, AudioMulch and Reaktor. The survey asked about programming experience and why a specific software had been chosen. Further, it was interesting to see if and how people use Open Sound Control (OSC) and whether people use programming environments for graphics or video in the context of their music making.

5) **Comparison of Acoustic and Digital Instruments**: here the concern was with the difference of playing acoustic and digital instruments, and what each of the types lacks or provides. Questions were asked regarding people's dream software; what kind of interfaces people would like to use; and then if people found that the limitations of instruments are a source of frustration or inspiration. Did that depend on the type of instrument?

6) **ixi software**: These questions support the findings in chapter 6.

People were free to answer the questions they were interested in and to skip the others, as it would not make sense to force an instrumentalist to answer questions about computers if he/she has never used one. The same goes for the audio programmer that does not play an acoustic instrument.

5.2.5 The Methodology

The survey was introduced on the ixi website and it was posted on the ixi mailing list. It was also sent to various external mailing lists (including SuperCollider, ChucK, Pure Data, Max/MSP, CSound, AudioMulch, eu-gene, livecode). Figure 5:2 shows participants' preference for musical tools. We also asked friends and collaborators to distribute the survey as much as possible, and we contacted orchestras and conservatories and asked them to post the survey on their internal mailing lists. The survey could be answered in nine languages, but the questions themselves were only available in English or Spanish. Unfortunately, as a quarter of the visitors on our website originate from Japan (and where there is a strong culture of using audio programming languages) we did not have the resources to translate the survey into Japanese.

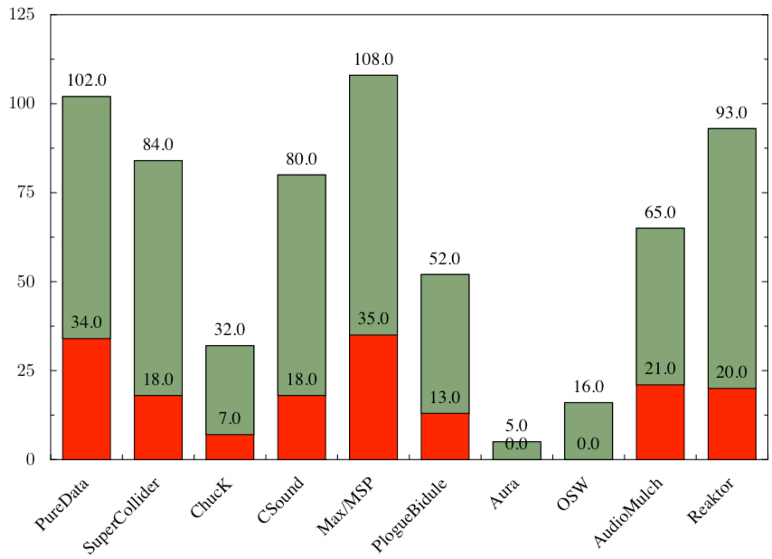

Figure 5:2. Tool-usage of the survey participants. The higher number shows how many people use or have used the specific tool. The lower number shows their tool of choice.

After three months of receiving replies, we started working on the data. We analysed each reply and put it into a database. All the quantitative data was retrieved programmatically into the database, but as the questionnaire was largely qualitative (where people write their answers in the form of descriptive narrative), we had to interpret some of the data subjectively. Here we created a continuum (marked from 1 to 5) where the following "archetypal" elements were extracted: a) Abstract vs. graphical thinking: i.e., the tendency for working with textual vs. dataflow programming environments. b) Preference of self-made vs. pre-made tools. c) Embodied vs. disembodied emphasis in playing and making instruments or compositions. d) Whether the person is a "techie" vs. "non-techie" where we tried to extract the level of people's "computer-literacy" and programming skills. e) Academic vs. non-academic. We were interested in the question of how these audio programming environments (that mostly have their origins in academia) have filtered out into the mainstream culture.

In order to test the reliability of this method, five random replies were selected and given to five different people for analysis. The comparison of the five people's analyses were satisfying. The results were almost identical, with some minor differences on the left or right side of the continuum, but there was no instance of opposite interpretations.

5.2.6 The Evaluation and Findings

5.2.6.1 The Survey Participant

The high mean age of the survey participants has already been mentioned. From analysing the demographic of the people answering the survey, the typical survey participant could be divided into two groups to which more than 90 percent of the participants would belong:

Group 1) People that have had over 20 years of studying music and playing acoustic instruments, therefore typically 30-40 years of age or older. They have been using the computer for their music for at least 10 years and usually have some form of programming experience. Many write their own software or use the common audio programming environments available today. This group has thought much about their instruments and why they have chosen to use digital technologies in the production and performance of their music.

Group 2) This group consists of younger people that typically have grown up with the computer and are highly computer literate. Many of them have not studied music formally or practised an acoustic instrument but use the computer as their instrument or environment for creating music. Here, of course, one could view all the spent time in front of the computer screen (performing any task) as part of the musical training.

Naturally, there was some degree of overlap between the two groups.

It is illustrative to look at which operating systems the participants are running their tools on, and here we see that 45 people use Linux/GNU; 105 use Windows; and 88 use Mac OS. Of these 16 stated that they use both Mac OS and Linux/GNU; 30 use both Mac OS and Windows; 25 Linux/GNU and Windows; and 7 used all three. Other operating systems in use were NeXTstep, BSD and Solaris, one person each system.

5.2.6.2 Acoustic vs. Digital Instruments

In terms of embodiment and inscribed theoretical content of the epistemic tool, the questions of concern here are how people experience the different qualities of acoustic and digital instruments. Apart from the experiential and perceptual differences, do people think that the tools enframe, inform or influence their work?

Many people found that an important difference in these two types of instruments lies in the fact that the digital instrument can be created for specific needs whereas the player has to "mould oneself" to the acoustic instrument. As the composed digital instrument can be very work specific, it lacks the generality of acoustic instruments. In this context, the question of originality came up frequently. People found it possible to be more original using the composed, digital instruments, precisely because of the lack of history and traditions. As one survey participant put it: "when playing an acoustic instrument, you are constantly referring to scales, styles, conventions, traditions and clichés that the instrument and the culture around it imposes on you. A musician can just play those conventions in autopilot without having to THINK at all.

It's easy and unchallenging." This, of course, is a double-edged sword as it is difficult in a live performance, using software tools, to refer to the musical reservoir in the spur of the moment. All such decisions have to be pre-programmed (i.e., pre-planned) thus subjecting the performance to a certain degree of categorisation and determinacy. The issue of originality also points to the limited scopes of some commercial software environments where users are virtually led into producing music of certain styles.

One participant talked about the enriching experience of learning the vocabulary and voice of an instrument like the viola to its finest details, whereas with computer technology the voice is too broad to get to know it thoroughly. In this vein, other participants expressed the wish for more limited expressive software instruments, i.e., not a software that tries to do it all but "does one thing well and not one hundred things badly." They would like to see software that has an easy learning curve but incorporates a potential for further in-depth explorations, in order not to become bored with the instrument. True to form, the people asking for such software tools had a relatively long history as instrumentalists.

Positive	Negative
Tactile feedback	Lacking in range
Limitations inspiring	No editing out of mistakes
Traditions and legacy	No memory or intelligence
Musician reaches depth	Prone to cliché playing
Instrument 2nd nature	Too much tradition/history
Embodied experience	No experimentation in design
No latency	Inflexible - no dialog
Easier to express mood	Less microtonality or tunings
Extrovert state when playing	No inharmonic spectra

Table 5:1. The acoustic instrument: reports of its positive and negative properties.

Positive	Negative
Free from musical traditions	Lacking in substance
Experimental - explorative	No legacy or continuation
Any sound and interface	No haptic feedback
Freedom in mapping	Lacking social conventions
Designed for specific needs	Latency frequently a problem
Automation and intelligence	Disembodied experience
Good for composing with	Slave to the historical
Easier to get into	Imitation of the acoustic
Not as limited to tonal music	Introvert state when playing

Table 5:2. The digital instrument: its positive and negative properties

Many people were concerned with the arbitrary mappings in digital instruments (Hunt et al. 2002). There are no "natural" mappings between the exertion of bodily energy and the resulting sound. One participant described digital instruments as "more of a mind/brain endeavour." He continued, "it is more difficult to remove the brain and become one with the physical embodiment of performing." Others talked about the perception of making the physical object vibrate and feeling the source of the sound in direct and natural ways being; something that computer systems lack with their buttons and sliders, soundcards and cables going out to the remote speakers. The computer was often seen as a symbolic system that can be configured differently according to situations, thus highly open, flexible and adaptable to infinite situations. The acoustic instrument was defined, in this context, as a non-symbolic tool that one has to establish an embodied relationship with; a relationship that is carved into the motor memory of the body and not defined by the superficial, symbolic and conversational relationship we have with our digital tools.

Some participants expressed the sense that they found their time was better spent working with digital technology, creating music or “experimenting with sound” rather than practising an acoustic instrument; one would be instantly creating music, even of releasable quality, as opposed to the endless hours spent on practicing the acoustic instrument. Conversely, others talked about the dangers of getting side-tracked when using the computer, constantly looking for updates, reading mailing lists, testing other people’s patches or instruments, even ending up browsing the web whilst trying to make music.

Some talked about the “frightening blank space” of the audio programming patcher (its virtually infinite expressive possibilities) and found retreat in limited tools or acoustic instruments, whereas others were frustrated with the expressive limitations of the acoustic instruments and craved for more freedom and open work environments. Naturally, this went hand in hand with the use of environments such as SuperCollider, ChucK, Pure Data and Max/MSP vs. preference of less open or more directive software like ProTools, Cubase, GarageBand, Fruityloops, etc. Here one might consider personality differences and cognitive styles as important factors in how people relate to the engineering tasks of building one’s own tools but also with regards to the guidance the tool affords in terms of musical expression (Eaglestone et al. 2008).

Another issue of concern was latency. An acoustic instrument does not have latency as such, although in some cases there is a delay between the energy applied and the sounding result. In digital instruments there might be up to 50 ms latency that people put up with when playing a hardware controller; many seconds latency in networked performances; but also the organisational latency when opening patches, changing effect settings or in livecoding where one has to type a whole function before hearing the result (typically by hitting the Enter button). This artificial latency is characteristic of digital instruments (although also to be found in some acoustic instruments, such as the church organ), but it is not necessarily always a negative property apart from the situation with using hardware controllers. This state of affairs has improved drastically in recent years.

Related to this, some people reported discontent with the uncertainty of the continuation of commercial digital instruments or software environments. Their production could be discontinued or not supported on new operating systems. Unless open source is used, the proprietary protocols could become unsupported, rendering the instruments objects of archaeology. In this regard, acoustic instruments have longer lifetime, which makes practising them more likely a continuous path to mastery. General trends from these comments are presented in Tables 5:1 and 5:2.

5.2.6.3 Affordances and Constraints

Here the investigation explores whether people relate differently to the affordances and the limitations of their acoustic and digital instruments, as described in section 4.2.

There was a common agreement that the limitations of acoustic instruments were a source of inspiration and creativity. People talked about "pushing the boundaries" of the instrument and exploring its limits. Many participants said the same about digital instruments, but more commonly people were critical of the limitations of software. People felt that software limitations are due to engineering or software design, as opposed to the physical limitations of natural material like wood or strings. Interestingly, the boundaries or constraints of these two types of instruments are of a different nature. In acoustic instruments the boundaries are typically those of pitch, timbre, amplitude and polyphony. In digital instruments the limitative boundaries are more of conceptual nature, typically those of musical forms, scales, structure, etc. From this perspective, the essence of the acoustic instrument could be seen as that of a resonating physical body, whereas the digital instrument's body is theoretical, formal, normalising and general.

Most of the skilled instrumentalists experienced the limitations of their acoustic instruments positively and saw the potential – both discovered and undiscovered – of the instrument as the foundation of an interesting relationship based on depth. People usually had an "emotional" affection towards their acoustic instrument (one of our questions asked about this) and they bonded with its character. This issue changed drastically with regards to people's feelings about their digital instruments. Survey participants often expressed frustrations with the technology, irritating limitations of software environments and dissatisfaction with how hardware needs constant upgrading, maintenance, and, not surprisingly, the use of electricity. One responder talked about how the limitations of acoustic instruments change or evolve constantly according to skill levels but also state of mood, whereas the limitations of software, once it has been learned and understood, are the limitations of its design. As another participant put it: "the creative challenge [in digital instruments] is to select and refine rather than expand."

In general people felt that the main power of digital instruments is the possibility of designing them for specific needs. The design process inevitably involves composition. The fact that people talk about "composing instruments" (Bahn & Trueman 2001; Schnell & Battier 2002; Momeni 2005) reveals a clear distinction from the acoustic world where instruments tend to be more general in order to play more varied pieces. This also explains why we rarely see the continuity of a digital instrument or musical interface through time: each instrument tends to be made for a specific and not general purpose. The capacity to inscribe conceptual structures in the tool itself renders it more specific and unique for a certain musical piece or performance and less adaptive for other situations. Nevertheless, there is a continuum where instruments are on the one side unique and specific, and on the other side general and multi-purpose. Creating a digital instrument always involves decisions with regards to where to place the instrument on this continuum.

5.2.6.4 The instrument maker criticised

The survey findings indicate that people were more critical of software tools than acoustic instruments. There could be many reasons for this; one being that musical software is such a new field and naturally experimental, whereas acoustic instruments have had centuries of refinement. Another observation – supported by the survey data – is that people normally start to learn an acoustic instrument at a very young age when things are more likely to be taken for granted. Students of acoustic instruments see it as their fault if they cannot play the instrument properly, not as an imperfection in the instrument design itself. The survey participants had different relation to the digital instrument: they are more likely to criticise it and view its limitations as a weakness of design, rather than their own insufficient understanding of the system. This is reflected in the way people relate to the instruments themselves. The fact that acoustic instruments seem to have existed forever (the survey confirms that the majority of people do not have a deep historical knowledge of their instrument) makes people less likely to step back and actively criticise their instrument of choice.

Almost all the participants stated that their acoustic instruments have been built from ergonomic and aesthetic/timbral considerations and saw the evolution of their instrument as a continuous refinement of moulding it to the human body and extorting the most expressive sound out if its physical material. There is, however, evidence that orchestral instruments were developed primarily with the view to stabilise intonation and increase acoustic power or loudness (Jordà 2005: 169). In reality, the young but strong research field of digital music instruments and interface building is perhaps more consciously concerned with ergonomics and human-tool interaction than we find in the history of acoustic instrument building. Ergonomics have at least become more prominent in the way people think when building their musical tools. An agreed view was that the difficulty of building masterly interfaces in the digital realm is largely because of the complexity of the medium and the unnatural or arbitrary nature of its input and output mappings.

In section 3.2.2.1 Heidegger's phenomenological modes were introduced. Many of the participants of this survey seemed to experience a constant oscillation between modes when using digital instruments. This was often referred to as a break in *flow* (see section 3.6.1). It was suggested that the use of the computer in general and digital instruments in particular is a constant oscillation between the two modes of being ready-at-hand and present-at-hand. We forget ourselves in working with the tool for a while, but suddenly we have to open or save a file, retrieve a stored setting, switch between plug-ins or instruments, zoom into some details, open a new window, shake the mouse to find the cursor, plug in the power cable when the battery is low, kill a chat client when a “buddy” suddenly calls in the middle of a session, etc. In this respect, many of the participants saw the computer as a distracting tool that did not lend itself to deep concentration or provide a condition for the experience of flow.

5.2.6.5 Entropy and Control in Instruments

Here the enquiry was on how people relate to the non-deterministic nature of their instruments and if it makes a difference whether the instrument is acoustic or digital. On this topic there were two trends of responses. It was mostly agreed that the accidental or the entropic nature in acoustic instruments could be a source of joy and inspiration. Some people talked about playing with the tension of going out on the “slippery ice” where there was less control of the instrument, and where it seemed that the instrument had its own will or personality. Typically, people did not have the same view of digital instruments: when they go wrong or become unpredictable, it is normally because of a bug or a fault in the way they are set up or designed. However, a strand of people have enjoyed and actively searched for such “glitches” in software and hardware, of which one trace has resulted in the well known aesthetic style called “Glitch” (Goriunova & Shulgin 2008; Moradi 2009).

Nevertheless, according to the survey data, the process of exploration is a very common way of working with software, where the musician/composer sets up a system in the form of a space of sonic parameters in which he or she navigates until a desired sound or musical pattern is encountered. This style of working is quite common in generative music and in computationally creative software where artificial intelligence is used to generate the material and the final fitness function of the system tends to be the aesthetic judgement of the user.

5.2.6.6 Time and Embodiment

Predictably, the survey confirms that the longer people had played an acoustic instrument the more desire they had for a physical control, emphasising the role of the body in a musical performance. People with highly advanced motor skills, the experts of their instruments, evaluate those as important factors of music making. Playing digital instruments is considered less of an embodied practice (where motor-memory has been established) as the mapping between gesture and sound can be changed so easily by changing a variable, a setting, a patch or a program. Performing the digital instrument is seen as a cognitive activity, an example of epistemically mediated embodiment, where a set of dynamic parameters becomes part of the embodied function. Some responders noted that working with digital instruments or software systems had forced them to re-evaluate the way they understand and play their acoustic instrument. Of course, the contrary has to be true as well.

It should be noted here that most of the people that answered the survey were both acoustic and digital instrumentalists and embraced the qualities of both worlds. It seems that people subscribe positively to the qualities of each of the two instrumental modalities – acoustic and digital – and do not try to impose working patterns that work in one type of instruments onto the other. In general people seem to approach instruments on their own merits and choose to spend time with a selected instrument only if it gives them some challenge, excitement and the prospect of nuanced expression. It is in this context that we witness a drastic divergence in the two types of instruments: as opposed to the willingness to spend years on practicing an acoustic instrument, people demand that the digital instrument has to be easily learned, understood and mastered in a matter of hours for it to be acceptable. This fact emphasises the role of tradition and history in relation to acoustic instruments and the role of progress, inventiveness and novelty with regards to instruments founded on the new digital technologies.

5.2.7 Interesting Comments

There were some comments that are worth printing here due to their direct and clear presentation:

> *"I don't feel like I'm playing a digital instrument so much as operating it."*
> *"Eternal upgrading makes me nervous."*
> *"full control is not interesting for experimentation. but lack of control is not useful for composition."*
> *"Can a software 'instrument' really be considered an instrument, or is it something radically different?"*
> *"The relationship with my first instrument (guitar) is a love/hate one, as over the years I developed a lot of musical habits that are hard to get rid of ;-)"*
> *"j'entretiens un certain rapport avec mes machines. Impossible pour moi de penser à revendre une machine."*
> *"I think acoustic instruments tend to be more right brain or spatial, and digital instruments tend to be more left brain and linguistic."*

5.2.8 Discussion

There were many surprising and interesting findings that came out of this survey. First of all, we were intrigued by the high mean age of the survey participants. We wondered if the reason for the high average age could be the nature of the questions, especially considering the questions regarding embodiment. Perhaps such questions are not as relevant to younger people who have been brought up with the computer and are less alienated by the different modes of physical vs. virtual interaction? One explanation might be that the mean age of the survey participants is reflecting that of the members of the mailing lists we posted the survey to, but that is questionable. More likely is that the questions resonated better with more mature people. It is illustrative that the majority of people answering the survey were involved with academia or had an academic or conservatory education. This helps to explain the high mean age but also the high level of analysis that most people had applied to their tools. We noted that the time spent playing an instrument enforces the musician's focus on embodiment and as such the questions of this survey might have connected better with the older musicians.

An important point to raise here is that whereas the survey focused on people's perception of the differences of acoustic and digital instruments, the fact is that most people are content with working with both instrument types and confidently subscribe to the different qualities of each. Many of the participants use the computer in combination with acoustic instruments, especially for things that the computer excels at such as musical analysis, adaptive effects, building hyper-instruments and the use of artificial intelligence.

A clear difference between the acoustic and digital instruments is the polarity between the instrument maker and the musician in acoustic instruments. This is further forked by the distinction of the composer and the performer. In the field of digital instruments, *designing* an instrument often overlaps with the musical *composition* itself (or at least designing its conditions) which, in turn, is manifested in the *performance* of the composition, most typically, by the composer/instrument builder herself. This relates to a specific continuum of expression in digital instrument design that we do not see as clearly in acoustic instruments: a digital musical instrument can be highly musical, a signature of the composer herself, rendering it useless for other performers. But it could also be generic, universal, and open for musical inscriptions, thus more usable to everybody.

Another interesting trait noticeable in the survey was the question of open source software. People professed that the use of open source tools was more community building, collaborative, educative and correct than working with commercial software. Many stated that open source should be used in education precisely due to these facts and because of the continuity that open formats have. Furthermore, many people were using Linux (or expressing desire to do so) because they felt that they had more control over things and are less directed by some commercial company's ideas of how to set up the working environment or compose/perform music. The questions of open protocols and standards, of legacy in software, of collaborative design and freedom to change the system were all important issues here.

5.3 The Phylogenesis of Musical Instruments

If the inventions and innovations of musical technologies typically happen through the mechanisms of cultural adaption or socio-technical symbiosis in the manner discussed in the previous section; how do musical instruments evolve through time? And which factors define that evolution? In order to explore this question we can now contextualise Stiegler's theories of epiphylogenesis and technical tendencies together with the actor-network theory of punctualisations and bio-matter dialectics in relation to phylogenetics, or the study of evolutionary relations between organisms. Phylogenetics are now increasingly taken into use in the analysis of cultural phenomena, especially within the material culture branch of anthropology (Tehrani & Collard 2002; Temkin 2004). This methodology is able to yield results that differ from the traditional historical narration of the instrument's evolution, and as such is more closely related to the cultural genealogy than traditional history.

5.3.1 The Phylogenetics of the Cornet

Niles Eldredge, an influential biologist (the co-author of the theory of punctuated equilibria), is an ardent collector of cornets (a small version of the trumpet). In 2002, curious about the wide variety in the function and shape of cornets, Eldredge started arranging them in phylogenetic or taxonomic relationships of shape, style, manufacturer and date. Eldredge's theory of punctuated equilibria suggests that evolution does not happen through gradual transformation of whole biological lineages, but rather through quick morphological ruptures and discontinuities[98] followed by longer periods of stability. Eldredge sees similar evolution in the phylogenesis of musical instruments: "I knew that there were periods of stasis in cornet development, and also periods of radiation and innovation" (Walker 2003: 38).

By running the design of selected cornets (seventeen characteristics or anatomical elements were identified, such as the bell-position) through a phylogenetic program, Eldredge retrieved results clearly indicating that musical instruments (and in general any designed artefacts) do not follow the same evolutionary trends as biology. They are subjected to different and more chaotic laws where the maker of the instrument can copy inventions from other species of instruments, go back in time and implement features that have been lost for generations of instruments, etc. This can be defined as cultural evolution (as opposed to biological evolution) and what characterises it is the possible co-opting of innovations at a whim. The phylogenetic evolutionary lines thus differ in biological and cultural evolution. In living systems, the evolutionary lines are typically V-shaped, shaped like branches that exhibit increasing diversity, but in cultural evolution they are typically straight, the result of an abrupt morphological change. Figure 5:3 shows this difference.

[98] This is called "punctuational patterns" and is reminiscent to the ANT idea of punctuation in the evolution of technical artefacts when they become blackboxed.

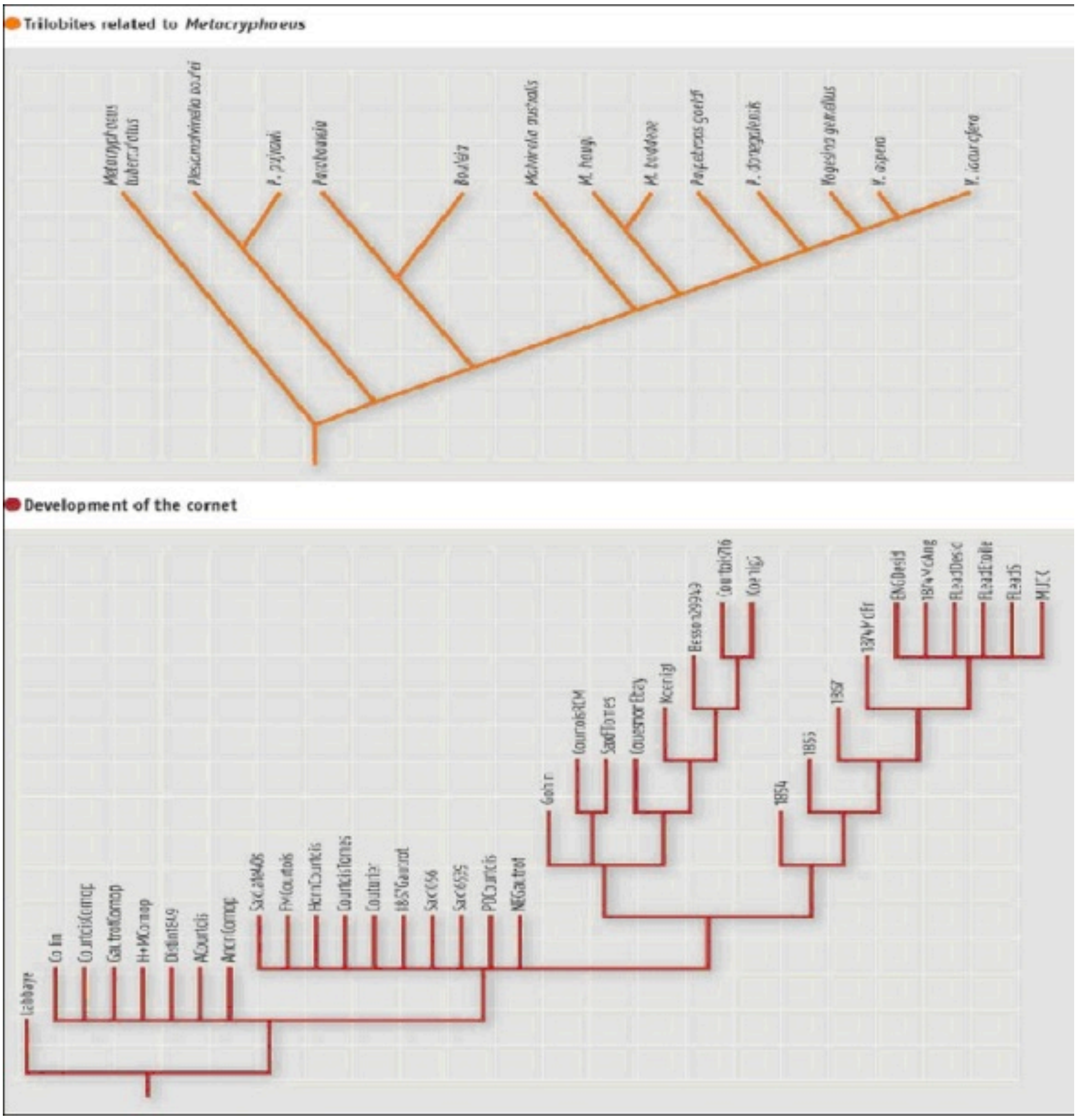

Figure 5:3. A representation of the difference in biological and cultural evolution. (Image by courtesy of Greg Miller, New Scientist 07/04)

5.3.2 A Revisit of Stiegler's Epiphylogenesis

Biological science, such as the phylogenetic theories of Eldredge, lacks the vocabulary or the discourse to analyse cultural artefacts as evolutionary phenomena.[99] Stiegler's ideas of ephiphylogenetics are relevant here, where the technological potential or "tendency" precedes the implemented technologies. For Stiegler, "the logic of invention is not that of the inventor. One must speak of techno-logic, of a logic literally driving technics itself" (Stiegler 1998: 35). He draws this thought from Bertrand Gille's "loose determinism," i.e., the idea that although one cannot anticipate technological evolution a priori, it is not a random innovation (as it often seems). Technological progress is driven by a specific logic innate to technology itself: "*The*

[99] Although the biologically inspired field of memetics (Dawkins 1976) has been suggested as one possible approach to the study of cultural transmissions.

system's dynamic offers the possibility of invention, and this is what is essential to the concept of technical system: the choice of possibilities in which invention consists is made in a particular space and particular time according to the play of these constraints, which are submitted in turn to external ones" (Stiegler 1998: 35). The external constraints here being those of economy, science, society and culture. Our culture is becoming ever more "techno-logical" in the sense that we are quicker to adapt to changing technological situations and more accustomed to techne's particular logic. The Frankfurt school calls this "rationalisation" referring to the heightened sense of functionality installed into all domains of life. As a system external to ourselves with its own dynamic, the technical structure is virtually impregnated with potential, and it only takes a human to be in the right place at the right time to instantiate a new technological token.

> Everything happens as though the technical milieu were constantly undergoing the effect of all technical possibilities, that is, as if the entire determining tendency directed onto itself the totality of its excitations at every moment. (Leroi-Gourhan, quoted in Stiegler 1998: 61)

Here we get a clearer picture of what Stiegler means with "organized inorganic matter" (e.g., ibid: 71). It is a structure of matter that is dead and would not evolve without human in(ter)vention. Nevertheless, it is a structure that transductively constitutes human nature. Technology is defined as external artefacts onto which the human delegates cognition, practically becoming an exteriorisation of its memory (what Stiegler later (2003b) calls mnemotechnics). Furthermore, the technical object is more than the sum of the scientific principles that it implements. Here Stiegler quotes Simondon:

> [a] certain mode of functioning and a compatibility that actually exists and that was fashioned *before being planned*: this compatibility was not contained in each of the separate scientific principles that served for the object's construction, it was discovered empirically. (ibid: 78)

Yet again we are faced with the material epistemology [studied in chapter 4] of epistemic tools. Technological objects prove to have a certain logic, instantiated in their multiplicity and variations, and they can exhibit intelligence, social structuring, understanding of natural laws, and alternative usage, that was never foreseeable in the process of its design and innovation.

In our context, looking at instrument building, Stiegler's claims that "[t]echnical evolution stems completely from its own technical object" (ibid: 66) and that "[t]he human is no longer the *intentional actor* in this dynamic. It is its *operator*" (ibid) illustrate how the human role within the technical ensemble changes with its increasing complexity. As Leroi-Gourhan's quote above indicates, the structure of technology increases exponentially in complexity over time. Add one new element and the connectivity of nodes explode in number. The tradition of instrument building that Eldredge analyses, the acoustic tradition where there is a history, tradition and a lineage of evolution, is therefore a much simpler domain for genealogical or phylogenetic analysis than the domain of digital instruments. The attempt to analyse digital musical instruments through phylogenetic criteria is practically impossible due to the invisibility of the underlying and blackboxed mechanisms (code libraries, protected protocols, obscure standards and closed hardware). We therefore have to acknowledge this complexity of the digital stratum and the impenetrability into the history of the digital instrument and rather look at its nature through taxonomical orders other than the phylogenetic one [see section 5.5].

5.3.3 The External and Non-genetic Evolution of the Cultural Artefact

Technical evolution not only works from technical tendency, the collective logic of technical materiality, but also differs from biological evolution in that it can have prototypes as ancestors that result in many divergent *and* convergent branches. The ancestor is never to be found, evolution does not necessarily branch dichotomously as there can be multiple lineages originating from the same node.

> Differences run deeper. In the historical development of man-made artefacts, a single technological function can be solved in many ways differing in underlying physical principles. The temporal sequence of alternative designs in material cultural systems, frequently taken as evidence for homologous similarity... is often not equivalent to the transformation series made up of ancestral and derived states, the basic underlying assumption built into all biological phylogenetic analytic rubrics. Rather, in material cultural systems, the historical sequence of design innovations consists of a set of alternative solutions to the same functional problem..., in some cases prompted by competition or patent protection for competing designs. This challenges the application of biological notion of homology, which, in turn, lies at the core of phylogenetic inference. In material culture, the basis of comparison is in most cases limited to features that perform the same function rather than sharing function and derived form by common ancestry. Thus, entities in question in the biological versus the cultural domain differ ontologically and epistemologically. (Eldredge & Temkin 2007: 150)

Eldredge and Temkin elegantly point out the differences in the evolution of biological and technological systems. The technological system's evolution may appear similar to biological evolution in places where traditional transmission is strong but intercultural exchange is weak. However, in the global, post-industrial and, in particular, the digital world, this is not the case anymore – all cultures now tend to share access to the same technological tendencies. Therefore, the evolution of cultural systems and technological artefacts is more complex and discontinuous than biological evolution with more branches, loops back in time and multiple siblings.[100] Whereas biological systems have natural boundaries (interspecies breeding is very rare), cultural systems are rife with lateral exchange amongst designs, yet they are also characterised by a surprising (from an evolutionary point of view) resistance to change that derives from cultural protection and the value of tradition.

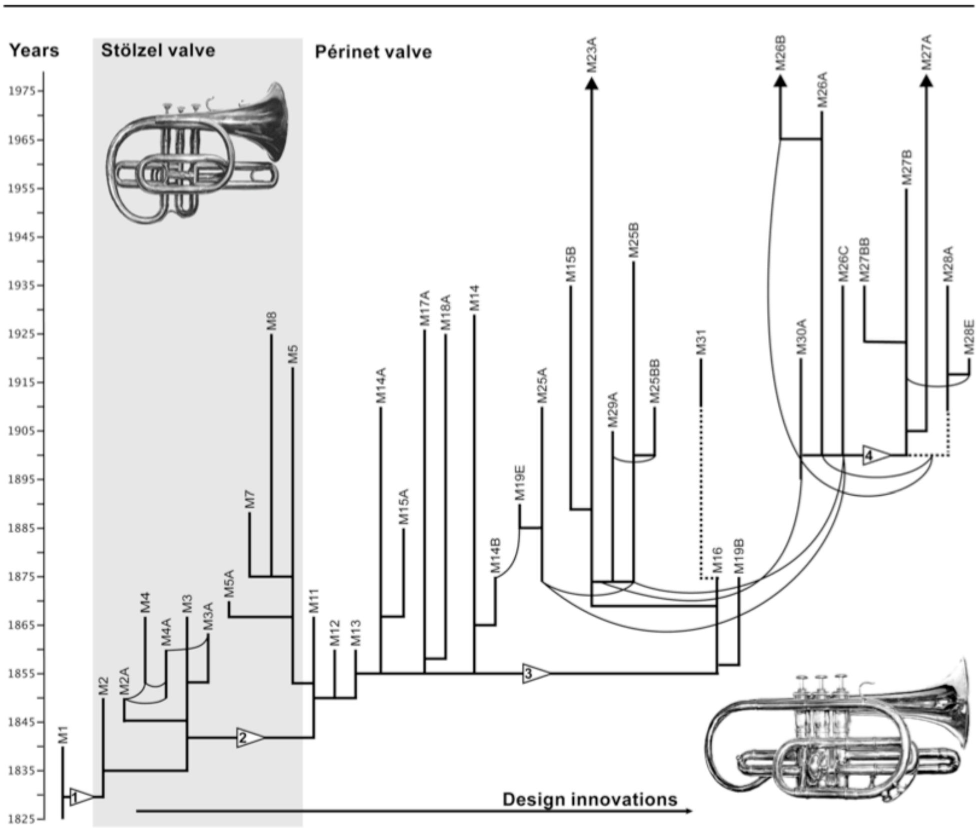

Figure 5:4. An evolutionary model of the cornet. Image from Eldredge and Temkin (2007)

In cultural artefacts the information resides in the object itself, but also in plans, sketches, patents, drawings, schematic diagrams, photographs, and manuals. It takes a human to transfer that information though a lineage of objects and implement an alternative design. The desire for change in cultural artefacts is often dependent upon survivability on the marketplace, but also alternative engineering solutions to already patented solutions. Mutation in cultural artefacts is therefore typically more varied, which means that there can be a total breach in morphology of a thing that performs the same function. This is equivalent to saying that there is not necessarily a genetic (as in Mendelian genetics) continuity in the evolution of the cultural or technological artefact. A good example is how the transistor replaced the vacuum-tube (which was not necessarily an improvement in all cases) in amplifiers. There is no sense of continuity in design

[100] It is perhaps interesting to note here that one of the goals with Eldredge's research is to show how biological evolution is not the result of "intelligent design" (as some religious creationists will have it). Cultural artefacts are products of intelligent design and the nature of their evolution differs drastically from biological evolution.

between the transistor and the tube but they perform principally the same function. However, as opposed to biological species and early technological inventions, the shift to (or naturalisation of) the transistor was simple, as it replaced an older and less optional technology.

5.3.4 Conclusion

Although both acoustic and digital instruments evolve through a discontinuous and eruptive processes, their technical tendencies, or potential, differ. Where the acoustic instrument maker can easily refer back to tradition, copy structures from other instruments or transform the instrument in various ways (all creative processes that can be explained by Boden's (1991) theory of creativity) the digital instrument maker is to be found in a context defined by the multiplicity of actors and black boxes that prove practically impossible to get a full understanding of. The digital instrument maker is conditioned by operating systems, programming languages, hardware interfaces, communication protocols and other constraining technical elements that cannot be easily altered. Interlineage transfers between instruments are thus profoundly different in digital instruments due to the rigidity and complexity of their materials: they have different evolutionary dynamics (of much higher and more disruptive discontinuities) than acoustic instruments.

As Kartomi (1990) illustrates, there are no limitations to the number or types of analytical taxonomies that can be formulated for musical instruments. The phylogenetic analysis of musical instruments provides us with one alternative history of the instrument. We can build a taxonomy of musical instruments from it, but that is a taxonomy based on genetics and not their various functions, such as the sonic or ergonomic qualities. One problem with phylogenetic analysis of digital technologies (or all complex machinery) is that their underlying structure is not visible, and typically so blackboxed in strata down to machine levels and machine histories, that it becomes an impossible analytical task. Hence, the history of the machine or the digital instrument will not be written from phylogenetic analysis. A more productive account is the genealogical one, as I will now attempt in the next section. However, functional analysis, sonic characteristics and ergonomic families should not be excluded either and this will be covered in section 5.5, where we engage with the construction of artificial taxonomies.

5.4 The Ontogenesis of Musical Instruments

Consider things, and you will have humans… consider humans and you are by that very act interested in things. (Latour 2000: 20)

Historical accidents, marketing decisions, the inventor's abilities for innovation (getting the product out – creating the demand) are all factors that come into play when we look at the history of digital musical systems. An interesting, however forgotten and unacknowledged, point of convergence in the history of computer music software happened in 1988 and relates to the Steinberg music software company. Steinberg was then a small software house in Hamburg that had a musical software product (a tracker) called Pro-24. The Atari computer had just come out (with a 16-bit operating system) which lent itself perfectly for the design of musical sequencers as it had an inbuilt MIDI port. The company was growing fast at the time and was developing a multi-track sequencer whose first release was called Cubit, later to become Cubase. That year, Steinberg was visited by a music researcher and programmer, Andy Hunt, who had developed a highly original, flexible and non-imitative platform for musical improvisation and composition called MidiGrid. MidiGrid did not imitate hardware or acoustic instruments, but provided an innovative way to work with data coming from MIDI interfaces (such as keyboards, drumpads, saxophones) and arrange it in a *realtime performance*. Steinberg were intrigued by the software but for some reasons never bought the idea nor showed an interest in marketing it (Hunt 2008, personal communication). Instead Steinberg focused on their multi-track software achieving great commercial success.

Considering the history of digital music systems as primarily editors or compositional devices and not performative realtime tools, one could now start to ask "what if" questions. What if Steinberg had decided to market MidiGrid? Would we have seen another branch of digital musical software (and instruments) that would focus on realtime performance, improvisation, and algorithmic composition? Could there have been a fork in music software where one branch would follow the multi-track simulation model, whereas the other branch would focus on creative exploration of the computer as a unique tool with specific affordances for musical expression? [see section 7.3.2 on expressive and production tools] This is a field that has become very strong today (as exemplified by NIME) but one cannot help but wonder how much better a kick-start it would have gotten, if it had been introduced earlier into the history of digital music systems.

Now, instead of pondering such "what if" questions, it will be more fruitful to actually look at instruments whose innovation *was* successful in terms of production and market success. This section will explore three different instruments whose origin and development are well documented: the saxophone (acoustic), the Moog synthesizer (electronic) and the reacTable (digital).

5.4.1 A Note on Method

If materiality and existing technologies provide the structure in which new inventions can happen and if the human is not the inventor of technology but its operator, how are musical instruments conceived? As Mayer (1999) shows, the history of modern media can be traced back in time through various threads,[101] for example to Leibniz' (1646-1716) *characteristica universalis*, a universal symbolic language used in formal logic, the mathematics of reasoning. We could go back to Boole's (1815-64) algebra that established the mathematics of binary logic that is requisite in the operation of computers. Babbage's Analytical and Difference Engines are important ancestors as well. However, history is not linear, it is not a logical progression through time, but rather an accidental, rhizomatic and non-teleological connection of events that can only be *narrated* as a historical overview.

Therefore, instead of a grand historical narrative, our enquiry will be partly genealogical (in the Nietzschean sense) which is an approach developed into a sophisticated methodological tool by Michel Foucault (1977b) who defines genealogy thus:

> Genealogy is gray, meticulous, and patiently documentary. It operates on a field of entangled and confused parchments, on documents that have been scratched over and recopied many times…. [it] requires patience and a knowledge of details and it depends on a vast accumulation of source material…. In short, it demands relentless erudition. Genealogy does not oppose itself to history as the lofty and profound gaze of the philosopher might compare to the molelike perspective of the scholar; on the contrary, it rejects the metahistorical deployment of ideal significations and indefinite teleologies. It opposes itself to the search for "origins." (Foucault 1977b: 139-140)

Having abandoned the search for the grand narrative of history, genealogy finds branches, accidents, hidden actors, and other influences that accompany every beginning and are typically overlooked in historical accounts. The genealogist is not looking for "total" knowledge, nor even pretending to have full knowledge or a comprehensive perspective.[102] History is not presented here in terms of universals, avoiding the exceptional and residing at the lowest common denominator of connected events. Genealogy introduces discontinuity into the reading, a closeness as opposed to distance, and a focus on the nitty-gritty of events from an alienated perspective, not from identification. From such a perspective, our reading will naturally involve an actor-network theoretical analysis where we explore which actors and which networks are

[101] A history that Maggie Boden (2006) has also given good account of in the field of cognitive science.
[102] An exemplary genealogical approach is taken by De Landa (1997) in his *A Thousand Years of Non-linear History* where he traces the histories of geology, biology and linguistics without the focus on the human as the protagonist in the historical proceedings.

enrolled into play in the creation of our musical instruments. Following the genealogical vision we will not be looking for the "origins" of musical instruments as a pivotal point in time, but rather inspect some of the traces that constitute their concretisation as blackboxes, as instruments that became pointilised in the social context, thus as historical agents.[103]

5.4.2 The Saxophone

The story of Adolphe Sax (1814-1894) is a story of success and failure. Adolphe Sax was the son of Charles Sax, who was a successful clarinet and brass instrument maker in Brussels, an appointed Instrument Maker to the Court of the Netherlands. Adolphe worked in his father's workshop from an early age, and contributed to various improvements of the bass clarinet, a design that has been the basis of all bass clarinets since. Adolphe Sax was also a skilled clarinettist. It was his interest in overcoming the clarinet's shortcomings that set him on a path to invent the saxophone. He wanted an instrument that would go up an octave when overblown and not by a 12th as did the clarinet. He was interested in the use of a single-reed mouthpiece on an instrument that would have a thick, strong and singing sound. The harmonic register of the saxophone centres around 2000 Hz which is the register of the human voice.[104] Sax presented the saxophone at the Brussels Exhibition in 1841 and shortly afterwards he was visited by an aide of the French king, Louis-Philippe, who was interested in the saxophone as an instrument for military bands. Sax subsequently moved to Paris.

In Paris, Sax was introduced to the composer Hector Berlioz, who spoke highly of the expressivity of the saxophone and Sax's inventive skills. A meeting with other French composers resulted in a funding to establish the Adolphe Sax Musical Instrument Factory, at No. 10 Rue Saint Georges. The attention Sax was receiving engendered enmity from the established instrument makers of Paris who harassed him in various ways, prompting Berlioz to write "The persecutions he suffers are worthy of the Middle Ages…. They lure away his workmen, steal his designs, accuse him of insanity, and bring legal proceedings against him" (Horwood 1983: 47). Even musicians, who were consultants to established instrument makers, would refuse to play in an orchestra where the saxophone would be used (Liley 1998: 5). Sax had almost established his instrument in 1845 when the French government decided to reform the military bands, and the saxophone won a public contest of what instrument should be used in these bands. The government's choice of the saxophone meant that Sax had a near monopoly

---

[103] In their fine book on synthesizers, *Analog Days,* Pinch and Trocco (2002) show their awareness of the subjective and situated nature of historical accounts by resolving to the terminology of analogue synthesis: "We have filtered stories to bring out certain themes and have muted others. We have shaped our account, giving it narrative structure…We have, on occasions, fed the stories back to the participants and hence produced a new version of events. Sometimes when stories do not match up, rather than get rid of inconsistencies, we have allowed discordances to remain… There are silences, and noise is everywhere. We have finetuned and patched the best we can" (ibid: 11).

[104] In jazz culture, the Saxophone is often called the "singing instrument".

of the military band market, a situation that threatened the bankruptcy of other instrument makers. In order to protect their interests, the other instrument makers founded *L'Association générale des ouvriers en instruments de musique*. The main objection they had to Sax was that he had patented the saxophone, which made it impossible for other manufacturers to make similar instruments. None of this affected Adolphe Sax too much, and he had bright prospects until 1848, when the Revolution sent King Louis-Philippe into exile. Sax's close ties to the former king was now a liability. The subsequent government presented a new bill influenced by *L'Association* for the reforms of military bands and Sax quickly became bankrupt.

Eventually, the future became rosier for Sax and in 1854 he was appointed Musical Instrument Maker to the Household Troops of Emperor Napoleon III. Napoleon helped him to settle debts with his creditors and another factory of instruments was set up in Rue Saint Georges, this time at No 50. Sax's next blow was cancer in the lip, and more lawsuits, together with the French surrender at Sedan in 1870, which resulted in an economic recession. The saxophone was taken out of the curriculum at the Conservatory, and Sax became bankrupt again. Musicians' support for Sax resulted in him getting a modest pension on which he lived until the end of his life.

Figure 5:5. Saxophones made by Adolphe Sax

The saxophone has various ancestors: Liley (1998) mentions an Argentinian instrument made of cow's horn; the alto fagotto is another instrument made in England that precedes the saxophone (Rendall 1932: 107); and *L'Association* in Paris tried to prove that an instrument made by Desfontelles in 1807 was the precursor of the saxophone, as it had curved mouthpiece and an upward turned bell. However it did not overblow in an octave like the saxophone, and was therefore effectively an early bass clarinet (ibid.). Other suggested predecessors have been the bathyphone, the Hungarian tarogato and the tenoroon. It is not clear how much Sax was influenced by these instruments. What *is* clear is that Sax was a skilled and knowledgeable instrument maker of the finest sort, working with the properties of physical material and the history of musical instruments, thus able to access a reservoir of technical solutions to posed problems. What is also well understood are the changes to the saxophone since Sax's invention. It is not the same instrument as that invented in the 1830s. The historical question of the "origin" of the saxophone thus branches out into further genealogical and phylogenetic investigations of older musical instruments on the one hand, and later refinements and changes in design on the other, but we will leave our investigation here.

From the perspective of Gille's increase in the speed of technological advances, we can note here that it took the clarinet a full century after Denner's improvements until Mozart used it in his Clarinet Concerto. The valved horn was initially resisted as well, and was only accepted eighty years after its invention at the Paris Conservatoire. The saxophone had become established after a lifetime of fifty years and to be revitalised with its adoption by jazz musicians in the 1920s. However, there is still a problem related to its youth. Although there are various orchestral works that include the saxophone, there are not many written dedicatedly for the saxophone, rendering its stature within the orchestral culture undefined.[105]

As most inventors sooner or later realise, the innovation of the invention (of establishing it as part of social practice) is much harder than the work behind the invention itself. Many inventions disappear or sway for other inventions (often of lesser quality) whose innovative success is greater. Sax realised this and was therefore eager to teach at the Paris Conservatory, even offering to teach without salary when his saxophone class was discontinued after the Revolution of 1848. Sax also set up a publishing house, which he operated from the late 1850s until his bankruptcy in the late 1870s. Sax published over 200 works for the saxophone, many of whom were written by famous contemporary composers.

5.4.3 The MiniMoog Synthesizer

Bob Moog's father was one of the first radio amateurs in America. He had a workshop in the basement of the family house, equipped with all that was needed in the exploration of electronics. Just as Sax, Moog spent his youth in his father's workshop developing radios, one note organs, playing with oscillators and eventually building a replica of the Theremin, the world's first commercial electronic instrument.[106] He studied electrical engineering at Columbia University, starting the same year (1957) as the RCA Mark II synthesizer was installed in the Columbia-Princeton Electronic Music Center. Moog never saw the RCA synthesizer though, as it was only used by the "high-brow" musical culture epitomised by the centre's director Vladimir Ussachevsky and the composer Milton Babbitt. In 1961, Moog began selling Theremin kits for $50 per kit and two years later he rented a store, hired people, and set up R.A.

---

[105] Much of that can be explained by its loudness that would drown the other woodwind instruments. Interestingly, from the taxonomical perspective, the saxophone is considered a woodwind instrument by classical players, but belonging to the brass family by jazz players.

[106] The Theremin had a rather long innovation time – in the 1920s it was promoted and became relatively well known as a home instrument, in 1950s it was used in sci-fi films and then in the 1960s it became a favoured sound in pop music. The Theremin has established its place in the history of musical instruments. It is widely used today, albeit considered hard to master due to the lack of tactile contact with the interface.

Moog Co. in Trumansburg, NY State, where he was based at the time working on a PhD at Cornell University.

A meeting with the New York composer Herb Deutsch the same year inspired the design of voltage-controlled pitch (the oscillator would have a pitch-range depending on the voltage fed into it). This naturally led to the idea that the output of one oscillator (low frequency oscillator, or LFO) could become pitch or amplitude input into the next one (Chadabe 1997: 141). Moog and Deutsch collaborated on the design, where the musical problems would be addressed/questioned by Deutsch and the technical problems solved by Moog. For example, Deutsch's wish to be able to "articulate the instrument" (Pinch & Trocco 2002: 27) as musical events was immediately addressed by Moog with the invention of the envelope generator.

Next summer, Moog and Deutsch visited the University of Toronto Electronic Music Studio headed by Myron Schaeffer. The Canadians liked what they saw and one of them, Gustav Chiamaga, "gave Moog an important idea" (ibid.). He suggested the filter module, which could naturally also be controlled with an LFO.[107] Moog was thus arriving at the idea of the modular synthesizer as a small portable studio built on units (oscillators, filters, envelope generators and amplifiers) that could be plugged into each other. The oscillators generated many waveforms (such as sine, saw, pulse and triangle) the latter three with rich set of overtones that could be filtered according to the sound designers/musician's wishes. This type of synthesis appropriately got the name of "subtractive synthesis."

Moog's designs were not particularly "original" as there were instruments and synthesizers already built, and written about, such as the work of German inventor Harald Bode. As already mentioned, important design features were typically responses to users' requests. From Wendy Carlos, Moog got the idea of implementing touch-sensitive keyboards, portamento control and filter banks. Moog's unique stance was in engineering excellence, combining the right elements together and being sufficiently well situated with respect to marketing them.

> In those years, I can't ever remember looking into the future, trying to understand what would be best for musicians. Ideas came from all over the place and what we wound up selling was the sum total of all those ideas… (Moog in Chadabe 1997: 142)

Putting a keyboard on the synthesizer was a decision Moog took after encouragement from his collaborators. In the early days, there was no reason to choose the keyboard as the principal interface as the synthesizer was not naturally an equally tempered, half tone and limited-to-a-small-number-of-octaves type of instrument. In fact, Moog often said that the keyboard was put there initially mainly for photographers – the instrument needed a musical element to it. To the layman, a keyboard instantly connoted music, whereas the dials, buttons and cables did not. The

---

[107] Such a filter controlled by an LFO is effectively an automatic wahwah pedal. Only later was Moog to design the ladder filter which became one of the defining characteristics of the Moog synthesizers.

seemingly innocent decision of implementing the keyboard as the interface was important, and it has coloured the sector of musical instruments and our music ever since. Another synthesizer maker, Don Buchla (based in San Francisco) had different ideas about the keyboard:

> I saw no reason to borrow from a keyboard, which is a device invented to throw hammers at strings… A keyboard is dictatorial. When you've got a black and white keyboard there it's hard to play anything but keyboard music. And when's there not a black and white keyboard you get into the knobs and the wires and the interconnections and the timbres, and you get involved in many other aspects of the music, and it's a far more experimental way. (Buchla in Pinch & Trocco 2002: 44)

Moog's entry into the synthesizer business occurred in 1965 when Alwin Nikolais visited the Moog studios and bought various modules. Soon after, the orders started to flow in, a particularly large one coming from Vladimir Ussachevsky at the Columbia-Princeton Studios. Musicians loved the sound of the analogue synthesizer and composers like Lejaren Hiller, Wendy Carlos (who wrote the music for Kubrick's *Clockwork Orange*) and Keith Emerson (from Emerson, Lake and Palmer) started to use the Moog synthesizer extensively in their work. The psychedelic 60s had found the ideal sound for their musical journeys, the Moog synthesizer. It was entering public consciousness, appearing on stage with famous musicians (such as The Doors, George Harrison, and Emerson, Lake and Palmer), and articles about it were written in *Newsweek, Times,* and *The New York Times.* Interestingly, Moog was not preoccupied or even conscious about the saleability of what he had created; he was always more of an inventor than an innovator.

Figure 5:6. The MiniMoog from 1971

In the early 1970s, Bob Moog had an established company that sold synthesizers – the Moog Modular – to labs and recording studios. It became particularly popular in studios resulting in members of the musician's union (AMF) (mostly session musicians), viewing the synthesizer as a threat to their livelihood: the synthesizer seemed to be able to perform many things that they could not do and easily simulated their playing; a worry that later proved to be unfounded. However, although some musicians (like Keith Emerson) were brave enough to use the Moog Modular on stage, Moog never thought it would be used in live performances due to the size of the synthesizers and their tendency to become detuned. At this time, it was seen as a studio tool only, not as an instrument for live performance.

A young engineer, Bill Hemsath, had been working in Moog's ever growing factory and in his lunchtimes he would work on a project where he compiled old junk from the attic where his office was, and tried to make a compact synthesizer where all patch-chords were hidden away behind the simple user-interface (thus called the "integrated synthesizer"). Due to the rejection of Moog himself, Hemsath's invention (The Min A model) was never put into production. However, as the company was now entering a period of financial problems, the engineering team decided on the production of the MiniMoog (Model D) at a point when Bob was out on a speaking tour. They took this decision against his will as they knew he would not agree. They also knew the company was becoming bankrupt and they would "all be out of a job anyway" (Pinch 2003b: 251). Moog was not happy when he came back from his trip, but eventually agreed on producing the MiniMoog, which became a great success.

A friend of Moog's, David Van Koevering, got a strong vision for the future of the synthesizer when he saw it for the first time: "I saw something… the power of the sound, the sonic energy, and I believed that it could become common, and I imagined it as powerful as the electric guitar to the first guys that ever played that thing… And I argued with Bob that it is a performance instrument" (Pinch & Trocco 2002: 238). It was a personal initiative of Van Koevering (who had been performing on the MiniMoog at Taco Bell restaurants, in order to create a vibe for the instrument) to embark on a sales tour around music stores all over the States.[108] But this was not an easy task as the market was highly resistant to the new instruments, or as Van Koevering phrases it "I've been thrown out of more music stores than any man alive, because before you would condition the market there was no market, and I had to invent the market" (ibid: 257). In the end, his trick was to go directly to the musicians who would then go to the stores and ask about synthesizers. Immediately the demand would increase and the store managers would eventually believe in the business potential of this new musical instrument. This is a sales technique well known in America when new products are introduced: the first step of the project becomes that of creating the market! Many would say that Van Koevering single-handedly invented the market for sound synthesizers.

5.4.4 The reacTable: Physical Objects on a Virtual Surface

The reacTable is a good example of an instrument-in-the-making. In the context of this chapter, writing about the reacTable is ideal, since its invention is fulfilled to a large extent but its innovation is still happening. It is a mature project, impressive and popular, and its inventors

[108] Something that never occurred to Moog himself: "I don't think I or anyone else in the company, went into a music store before… March of 1971" (ibid: 252).

have recently made a start-up company to market and sell the instrument. The project has a commercial potential, but it might just as well vaporise due to the quirkiness of the market.

The project was developed by a core team of Sergi Jordà, Martin Kaltenbrunner, Günter Geiger and Marcos Alonso (but with others involved such as AudioMulch's Ross Bencina and various graduate students), at the Pompeu Fabra University in Barcelona. It was a continuation of ideas that Jordà had been working on with his FMOL project – an online multi-player improvisation instrument that would present the instrument graphically and visualise the sounds played. In a 2003 paper, Jordà defines the aims of the reacTable:

> The reacTable* aims at the creation of a state-of-the-art interactive music instrument, which should be collaborative (off and on-line), intuitive (zero manual, zero instructions), sonically challenging and interesting, learnable, suitable for complete novices (in installations), suitable for advanced electronic musicians (in concerts) and totally controllable. The reacTable* uses no mouse, no keyboard, no cables, no wearables. It allows a flexible number of users, and these will be able to enter or leave the instrument-installation without previous announcements. The technology involved should be, in one word, completely transparent. (Jordà 2003: 4)

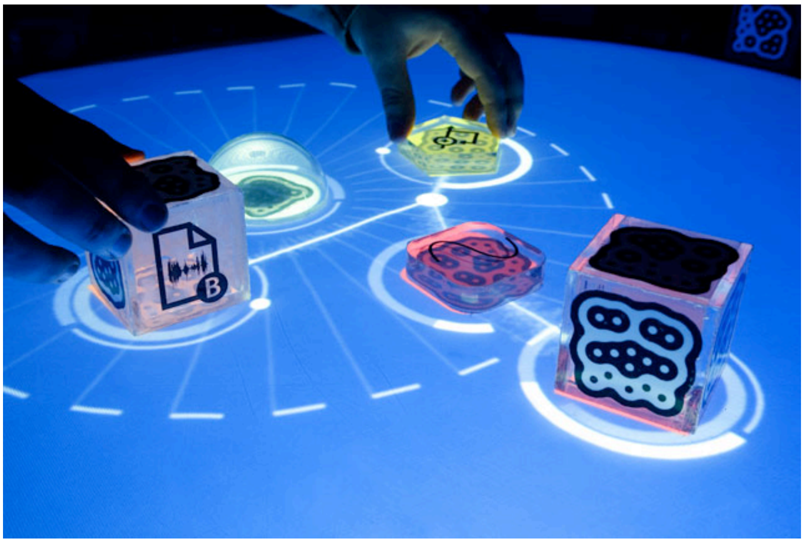

Figure 5:7. The reacTable. A camera below the screen detcts the cube's shape and returns objects for sound generation

The reacTable is a semi-transparent surface onto which the screen of a simulated synthesizer is projected from below. The user places cubes with special patterns on them onto the table and a camera reads the pattern and creates the right oscillator or filter in the synth. Through "dynamic patching," the user is able to route signals into each other, such as plugging a low frequency oscillator (LFO) into the frequency input of another oscillator. That oscillator might go through a filter before it is routed to the digital to analogue converter (represented by a white dot in the middle of the table). The user is therefore both designing and playing the instrument at the same time. The instrument allows for a certain degree of mastery through practice and in fact it resembles the internal workings of the modular synthesizer so knowledge in one field can easily be translated to the other. The sound-engines have been created in Pure Data and SuperCollider, and as both of these environments support OSC, it is relatively simple to set up a multi-table performance, where players can interact over the internet.

The interesting features of this instrument is the way it visually represents the internals of a synthesizer. Apart from the obvious play and educational value of this interface (where people frequently spread around the table and play together on the instrument), it provides us with an interesting hybrid of the physical and the virtual. Physical objects are used to represent the oscillators, effects and filters. This is done from the desire of bridging what Ishii and Ullmer call "the great divide between bits and atoms" (Ishii & Ullmer, 1997). The reacTable combines in a powerful way ideas from various directions into a unique user-interface. From the Moog modular it gets the idea of modular patching; from various TUI (Tangible User Interfaces) projects, such as audio d-touch (Constanza et al. 2003) and BlockJam (Newton-Dunn et al. 2002), it derives the camera vision system and idea of physical blocks to represent virtual objects; from Pure Data it gets the power of its sound engine, allowing for the dynamic patching of oscillators, filters and other effects; OpenGL techniques for drawing sound take care of visualising the patchwork; and finally from the fields HCI and CSCW (Computer Supported Cooperative Work) they derive the decision to have a circular table in order to erase power structures and allow for ad-hoc organisation of work (Bødker 1996; Fitzpatrick 2002). All this is combined together with highly effective interface design, usability design, and sophisticated sound and graphics.

The inventors of the reacTable are not trying to hide their influences: a whole section on the reacTable website covers related projects and influences. They are realistic about the nature of inventing in the late 20th and early 21st centuries where technology is so complex that the inventor becomes a channel through which the techno-logic flows in this context. It is clear how the Nietzschean-Foucaultian genealogy provides a rather more modest and specific methodology to analyse inventions and human creativity, compared to the more romantic and historical views of creativity.[109] This also indicates that the technologies used in the reacTable are materials, with their affordances and constraints, that are taken into use and shape what is possible in the design itself. Just as Sax would be constrained by the physics of his materials and Moog with the nature of electricity and his oscillators and filters, the reacTable is defined by the programming environments chosen, the shape recognition libraries, sound engines, etc. From the designer's perspective the constraints are similar in all these endeavours, but the claim here is that the constraints of digital technologies are of a different nature than the constraints posed by physical materials or electricity – they are the designed constraints of symbolic systems [see chapter 4].

---

[109] A critique exemplified in Maggie Boden's work *The Creative Mind.* (1990). Boden writes: "What makes the difference between an outstanding creative person and a less creative one is not any special power, but greater knowledge (in the form of practised expertise) and the motivation to acquire and use it" (ibid. 35). Note that she defines this knowledge as practised, or embodied knowledge.

The reacTable has been presented in academic conferences, such as NIME, ICMC and Linux Audio; in arts festivals such as Ars Electronica, Transmediale; and now the startup company behind the technology has hired two musicians to tour the world with the instrument. In a manner similar to Van Koevering's innovation of the Moog synthesizer, the advertisements in the media for reacTable performances focus on the instrument itself and not the musician playing it: the market is being created. The reacTable has also its power-adopter (an early user of the technology with high social capital). In 2007, Jordà got a phone call from Björk Guðmundsdóttir who wanted one instrument for her next tour. In a 2008 workshop[110] Jordà explained that this resulted in an incredible boost in public awareness. Reminiscent of how Moog could follow Emerson, Lake and Palmer's tour around Northern-America by noting from where phone calls from music stores ordering Moogs were coming, the reacTable team was able to follow Björk around the world and see how she was using the instrument by typing the instrument name into the YouTube video sharing site. Minutes after her concerts would end, videos would appear on YouTube. Before Björk's tour, the reacTable team had around 8000 visitors to their YouTube page, but suddenly they had millions and that was only visitors to *their* page, not all the videos uploaded by people recording Björk's concerts. (Jordà: personal communication).

5.4.5 A Juxtaposition of the Acoustic, the Electronic, and the Digital

If we compare the origins of these three musical instruments, we see that they have much in common. They are all created by enthusiastic inventors because of some lack they identified in existing instruments, but also because of the sheer joy of inventing; of creating a truly original musical instrument. All of the inventors were young – in their 20s or 30s. All had to deal with certain inertia or initial criticism about their projects which they had to overcome. However, having invented the instrument is not enough, for the next step is to market it and make the invention a true innovation.[111] Here another skill is essential: the enthusiasm and convincing power needed in public relations and networking. The market has to be created and the best advertisement possible would be if a person with strong social and cultural capital (Bourdieu 1986) would adapt the instrument in their musical performance. In the three instruments analysed above, there were always such "power-adopters" who boosted the interest in the instrument and helped in creating the market. For Sax, Berlioz's comments, the military bands, and the saxophone classes in the Paris Conservatoire were important to create the demand. In a

---

[110] International Computing Music Research Workshop 2008 at the Open University, Milton Keynes, UK. <http://musiccomp2008.open.ac.uk/>

[111] Many excellent instruments have been made in history that have never gone to the innovation stage and are thus forgotten or stored as archives in obscure locations. The story of the Finn Erkki Kureniemi is a good example. In the early 1970s, Kureniemi designed synthesizers (both audio and video), MIDI protocols (he called it DIMI), hardware and software for music which were highly impressive. Hunt's MidiGrid is another example.

different communication situation, Moog was happy if print and broadcasting media reported on the use of the MiniMoog, thus creating the demand, but Koevering's relentless marketing proved indispensable. In a yet another media landscape (that includes a new power called Web 2.0, see O'Reilly 2005), the reputation of the reacTable spreads around the world with the link-ranking structure of YouTube and the social networking of blogs, open source communities, and DIY people that are impressed by the technology and find inspiration in its design. Table 5:3 describes the structural similarities that can be detected in the creation of the three instruments:

	The Saxophone	**The MiniMoog**	**The reacTable**
Inventor	Adolphe Sax	Bill Hemsath	Team at PF
Internal inertia	-	Bob Moog	-
Opposition	The Association	Musicians' Union (AFM)	-
Critics		Ussachevsky Babbitt Buchla	Various Catalan media
The power-adopter	Berlioz Military bands Debussy	Sun-Ra Keith Emerson Wendy Carlos	Björk
Marketer/innovator	Sax	Van Koevering	Sergi Jordà
Strategies for innovation and establishment	Publishing house of music written for the Sax. Teaching the instrument at the Paris conservatory.	Electronic music evenings with concerts and hands-on sessions.	The release of reacTV source-code. The use of YouTube for marketing hype.
Time to establish	60 years	30 years	5 years
Economical boost	French military bands	Rock music Psychadelia "Switched-On Bach"	Björk and DJ culture.
Networks enrolled in development	2-3	10	100 (OSC, table, HCI, etc)
Size of team (principal)	1	2	5-10
Reason for invention	Overblow at octave Stronger sound Singing	Interest in new sounds. Joy of exploring electronics	Physicality in virtual instruments. Visualisation of synthesis.
Nonhuman actors	Clarinets, tools, concert halls, materials.	Oscillators, electronics, technoculture	Projectors, cameras, realtime video libraries, shape recognition software, sound synthesis, etc, etc.
Patents	Whole instrument	Filters	-
Public awareness	Word of mouth	Print and broadcasting media	The internet, YouTube and blogs in particular

Table 5:3. A comparative analysis of the actor-networks involved in the invention and innovation of three musical instruments whose history are relatively well documented

There are plenty of differences as well, and here we want to focus on differences in materials, technique and design rather than the social or historical issues. Looking at the material context of these inventions, let us first picture Sax in his workshop. There is an abundance of raw wood, brass and reeds. There are drawings, instructions, manuals and models. And there are ready made instruments that can be used as prototypes. Sax starts as an apprentice of his father and learns his skill over the years. It is a skill that requires tacit knowledge and intuition of materials and sound physics. His knowledge is highly embodied and non-theoretical. The development of the saxophone involves iterations of designing, building and testing. Sax is working with what physical materials afford in terms of sound. A change in curve, in width, in material means a change in sonic timbre. Here the physical materials provide the basis for explorations of sound and interface design. The interface is conditioned by the properties of the materials and various solutions are used to extend the scope or possible range of the human body with the use of levers, keys, and other mechanisms. Sax's instruments are therefore also dependent upon the physics of the human body as well.

Figure 5:8. Bob Moog playing with an early version of the reacTable. Moog would be familiar with the internal workings of the reacTable, although the interface is novel.

Bob Moog's entry into his field was similar to that of Adolphe Sax: as a young man he enjoys working in his father's electronic workshop. Just like in Sax's atelier, there are manuals, drawings, schemes and models, but in addition there are now manuals for the technological ingredients. Moog acquires his first chops in building electronic instruments by copying an existing instrument, the Theremin. As someone with an inquiring mind, he starts to explore the material properties of electricity, magnetic waves, oscillators, capacitors, inductors and transistors. Here, the creative process is also one of designing, building and testing through a cycle of iterations. The idea comes from working with the materials, not from sitting in an armchair philosophising. However, unlike Sax's materials, Moog's come with instructions and schematic diagrams that describe their behaviour. There is an increased logic of calculation, science and engineering. Fourier's and Helmholtz' theories are now well known and Moog can draw from that knowledge in his designs of oscillators and filters. As electricity flows through wires, interfaces could be built in any shape or form, but they are still constrained by physical mapping, so when the instrument has been wired up, its functionality is not easily changed. This gives the machine an instrumental quality, something with character that has to be explored in depth.

The team behind the reacTable lives in a different world again. The "workshop" is in the form of offices in a university department with desks and computer monitors. There is nothing there that Sax (paying a visit from the past in a time-machine) would associate with music, except perhaps a MIDI keyboard lying around somewhere. The materials of the reacTable team are many: tabletop made of glass, projector, camera, physical cubes, designed shapes sticked onto the cubes, amplifiers, speakers and computers. Behind this surface lie audio programming languages, shape recognition systems, motion tracking libraries, sound visualisation systems rendered on screen cards, mapping engines between gestural recognition (cubes and/or fingers) and sound and audio engines, etc. And supporting those "materials" lie systems of digital signal processing, programming languages, extension libraries, operating systems, hardware protocols, physical interfaces, etc. The recursion is practically infinite. From an actor-network perspective, Sax enrolled perhaps 10 (blackboxed or not) networks in his instruments, Moog would enrol 100, and the reacTable team (so much larger than Sax's or Moog's team) might speculatively enrol (if we follow the exponential curve) 10.000 networks in their instrument. There is a drastic increase in complexity, which means that the inventors necessarily have to rely on black boxes. Moog might not question the oscillator until it malfunctions and the reacTable team would not question the shape recognition library they used until it proved insufficient. When it did, they had to open up the black box and reimplement it (Kaltenbrunner & Bencina 2007). All the systems used in the reacTable originate from technoscientific knowledge. There are relatively few physical material properties at play (although of course at the machine level we find matter) compared to the symbolic code that constitutes its internal (and symbolic) machinery. The inventors are knowledgeable about digital signal processing, sound physics, audio synthesis, gesture recognition, human-machine interaction, and the culture of musical performance. In general it is non-tacit knowledge of symbolic systems in the form of code; how they work and interact with each other. From a design perspective, any interface can be designed for any sound. There is no natural mapping between gesture and sound in digital systems. In acoustic instrument the performer yields physical force to drive the instrument. In electronic instruments there can be mixture of both physical and electric force. In digital instruments, the force is virtual, it can be mapped from force-sensitive input devices, but that mapping is always arbitrary, in contrast to what happens in physical mechanisms. The concept of virtuosity is therefore transformed according to which type of instrument the performer is operating.

There is a huge number of factors and events that led to the innovations of the Sax, the MiniMoog and the reacTable. It becomes clear that "originality" is not a term that is useful in a genealogical analysis of technological inventions like these; it has a better place in obituaries or second-rate history books. A clear direction of conceptual and material energies, being at the right time and place, a dose of luck and an awareness of the importance of innovation, and public relations all have to come together in order to create an instrument with longevity and popularity.

5.4.6 Conclusion

This section has illustrated the strong similarities in the innovation of musical instruments. Through a philosophical-genealogical methodology, patterns were extracted that are shared in the cases of all the inventions. However, the increased technicity of the electronic and digital technologies became obvious, and this was discussed in the context of the actor-network theory and the theory of epiphylogenesis.

Finally, this section compared the material affordances of physical materials, electronic materials and digital materials respectively. The difference in mapping became obvious through that exegesis. For example, if a keyboardist had the idea of inverting the pitches of a keyboard, the piano player would have to rebuild the piano, the synth player would have to enter the electronic mechanism of the synthesizer and add or rewire some voltage controllers, but the digital musician might simply change it through one line of code such as:

```
note = 127–note; // pitch is here represented linearly as MIDI numbers (0-127)
```

That said, I acknowledge the blackboxing of software, or digital musical systems, and do not pretend that they are completely open. This topic is explored in section 7.4 through interviews with practitioners. The thrust here is that we find an ever intensifying increase in blackboxing and reliance on established technological networks. This point becomes clearly apparent in the evolution from acoustic instruments, through electronic instruments, to digital instruments. Another study, involving open source culture might give a more detailed account of this situation, but space will not allow for that here (see Tuomi 2001; Østerlie 2003).

5.5 Taxonomies of Digital Musical Instruments

In section 4.3.1 we read about Nietzsche's objections to classifications in the sciences, presented as a captivation of nature's chaos into cold and gray conceptual nets. The evolutionary biologist Stephen Jay Gould (who co-authored the theory of punctuated equilibria with Niles Eldredge whose phylogenetics of cultural artefacts was presented above) disagrees: "Classifications are theories about the basis of natural order, not dull catalogues compiled only to avoid chaos" (Gould 1989: 98). For Gould, classifications set things into order, compose reality and provide constraints. It shall not be pretended here that musical instruments are based on "natural order," nor decided whether such order exists outside the classifying subject and the systems used. However, having seen the limitations of phylogenetic trees in the classification of musical instruments, it becomes relevant, from the perspective of the digital instrument designer, to present different taxonomies for the design of musical instruments. These taxonomies will be beneficial to the designers, critics, composers and performers of digital musical instruments. They serve as talking-points made in order to clarify aspects within this discourse, but they are not intended to be exhaustive in any way or form.

5.5.1 Introduction

As already stated, the classification of musical instruments can be performed from various analytical perspectives: their origins and genealogy (phylogenesis), their interaction modes, materiality, usage in musical genres, type of sound, etc. It is the task at hand that defines what type of taxonomy is relevant. Classification is not a one-dimensional activity resulting in perfect structures, although the tendency for believing so has historically been strong.[112] The creation of one instrument might originate from many parents, a fact which is not well represented in genealogical trees. The concepts and classifications are often so interrelated that "they are practically indivisible, except on an analytical level" (Kartomi 1990: 271). Furthermore, the categories used in such classifications are culturally dependent, a good example being Southeast Asian musical traditions that classify musical instruments into male and female (ibid.; Doubleday 2008). These taxonomies can change through time or in different cultural settings and geographical locations, as the example of the saxophone exemplifies (being woodwind in one tradition but brass in another).

In Western music, instruments are typically described as belonging to one of the following groups: woodwind, brass, strings and percussion.[113] This is the classification composers use when they write pieces for orchestras, organising their score with the woodwind at the top, then brass, percussion and strings at the bottom. This system is notoriously imprecise, where some instruments do not fit anywhere and others find themselves in more than one category (Hopkin 1996: 30). There are other classification schemes that allow for better classifications. These schemes are typically found in the scientific study called organology: the study of musical instruments which was established in the late 19th century. At the time museums had started collecting and exhibiting musical instruments and the musicologists Erich Moritz von Hornbostel and Curt Sachs developed a system for classifying musical instruments which they published in 1914. This system is arguably the most widely used taxonomic system for musical instruments. Hornbostel and Sachs based their system on a 19th century system by

---

[112] In fact the structure of the Linnaean system into kingdom, phylum, class, order, family, genus, and species (as used in biology) originated long before scientists knew that life evolves.

[113] Such taxonomies are obviously obsolete when we consider the computer's capacity for creating any sound in the world (it is not a matter of whether it can, but how it is done. We are still discovering synthesis methods that afford practical methods for digital sound synthesis). In the context of the computer, a taxonomy of sounds has been proposed that is independent of the physical material used to produce it or the synthesis type (Rocchesso & Fontana 2003; Augoyard & Torgue 2005).

Mahillion, who, in turn, had taken his categories from the Natya Sastra, a two millennia old Indian treatise of dance and music (Khartomi 1990: 197).

In the Hornbostel-Sachs system, instruments are classified into the following classes: idiophones (bells), membranophones (drums), chordophones (string instruments), and aerophones (flutes and brass instruments). Electrophones were added later (Campbell et al. 2004: 40). Table 5.4 shows the musical instrument classification scheme and the typical human interaction with the instrument that excites the sound. It is clear to us, however, that there is a category lacking in the electrophones, i.e., d) digital. This is addressed by Kvifte (1989), who proposes an emic description system for digitally synthesized sounds and their generators. These types of taxonomies are useful, but highly abstract and very rough in their categorisations. Some instruments do not fit anywhere, whereas others could be placed in many categories.

Major Class	**Method of excitation**	**Examples**
Idiophones: Vibrating rigid object	a) striking b) plucking c) rubbing d) shaking	Xylophone, gong, triangle Jew's harp, lamellaphone Glass harmonica Maracas, jingles
Membranophones: Vibrating stretched membrane	a) striking b) rubbing c) singing	Timpani, bass drum, tabla Rommelpot Kazoo
Chordophones: Vibrating stretched string	a) striking b) plucking c) bowing	Dulcimer, clavichord, piano Harp, guitar, harpsichord Violin, viol, hurdy-gurdy
Aerophones: Vibrating air	a) air jet b) mechanical reed c) lip reed	Flute, recorder, organ flute pipe Clarinet, organ reed pipe, accordion Trumpet, bungle, cornetto
Electrophones: vibrating loudspeaker	a) electronic b) electromechanical c) electroacoustic	Ondes Martenot, synthesizer, computer Hammond organ Electric guitar

Table 5:4. The traditional taxonomy of musical instruments, based on Hornbostel-Sachs. Taken from Campbell Greated & Myers (2004: 40)

In the early studies in organology (Sachs 1940; Clutton 1961), the assumption was that improvements or inventions in the design of musical instruments originated from the demands of composers. When composers wanted a new sound or performance technique, the maker of musical instruments would try to accommodate them. The "early neglect" of the piano forte shows that "when an instrument maker invents something before the composer is ready for it, his reward will be slight" (Clutton 1961: 88). More recent studies reveal that there is not any singular relationship that can be used to explain how instruments evolve. There are economic, technological, political and social forces that lie behind all such evolution (Libin 2000: 197). The instrument maker might respond to the needs of musicians whose practices are changing due to new social structures which, in turn, define where and how music is played. The composer might be taking advantage of new cultural settings and begins composing pieces where new techniques are needed. The increasing professionalisation of musical performance results in more technically able performers that can interpret more complex pieces, in the same stroke pushing the limits of the instrument itself (DeNora 1995, 2003; Pinch & Bijsterveld 2003a). The strong tradition of craftsmen guilds created a certain resistance for innovation (as we saw in the example of Sax above) and political control demanded that the development of instruments associated with public revolt (such as drums, bagpipes, horns, etc.) were hampered. Bijsterveld and Schulp (2004: 667) show how innovation in musical instruments normally happens at intersections, or what they call the "areas of go-betweens." By this they mean that innovation typically takes place through individuals who operate with "creative marginality" in the intersections of disciplines and traditions.[114] The power of the go-betweens is that they are able to "transfer and 'translate' methods of analysis from one discipline to the problems of other disciplines" (ibid: 667). In Maggie Boden's (1991) vocabulary, one would talk about the merging of search spaces, or even transformative creativity. These observations explain to some degree the immense developments and innovations that have happened recently in the field of electronic music, both in technical and aesthetic terms.

It is therefore hard to find a logic in the evolution of musical instruments. The field is highly complex, interdisciplinary, and dependent upon a multiplicity of technological and sociological factors. However, considering the idea of epiphylogenesis and the tendencies of our technical systems, it is illustrative how the technological evolutions of musical instruments define the music we listen to and the contexts in which we listen to that music. Musicianship changes as well due to technology: ergonomics improve, artificially intelligent assistants help composers, sound becomes cleaner and more sophisticated, and musical tasks change. Having found the genealogical approach too diffused in digital instruments, and questioned the validity of classification systems, it is proposed in the next section to generate an alternative system: a taxonomy of musical functions and tasks. This is solely done from a practical purpose, in order to enable and ease the analysis and discussion of digital musical instruments, and not to be seen as rigid systems of categorisations.

[114] Creative marginality is a concept found in the work of Dogan and Pahre's (1990).

5.5.2 The Intelligence of an Instrument

In general, the simulators presented below in Figure 5:10 can be analysed as *production tools*, whereas the patchers and the controllers combine an area outlined as *expressive tools*. Production tools do not lend themselves for realtime work and rarely explore the computer creatively as a medium for expression. Their goal is to translate physical practices into virtual. They are good as off-line tools for composition, recording, manipulating data, mastering, post-production and other non-performative activities. Expressive or creative tools, on the other hand, function like a workshop for building expressive and creative systems. Here the user can design an instrument, a composition, or a combination of those, with infinite possibilities in interfaces and mapping of gestural data to sound. This analysis could benefit from a taxonomy of instrumental knowledge.

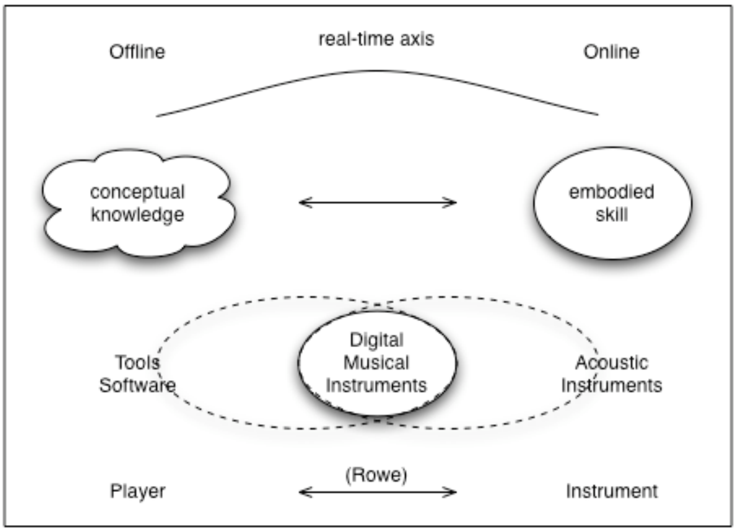

Figure 5:9. A concept field of production and performance systems. It is clear here how this maps to Ihde's ideas of embodimental relationships and hermeneutic relationships with the world through technological artefacts.

In Figure 5:9 we see how on the one side of the continuum we have production systems where all kinds of automation can take place, from artificial intelligence systems to random generation. These systems are highly defined by conceptual knowledge of the domain field and therefore culturally dependent. Nonetheless, there are technologies that allow for embodied interaction, and this side maps to the creative systems intended for real-time use. Furthermore, Rowe (1993) makes a distinction between the paradigms of instruments and players as musical tools: the players are off-line and "contain" the music, whereas the instruments are there for realtime control. This relates to Rowe's dimension of score-driven versus performance-driven systems. In Figure 5:9 the left side would typically fit the player and the score-driven side of the axis, where at the right side would be the performance instrument. As these are continuous scales, digital musical instruments can be placed anywhere on the scale, but acoustic instruments would typically be placed on the right side of the axis. This continuum is therefore related to the epistemic nature of the tool. We design our tools with complex algorithms that represent musical models and even utilise machine learning (a non-symbolic technique) to implement the intelligence of our systems.[115] The graph shows how off-line production tools can have high degrees of automation and non-performative contexts. These are typically the production tools of composer studios. On the right side of the continuum are instruments in their simplest gesture-to-sound mapping dimensions (practically a construction of an acoustic instrument).

5.5.3 Taxonomy of Instrumental Functions

Looking at musical instruments as they appear to us today, we are faced with a panoply of tools for playing, composing, editing, building instruments, etc. Furthermore, the current increase in the use of AI techniques or computational creativity within experimental software, such as in academia or street-DIY, is drastically changing the field. In order to clarify the picture, we propose the following taxonomy of musical instruments based on the mode we *interface* with them and their *physical substrata*:

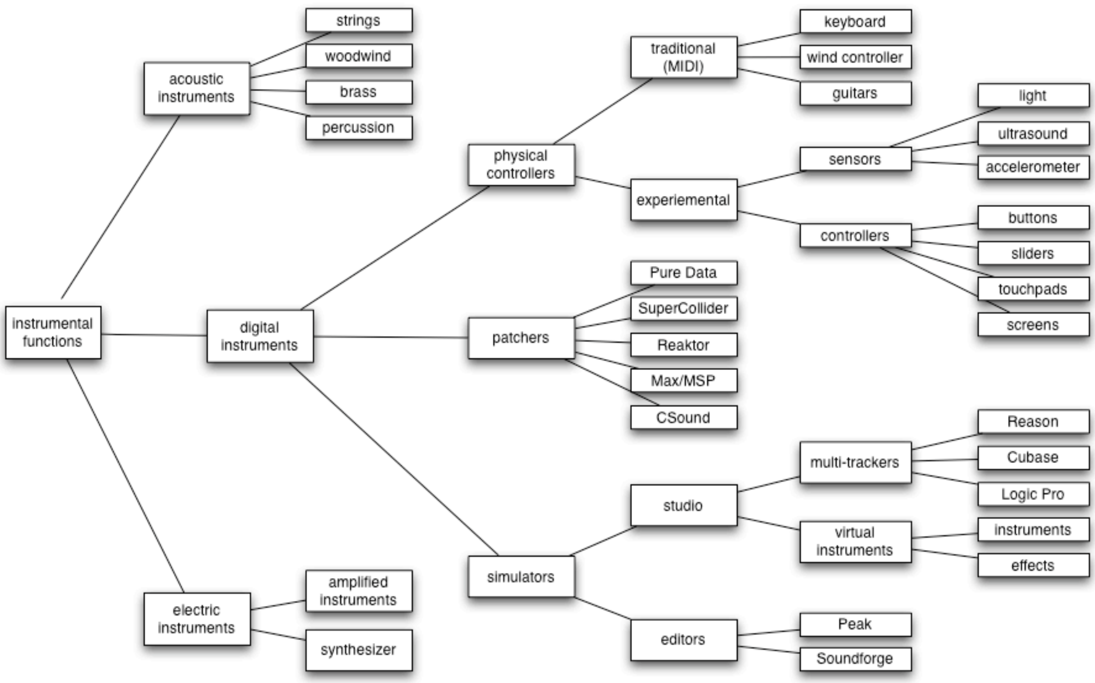

Figure 5:10. A tree of instrumental fuctions in the realm of music.

This taxonomy is not intended to be complete in any way, but shows the main categories of instruments and tools in use today. Obviously, as in the Hornbostel-Sachs system, the categories are not exclusive and should rather be seen as general heuristics. Some instruments might extend over more than one category. The categories of **acoustic** and **electronic** instruments are of course a constant source of inspiration in both sound and interface/interaction design, but the study of their qualities are outside the scope of this thesis and will not be studied here. **Digital**

[115] Jordá (2005: 92) conflates intelligent instruments with automation, but considering latest developments in musical systems, I think it is important to make that distinction clear.

instruments can be divided into *physical controllers* that provide a gestural control over a sound engine in the form of software; *simulators* that base their realistic design upon the ergonomics of real-world practices, such as writing notes on paper, layering sounds on tape tracks, turning knobs, pressing buttons or moving sliders, etc.; and *patchers* that provide various functionalities for programming audio synthesis, compositional algorithms and the mapping of data from inputs such as human gestures or environmental variables. The origin of the patchers can be found in academia and research laboratories, but with the popularity of personal computers, high computer literacy in young people, and the commercial companies entering this field, the patchers have now become a popular way for people of all styles and musical traditions to create music, instruments or physical controllers or installations.

Various sections in this thesis deal with the nature of musical performance or composition using patchers: section 5.2 presented a survey that focussed on people that have experience of using both acoustic instruments and patchers in their work; section 7.4 is an extended study of the mindset of professional musicians using patchers in their creative work; and section 7.5 explores the response of the *creators* of patchers (or audio programming language designers) to survey questions posed to them.

5.5.4 Taxonomy of Activities

Yet, another taxonomy is needed for the explanation of what we *do* with these tools. What are the main activities people perform with their musical tools? Below we present a taxonomy based on *activities*:

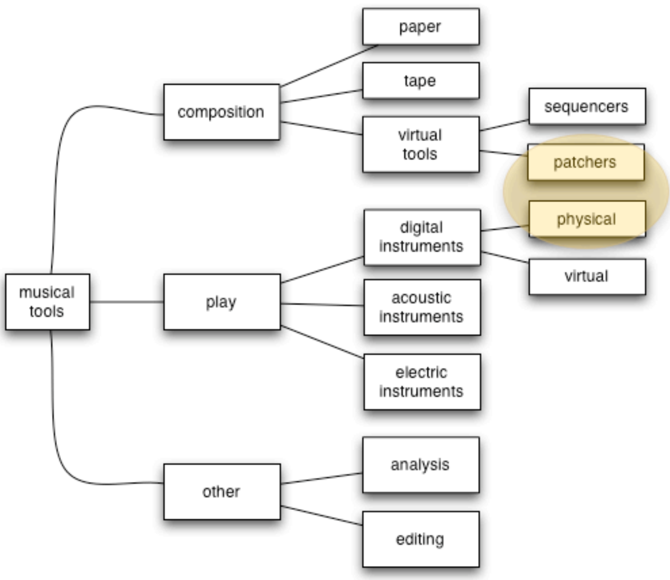

Figure 5:11. A taxonomy of activities.

Here musical actions are divided into those of playing and composing (and include other tasks for the sake of completion, such as analysing musical onsets or timbre, editing, etc.). If we look first at instruments for musical *play,* we see that for millennia we have designed acoustic instruments and they have become ever more sophisticated and specialised. The roughly half a century old tradition of electronic instrument design has more or less consisted in the amplification of acoustic instruments or the simulation of them through the use of analogue synthesis with oscillators. The interfaces here are typically quite conventional and based on the tradition of acoustic instruments. Black horses like the Theremin are the exception. Digital instruments introduce another mode, as they are either *physical* (traditional MIDI controllers, experimental controllers) or *virtual* (screen-based or sensor-based – for example in video analysis of human gestures).

The pen and the paper have been the traditional tools used for *composition* for some centuries now. There are various forms of musical notation, but it was mainly in the European Renaissance that the musical score became a sophisticated form for the encoding of musical expression. With electricity other means of composing music emerged, such as the electromagnetic tape. Music could now be assimilated using any sounds in any combinations which prompted Varese to define music as “organised sound” (Varèse 1936). When the sound of a train passing could be layered under the sound of a flying mosquito, all dynamic amplification relationships became relative. The French tradition of *musique concrète* led by Pierre Schaeffer is a good example of the use of tape for compositional purposes.

With the invention of digital computers, a revolution took place in terms of what became possible in composition. Computer scientists immediately started to emulate the way we write or play music through representing scores and piano keyboards on the screen. With popularisation of personal computers musical software even became a profitable commercial field. Having access to the CPU of a computer through human scriptable programming languages meant that it was possible to write music in very different and novel ways. Suddenly the computer could perform complex algorithms that would not make sense to perform using pen and paper (Pecquet 1999; Supper 2001). Algorithmic music through generative rules became possible (Xenakis 1996) and various experimentations were conducted in terms of how the machine could be controlled through habits already established from practising acoustic instruments. The patchers are the best environments for this type of musical composition. Languages such as SuperCollider (object oriented textual language) or Pure Data (graphical programming language) are good candidates. In practice, the patchers are used not only for composing algorithmic or linear music, but also to “compose instruments”[116] (Bahn & Trueman 2001; Schnell & Battier 2002) that can be used in live performances. The patchers also used in a style of performance, called livecoding, which mixes the *performance* of music with the *composition* of music and the *design* of instruments. All this happens as a live performance in front of live audience (Collins et al. 2003). The patchers function as creative platforms for composition, instrument design, and play. Moreover, in the field of NIME, the patchers are most typically used as the conceptual engines of digital instruments, i.e., when someone designs a new controller, it has to be coupled to a sound-engine through some software and normally people use a patcher for that. I have outlined an orange circle around the area in which this

[116] Composing instruments was a term often used by early pioneers in electronic music, such as David Tudor and Gordon Mumma.

thesis is conducted, with a general interest in the digital musical instrument, but a specific focus on the epistemic dimension, or the instrument's compositional nature.

5.5.5 Taxonomy of Interactions

This research is conducted in a field where patchers are used to design the conceptual engines that serve as sound engines, compositional tools, musical collaborators, instruments and mapping couplings from the space of human gesture to the parameter space. In digital instruments, the use of metaphors from the physical world of hardware and acoustic instruments is the most common practice.[117] This does not have to be so. In contemporary experimental instruments designed for the computer we can draw upon a reservoir of possible action metaphors that are easily understood by users as they relate to other common activities (such as computer games, interface design, web design, hardware, etc.).[118] Figure 5:12 illustrates from where metaphors for interaction can be derived in the design of new musical interfaces.

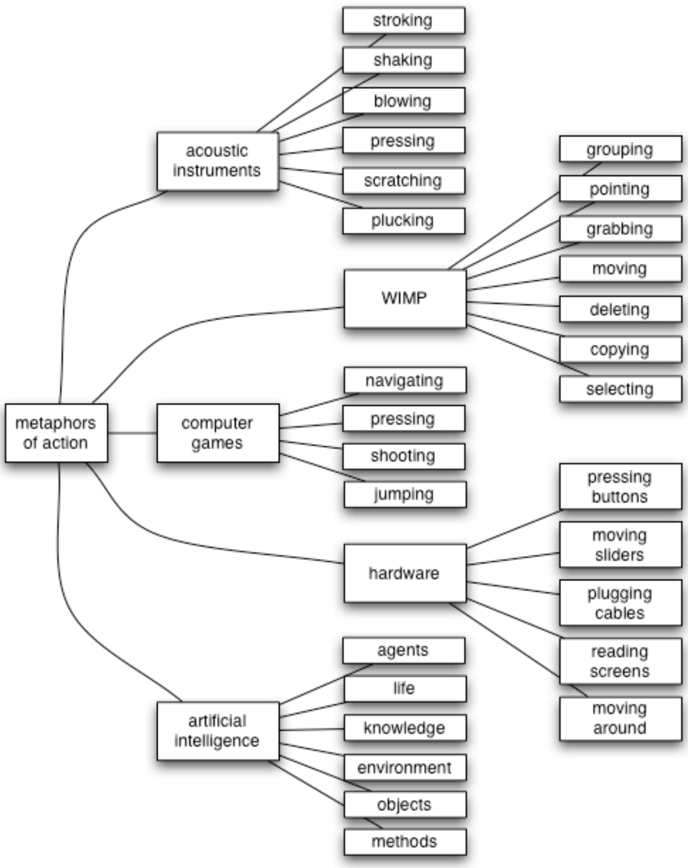

Figure 5:12. A taxonomy of interaction

[117] This is arguably because much of the innovation in the field is driven by the commercial software houses and they tend to build on known metaphors for design as it is easier to market products when they relate to already established activities.

[118] A relevant research question becomes: to what degree do people's mastery of alternative interfaces depend upon their experience of fields like gaming and computer use in general?

This taxonomy of actions is suitable for NIME devices, but can also be of use for the design in screen-based musical systems. At the time of writing, laptops and smaller gadgets are increasingly built with various types of alternative sensors such as accelerometers, temperature sensors, cameras (which can be turned into light sensors, colour sensors, movement sensors, etc.), touchscreens, etc., together with more traditional ones, such as microphones, trackpads, mice and keyboards. The question is how to use these channels of input creatively (Fiebrink et al. 2007). The hardware affordances are therefore practically infinite for the designer of digital musical systems. Explorations of Nintendo Wii, iPhone, gamepads, and the MacBooks show that hardware made for general purpose can become interesting, constrained devices for musical expression and production.

The digital computer, in substance free from material properties and traditions, is an ideal platform for experimentation in the creation of musical instruments. The semiotics of interface design, the ergonomics of interaction design, and the general openness where the reservoir of human actions can be used as metaphors or metonyms in the design of musical instruments. These issues will be explored further in the next chapters.

5.6 A Dimension Space for Musical Epistemic Tools

Birnbaum et al. (2005) provide an interesting visualisation of the parameter space of musical devices. They identify seven dimensions of continuous scales onto which the properties of specific musical devices can be mapped. They focus on *musical control* (whether the instrument's expression is at the timbral, note or score level); *degrees of freedom* (an axis that represents the number of input parameters that can be controlled); *feedback modalities* (the degree of realtime feedback to the user); *inter-actors* (number of people involved in the performance of the device);

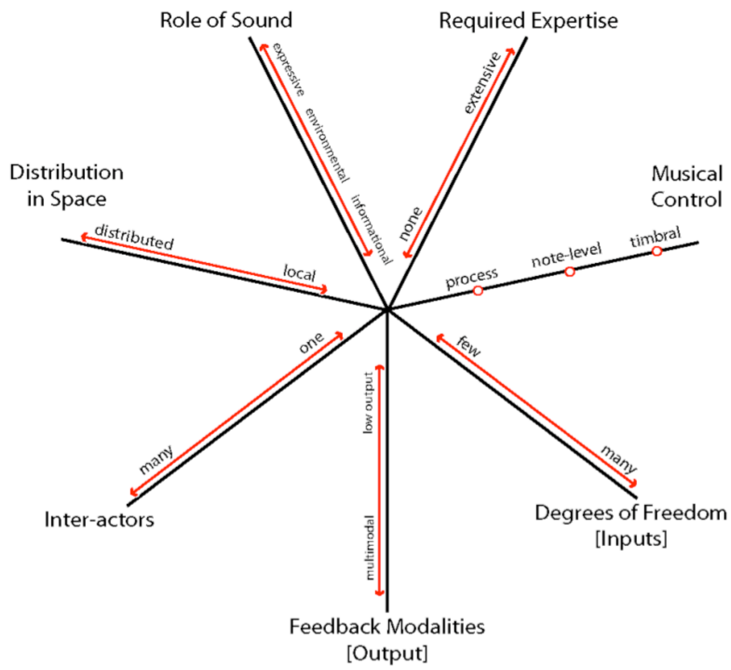

Figure 5:13. A "phenomenological" dimension space. Birnbaum et al. (2005)

distribution in space (the total physical area inhabited by the instrument); *role of sound* (informational, environmental or expressive); and *required expertise* (representing the level of practice and familiarity needed in the performance). As shown in Figure 5:14, this visualisation of parameter spaces can be highly useful as a conceptual tool for evaluating and classifying digital musical instruments.

In chapter 1 it was stated that the approach of this thesis would be epistemological-engineering based rather than phenomenological-engineering based. This perhaps echoes a distinction Truax (1989) makes in one of the early texts about the epistemic nature of computer music languages. Truax defines a dichotomy between the *structure of the system* and *the behaviour of the system in use* (ibid: 156). The former represents the affordances the systems lends for thinking in general and conceptualisation of musical problems in particular. The latter is how the system functions when used in a realtime performance.

I propose that we look here at musical systems from the epistemological or music-theoretical perspective and forge another type of dimension space, that of epistemic tools. Whereas Birnbaum et al's approach is phenomenological, i.e., focussing on the human body and its expressive potential in the relationship with digital tools, the epistemic dimension space addresses the culturo-theoretcial aspects that so prominently define the nature of digital musical systems.

In the epistemic dimension space of digital musical systems, the parameter axes are:

- *integrated composition* – The degree of musical composition inscribed in the tool.
- *expressive constraints* – The expressive limitations outlined by the tool's design.
- *autonomy* – The degree to which the instrument affords expressive autonomy. Here the performer offloads certain musical tasks to the instrument that

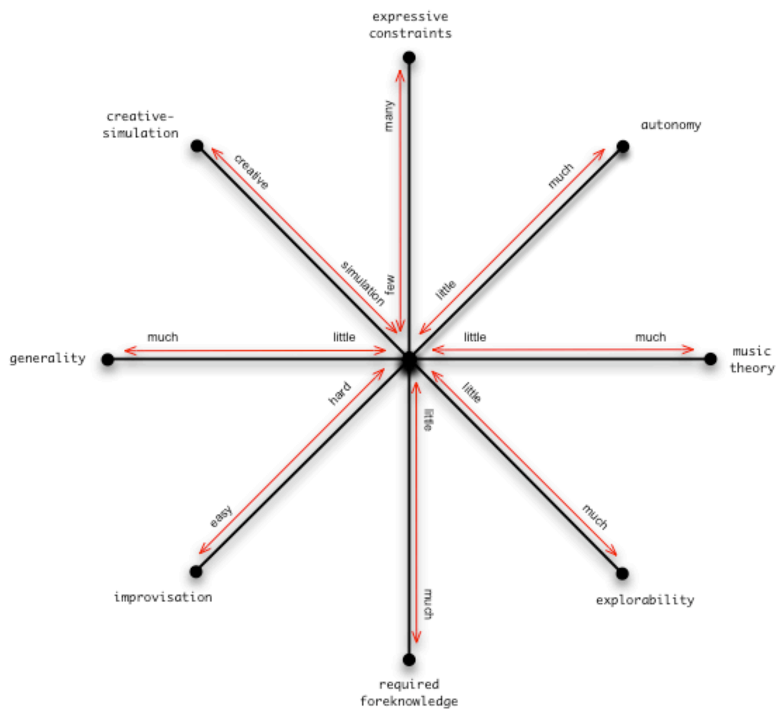

Figure 5:14. An epistemic dimension space. Here the focus is on the properties that characterise the digital musical instruments

continues performing (often with some AI capacity that makes its performance progressively more interesting).

- *music-theoretical mass* – The amount of culturally specific music theory in the instrument. This includes compositional elements.
- *required foreknowledge* – Many instruments do not need much musical knowledge (both performative and compositional) as they contain it already.
- *creative – simulation* – Whether the instrument is novel or focussed on imitation of established tools and practices.
- *improvisation* – The degree to which the instrument lends itself to free improvisation.
- *generality* – How open in expression the instrument is and how well it copes with the multiplicity different of musical situations.
- *explorability* – How much depth the instrument holds. This factor is critical with regards to how engaging the instrument is and affects learning curve and the possibility of flow.

Obviously, many of these axes would not be relevant in the analysis of acoustic instruments. It is precisely because of the epistemic nature of digital tools that they become relevant in the analysis of digital musical instruments. The above dimensions address parameters that are unique to heavily abstract, conceptualised and symbolically designed musical tools.

Below, I will analyse a few digital musical systems using the epistemic dimension space. This is a personal mapping of some well known digital musical instruments onto the epistemic tool dimension space. It is appropriate to begin by analysing Waisvisz's instrument The Hands. Firstly because it is one of the most famous digital musical instruments, but secondly due to Waisvisz's statement that he had to regularly "freeze" the development of his instruments in order to gain an in-depth relationship to them.[119] The analysis of the Hands is divided into two distinct graphs, one for the general interface (the physical controller) and the other for the specific instrument (when the controller is coupled to a mapping and sound engine) [see section 7.2.3].

[119] Note that "instruments" are here in plural. Although The Hands can be seen as an instrument, it is only when coupled with sound and mapping engines that we can talk about an instrument. Every new sound engine results in a new instrument, requiring him to rehabituate himself with the controller's functions.

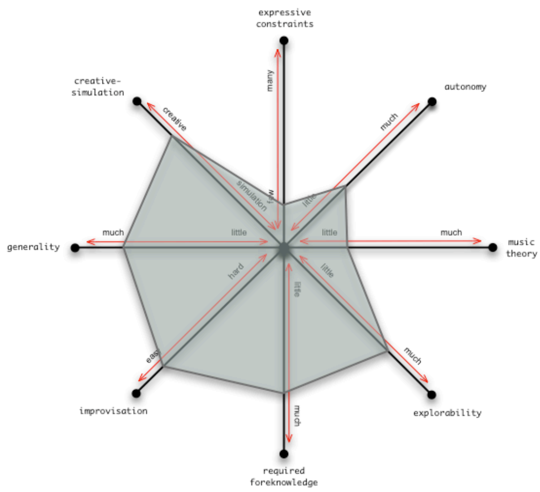

Figure 5:15. The Hands – As Gestural Interface.

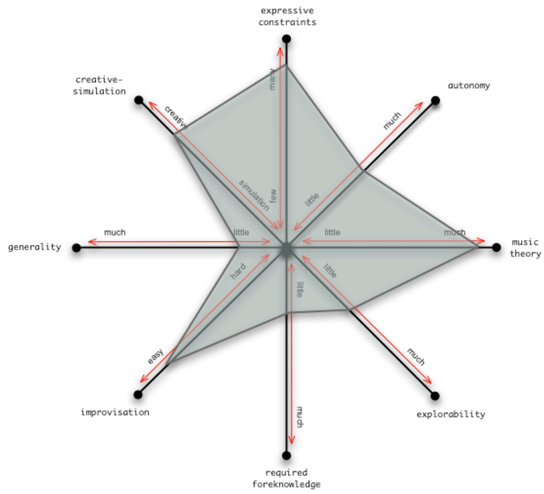

Figure 5:16. The Hands – As a musical (composed) Instrument.

In Figure 5:15 we see the Hands as gestural interface: it is open for improvisation; it affords large areas for exploration and there is relatively high degree of musical foreknowledge required to design the tool or compose for it. The controller is very general and could be utilised in almost any type of music. From its nature as a sensor device, we see there is little music theory inscribed in it (a keyboard has much more for example), and few expressive constraints.

Exploring Figure 5:16 however, where the Hands are manifested as musical instrument, it is clear how the music theory has become stronger; more expressive autonomy and constraints have been designed into the system. It is therefore less general and can be used in fewer musical contexts. It is still ideal for improvisation in most cases.

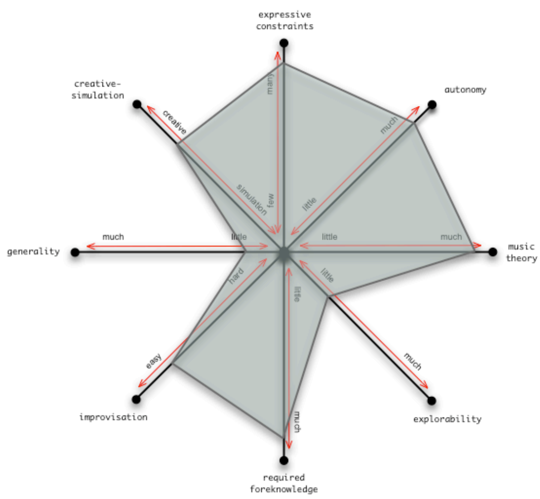

Figure 5:17. Voyager

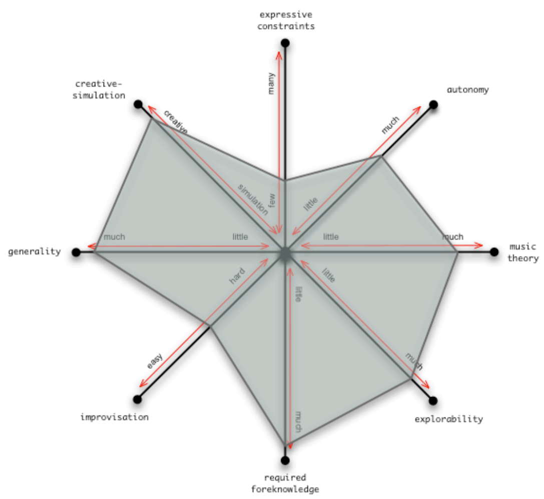

Figure 5:18. SuperCollider

The Voyager system by Lewis (2000) is a generative performance system that analyses the playing of the human performer and responds appropriately. As such, it is less an instrument than a co-player. As seen in Figure 5:17, it scores high on improvisation, music theory and expressive constraints. It is clearly a system that is designed by Lewis, for Lewis, and would not work for all musicians, though others have played with it, for example Evan Parker.

SuperCollider (Figure 5:18) is by many considered the most expressive and free musical environment available today. Such freedom naturally comes at the cost of a rather low-level working space, where much foreknowledge is required. It also implies certain musical theoretical concepts, for example in the way its Pattern libraries are built. It affords great explorability, it is very general in its use and scope, and it opens up to almost infinite fields of creative productions. Naturally, considering SuperCollider's openness, it can be located practically anywhere on all the axes, as the system can be designed for all purposes. I place the SuperCollider graph next to Voyager's as obviously, *using* SuperCollider, one can build such a constraint compositional/performance system.

The reacTable (Figure 5:19) is an excellent improvisational tool and it affords much explorability. It does not require much musical knowledge, nor does it contain much music theory as its functions in the current state is primarily on audio synthesis, simulating the modular synthesizer. (This is why it fits in the middle of the creative-simulation axis). It is quite constrained in expressivity; and in the context of generality, it depends whether we are analysing the physical instrument itself or its software-side, the mapping and sound engine.

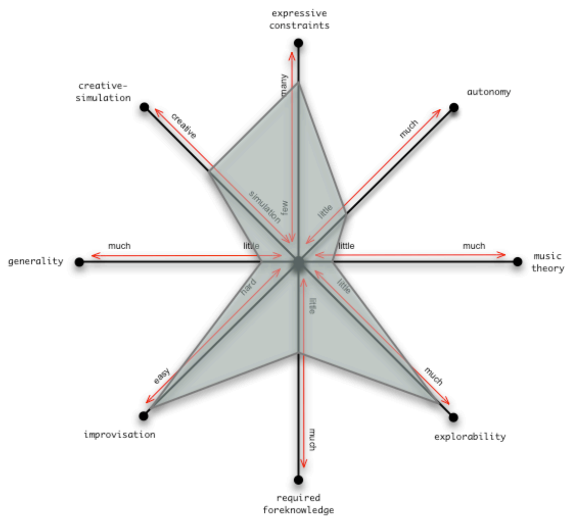

Figure 5:19. The reacTable

Reaktor (Figure 5:20) is in many ways related to Max/MSP or Pd. It is a modular data and signal-flow patch space where the user is able to use raw ingredients to build up complex applications or compositions. However, the building blocks of Reaktor are typically on a much higher level than in Max or Pd, which results in less expressivity, but at the same time a smoother learning curve. This also means that it is easier for the user to get "good sounding" patches working quickly as there is much more knowledge (both musical and signal processing) inbuilt into the individual building blocks of Reaktor than in Pd, for example.

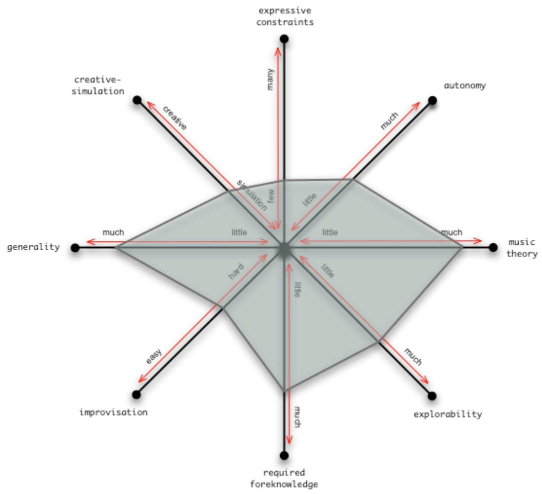

Figure 5:20. Reaktor

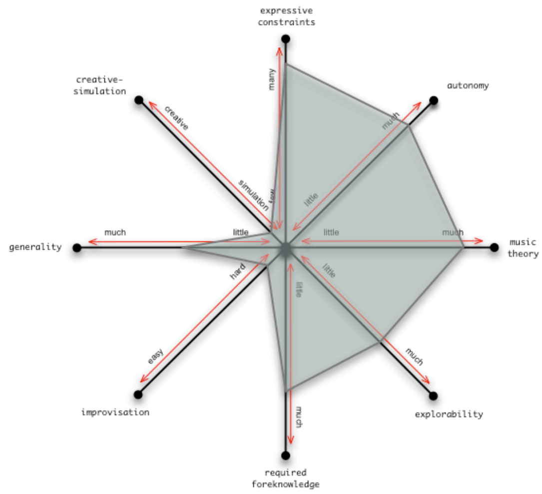

Figure 5:21. Reason

Unlike Reaktor, Reason (Figure 5:21) tries to realistically simulate the rack hardware devices typically found in recording studios. It is a pure representational simulator where the screws, the painting tape (used for labelling knobs), and the swinging cables on the back are all attempting to create the feeling of the "real thing." As such, it is a successful and beautiful software. However, it scores low on improvisation, as it is not really a device for embodied expressivity. It is highly deterministic and contains expressive constraints and music theory to a large degree. Participants in our surveys stated that they could hear the patterns of Reason, for example.

5.7 Conclusion

Bertelsen, Breinbjerg and Pold (2007) apply Beaudouin-Lafon's [see section 6.3] ideas of instrumental interaction, and define a specific quality they call *instrumentness*: "[Instrumentness] can be seen as a HCI perspective addressing how the interface mediates the user's relation to the software and to the outcome of the interaction" (Bertelsen et al. 2007). This is a useful concept as it manages to transcend the common idea in HCI that interfaces should be transparent, that interaction should be direct (as derived from Shneiderman's (1983) direct manipulation), and that tools should be directly understandable and easily learned. This is obviously not the case with musical instruments. Such instruments would become boring very quickly; we saw in section 3.6.1 how Csikszentmihalyi (1996) explores this equation in his work with the concept of *flow*, where it becomes clear that an instrument needs to be challenging in order for people to keep up the interest. The musical instrument would have to constitute an equally distributed amount of skill and challenge. Too much challenge without skill results in anxiety, but too much skill without any challenge would result in boredom.

Bertelsen et al's observation is related to the focus of epistemic tools; i.e., that such tools should be analysed through the concept of instrumentness. Supporting this concept are layers of more finegrained properties analysed in the epistemic dimension space. We are therefore in a position to reflect on a more profound level on the question we ended the introduction to this chapter with: are digital musical instruments weak from an embodied perspective? Are they unsuitable for the pursuit of mastery? This chapter has attempted to answer these questions. It has been suggested that virtuosity in digital musical instruments shifts from being one of the body, to one of knowledge of mapping epistemic/virtual embodiment. Virtuosity in this domain means that we become masters of information processing and the understanding of arbitrary mappings in epistemically mediated embodiment.

This chapter has been concerned with the conditions of such instrumentness in musical instruments. The implications of making digital instruments, particularly screen-based instruments, have been studied. Through an extensive survey, a phenomenology of digital instruments was introduced, and the origins, or ontogenesis, of musical instruments were reviewed from a genealogical and actor-network theoretical perspective. For the purpose of clarifying the nature of digital instruments, in order to show their functions and fundaments, taxonomies relevant to the design of digital instruments were provided. The focus of this chapter has therefore been on the core of the digital musical instrument, its body, emphasising the epistemic side.

Chapter 6

ixiQuarks: Theory and Practice

This chapter introduces the ixiQuarks, a screen-based digital musical instrument aimed for live performance and improvisation. Much of the theoretical and empirical body of this thesis is based in the ixiQuarks, as it served as testbed for experimentation in design and user testing. The ixiQuarks are presented as epistemic tools, and an introduction to the thought process of designing a musical instrument is provided where the relationship between technological affordances and creativity is emphasised. Finally, surveys and user studies on the ixiQuarks are discussed and interpreted in the context of the theoretical issues described in part I.

6.1 Introduction

Part I covered the theoretical grounds for this project. It investigated the way in which the digital musical instrument serves as a carrier of musical ideology. This enquiry was conducted through an exegesis of the philosophy of technology and related discourses on technology [chapter 2]. As musical systems are both intended for compositional *and* performative work, it was relevant to study the embodiment status of digital musical instruments in comparison with acoustic instruments; how one learns and relates to digital instruments in terms of both their design and performance [chapter 3]. These two chapters led to the possibility of establishing the conceptulisation of *epistemic tools*, and here digital music systems were defined as a subclass of those [chapter 4].

In this chapter, I introduce and contextualise ixi software briefly; the ideology developed and implemented in these musical tools. More specifically, the ixiQuarks are presented. They are the software designed for this current research; based on them I have performed three different user tests and two surveys. The chapter provides a good ground to emphasise the importance of mapping in digital musical instruments. Mapping defined to be of both ergonomic (relating to phenomenology) and compositional (relating to epistemology) nature; something that can be in constant flux and dynamic, and this is the primary feature that defines digital musical instruments. The semiotic nature of mapping is explored in the context of digital musical instruments as epistemic tools. Finally, this chapter ends with a report on user tests that have been conducted of the ixiQuarks.

Section 1.3 introduced some of the fundamental ideas supporting the development of the ixiQuarks. With ixiQuarks, the aim was to create an environment that afforded a rapid construction of audio signal paths, the use of constrained instruments and support for livecoding or adaption of code in realtime performance. This expressive musical system was developed for normal computer/laptop setup that could be used as a testbed for questioning the *instrumentality*, or *instrumentness* (Bertelsen et al. 2007), of software based tools. In his PhD dissertation, Sergi Jordà asks if some instruments are better suited for improvisation than others (Jordà 2005: 203). His answer is affirmative. Below I will explore what constitutes a rewarding improvisational software-based musical instrument.

The instruments that exist under the generic rubric of ixi have all been designed from a specific interaction model that has been developed since ixi started in 2000 [see section 1.3]. Our experience is that once people become acquainted with the family of basic concepts and design ideas, they find it easy to move from one application to another. The ixi instruments are all based on the outlining of limitations from the same interaction model; their design is therefore very much a form of play with affordances and constraints. As opposed to acoustic instruments, screen-based digital instruments are not of physical material so all mappings from a GUI element to the sound are arbitrarily designed. This arbitrariness becomes even more prominent considering that there is hardly a tradition for creating such instruments. As seen in the interaction model described below, the metaphors used in ixi software are novel in the design of musical instruments. The determination to have no, or relatively few, musical references (such as depicting keyboards, strings, notes, etc) in the interfaces has reportedly liberated musicians from constraints they have felt with commercial software [see section 5.8.2.3]. This decision derives from the wish to get away from the cultural constraints that are connected to the traditions of historical instruments, encouraging the creation of new music by providing other sets of constraints than the classic musical structures (in both compositional and design terms).

Preceding chapters have elaborated on the nature of interface and interaction design as conceptual constructions. The interface and interaction design outlines the fundamental music theory of the instrument and its creative potential. They are the core of the musical instrument, the engine that defines its character. However, the question still remains of *how* we control those engines, whether it is to be done directly using some hardware, or through a graphical user interface, which presents a control layer between gesture (mouse) and sound. The concept of "embodiment" is central to this phenomenology, and has resulted in much research in tangible computing within the field of HCI (see for example Ishi & Ullmer 1997 or the research of the Tangible Media Group at MIT). However, as explained in earlier chapters, the very nature of the arbitrary relationship between gesture and sound means that there is no "correct" way of designing this coupling. This means that we have a situation where musical virtuosity in the context of digital instruments shifts from instrumental dexterity and sophistication, to a profound practical knowledge of programming, synthesis, mapping theory and electronics. Here, knowing the instrument involves the habituation of an epistemically mediated embodiment. The screen-based interface is an example of such mediated embodiment.[120] Below we explore the design of ixiQuarks as an instance of designing virtual embodiment.

6.2 Interfacing Sound: The Interface as Instrument

6.2.1 The Question of Mapping

The Theremin, one of the earliest electronic musical instruments invented by the Russian Leon Theremin in 1919, is probably the first instrument where the performance gesture does not involve a physical contact with the instrument itself. The Theremin exemplifies the problematics of mapping [explored further in section 7.1] as an important design feature of new musical instruments. Although the pitch and amplitude parameters of the Theremin are still controlled by physical laws – the nature electromagnetic waves, oscillators and antennae – the manner in which the electronics are wired defines the instrument. In digital instruments this mapping is arbitrary to a much higher degree and it presents itself as the most characteristic feature of digital musical instruments.[121] As the gesture-sound coupling is not implicit but has to be explicitly designed in digital instruments, mapping becomes one of the main elements in their design. Much research has already been performed in this field (Winkler 1995; Chadabe 1997; Hunt & Kirk 1999; Hunt & Kirk 2000; Orio, Schnell & Wanderley 2001; Hunt, Wanderley & Paradis 2002; Wanderley & Battier 2000; Wanderley 2002; Bongers 2000; Miranda & Wanderley 2006; Jensenius 2007).

After focusing on the nature of input devices for musical applications,[122] Wanderley (2001) explores two possible modes of mapping: a) the use of artificial neural networks to set up systems that learn through pattern recognition (Modler 2000; Collins 2006), and b) the use of explicit mapping in which three types of mapping are available:

> *One-to-one*: where one input element in the gestural controller is mapped to one parameter in the synthesis engine.
>
> *Many-to-one*: where many gestural features are mapped to one sonic feature.

---

[120] It should be noted that mediated (or epistemic) embodiment does not have to include this graphical representation between gesture and sound. The virtual mapping layer (see figure 6:1), is such a condition as well.

[121] One could argue that mapping is not arbitrary in digital musical instruments; that at the bottom of the whole system lies a microprocessor directing electronic currents, but this point ignores the fact that programmers typically do not design systems at that level. They are designed at the level of symbolic language.

[122] where there are four main features:
Learnability – the time it takes to gain control over the device;
Explorability – the extent to which the controller allows for exploration of sonic dimensions;
Feature Controllability – how the performer perceives the features of accuracy, resolution and range;
Timing Controllability – temporal precision being highly important in any musical instrument. (Wanderley 2001: 53).

One-to-many: where one gestural parameter is mapped to multiple parameters in the sound engine.

Certain authors (Jordà, Bongers, Chadabe) encourage people to see the digital musical instrument as a whole. This is a sensible suggestion to a composer or a non-technical performer who approaches a specific musical instrument for the first time. However, the explicitness of mapping is the reality of digital instruments, so a thorough understanding of mapping and the nature of the sound engine's parameters (such as Wanderley's above) are more productive for a proper understanding of the instrument. This is represented here in Figure 6:1 and discussed further in section 7.2.3. The composer should ideally be aware of the contingency of the mapping layer and the performer should ideally be able to adapt these parameters to the composition. Of course, we find that this situation of a non-technical composer and non- technical performer working with a digital instrument rarely happens. Most often, the composer is simultaneously the developer of the instrument and the performer of the composition, a situation in which it is often hard to differentiate between instrument design and composition.

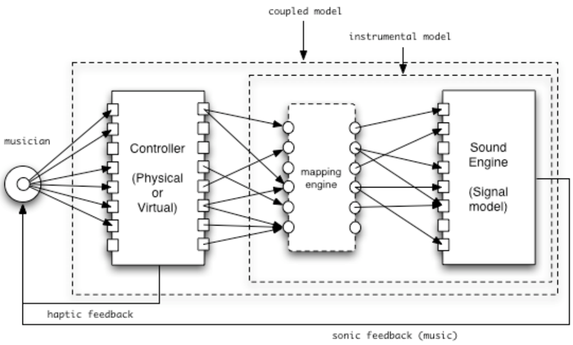

Figure 6:1. An illustration of the conceptual parts of the digital musical instrument

The design of a digital instrument typically involves compositional decisions as well. This leads to the definition of the instrument as primarily a combination of a sound engine and a mapping layer where the gestural interface is of secondary importance. Certainly, the interface affects what can be played and its affordances inspire what is composed, but considering the amount of symbolic knowledge (of synthesis and musical structure) that is inscribed into the digital engine of the instrument, I would like to emphasise the virtual nature of digital instruments and reduce the focus on physical interfaces. As I would further like to question the cultural values that found the design of virtual instruments, I propose a model of the digital musical instrument that emphasises the symbolic engine of the instrument, a model that could be described as that in Figure 6:1. This is the model used in the design of the ixiQuarks.

6.2.2 A Microhistory of Screen-based Instruments

In 1986 Laurie Spiegel wrote a graphical interface for MIDI control on the Atari computer called Music Mouse. It was an application that mapped two piano keyboards onto a two-dimensional grid and the performer would use the mouse/cursor to play the keyboard triggering MIDI notes. It may sound simple, but the functionality of this small application was more than just playing notes by moving a mouse. It had various scale and transposition settings; parallel and contrary motion, voicing, and grouping switches; loudness and tempo faders; 4 rhythmic treatments: chord, arpeggio, line, and "improvisational," and further parameters which are too specific to go into here. There were powerful features in this instrument where its logic took care of dividing spatial dimensions into scales (as opposed to raw pitches). The Music Mouse is therefore a good example of an epistemic tool, i.e., how creative musical decisions have been encoded in the instrument itself. One of the limitations of this application was obviously the "unifocal" nature of the mouse interaction, contrasted to the ten fingers playing a real keyboard, together with the lack of automation and memory of the player's behaviour.

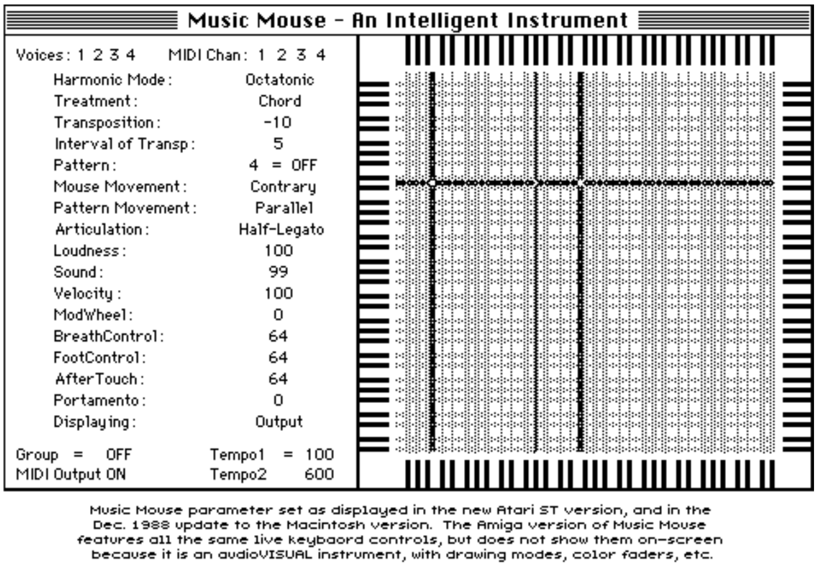

Figure 6:2. A screenshot of Laurie Spiegel's Music Mouse.

Andy Hunt's MidiGrid (Hunt & Kirk 2003) [see section 5.4] was another early approach in the creation of instruments with interfaces representing musical structures visually. The user plays a MIDI controller (such as keyboard or a sax) and the software receives the information and manipulates it. MidiGrid is built out of blocks where information can be stored and routed into other blocks and played. The software allows the user to build up increasingly complex structures by layering MIDI notes and

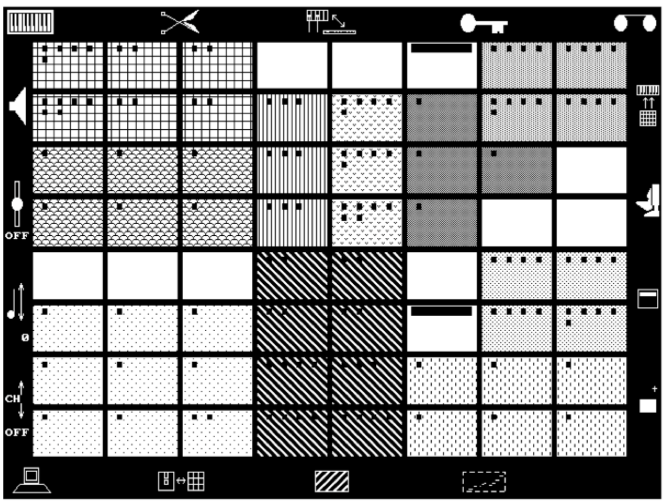

Figure 6:3 A screenshot of the MidiGrid

triggering routines represented with the graphical shapes of boxes. At the time of creation, MidiGrid was a highly inspirational software that attempted to take the field of computer music software into the domain of live musical performance. Limits of computing power and a premature market situation render MidiGrid a good example of technology that appeared too early for its time.

If we fast-forward into the late 1990s, Golan Levin's audiovisual performances (Levin 2000, 2005a)[123] represent a research field that arose with flexible programming languages and increasing computing power for both graphics and sound. Here, the two-dimensional screen space is seen as a realm where sound can be depicted as graphical objects. The user performs in this space through various interfaces (mouse, Wacom tablet, voice analysis) and the interface is often seen as an integral part of the performance, thus giving name to the style of "audiovisual performance."[124] Levin builds systems of audiovisual control that are both compositions and instruments. Visual forms are drawn on the screen, which result in the synthesis or playback of sound. The forms can be animated and sequenced in various ways, depending upon which instrument we are talking about. Levin's performances are impressive, but his instruments are highly personal and do not lend themselves to general use. This is where ixi's approach differs from Levin's. The ixi instruments are not seen as a personal expression of their creators, but rather general tools for musical exploration, albeit highly constrained and often eccentric. Again, it is hard to fit these things into a scale of black and white – they have to be placed on a grayscale continuum.

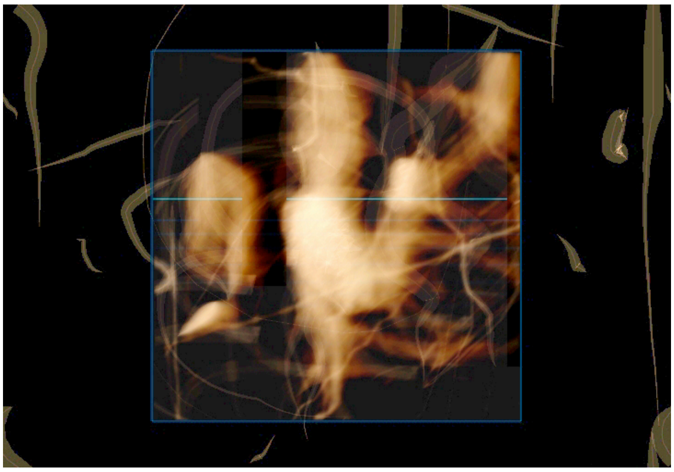

Figure 6:4. Golan Levin's Yellowtail audiovisual software

The situation today has improved greatly since the 90s considering the exploration and creation of virtual musical instruments or music systems of any kind. Open source programming environments, such as Processing, Python, OpenFrameworks, Pd/Gem, Max/MSP/Jitter, Flash, QuartzComposer, vvvv, etc., now enable the inventor of screen-based musical instruments to develop, quickly and effectively, musical instruments that build upon decades of research in musical interactivity and performance. With ubiquitous computing and the use of mobile devices, musical instruments now extend into people's daily lives when commuting, sitting in a

[123] See Levin's webpage: http://www.flong.com/

[124] It should be noted here that most often in "audiovisual" performances, the visuals are generated from amplitude or spectral information taken from the audio stream. Less common is the direction of visuals generating sound.

café, enabling modes of non-verbal communication. What has not been explored in-depth are the conceptual and ideological conditions of these projects, or rather, from the perspective of philosophy of technology, to what extent we are influenced or caught up in the network of an ever more complex web of blackboxed technologies that define our expression.

6.3 Interfaces as Semiotic Machines

In our work with ixi software, we have concentrated on creating abstract screen-based interfaces for musical performance, and have explored the affordances of the typical computer setup as a host for inspiring musical instruments. As already mentioned, these graphical interfaces do not necessarily relate to established conventions in interface design, such as using buttons, knobs and sliders; nor do they necessarily refer to musical metaphors such as the score (timeline), the keyboard (rational/discrete pitch organisation) or linear sequencing (such as in step sequencers or arpeggiators). Instead, musical structures are represented using abstract objects that perform actions such as moving, rotating, blinking, and/or interacting. The musician controls those objects as if they were parts of an acoustic instrument, using the mouse, the keyboard, or other control devices such as a pen tablet or custom made interfaces. ixi has created over sixteen software instruments, each exploring different modes of interactivity where some of the computer's unique qualities are utilised in fun, inspirational and innovative ways. By the "computer's unique qualities" I mean techniques such as remembering the musician's actions, following paths, interaction between agents, generativity, randomness, algorithmic calculations and artificial intelligence (such as rule-based techniques or artificial neural networks); all things that our beloved acoustic instruments are not very good at.

Over the course of our work, we have developed a loose and informal design language for these instruments – a semiotics that suggest to the musician what the functionality of each interface element is, and what it signifies in a musical context. Research on semiotics in Human-computer interaction (Andersen 2001; Beaudouin-Lafon 2004; Nadin 1988) typically focuses on the chain of meaning from the software designer to the software user. The user is the *receiver* of information and the aim of HCI is traditionally to make the interaction between the two systems (the human and the computer) intuitive, representational, and task oriented (based on real world tasks). Lacking is a stronger discussion of the situation where the computer is used as a tool for realtime artistic creation – an expressive instrument – and not a device for preparing, organising or receiving information. In artistic tools there is an important addition, where the signifying chain has been prolonged: a meaning is created by the artist (the user of tools), deploying software to achieve end goals, but this very software is also a system of representational meanings, thus influencing and coercing the artist into certain work patterns.

In her work on semiotic engineering, de Souza (2005) defines “intellectual artefacts” as tools that encode a particular understanding or interpretation of a problem situation, but also a set of solutions to that problem situation. This encoding is fundamentally semiotic. Ideally, both the user and the designer have to share an understanding of the same semiotic system. This shared understanding can only be achieved through a hermeneutic process where the behaviour or understanding of the user adapts to the model the designer has in mind. In this process the user will inevitably change her musical ideas according to the affordances of the instrument itself. This dynamic that the user of creative software finds herself in could be understood as a “dual semiotic stance,” with reference to Jakobson’s six-fold model of communication (Jakobson 1960). Here the idea is that the *sender* sends a *message* (that is expressed in a *code*) in a certain *context* to a *receiver* through some *channel*. Figure 6:5 represents Jakobson’s model, but it has been doubled here, such that it shows the dual semiotic stance of the user of creative software. On the left we have a representation of the instrument as a semiotic system, but on the right the music itself is seen as the meaning conveying medium.

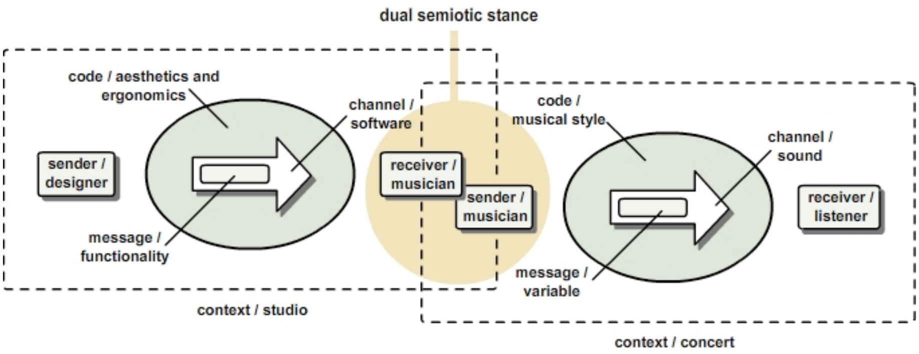

Figure 6:5. A representation of the dual semiotic stance of the user of digital musical systems

Chapter 2 surveyed various theories of technology and culture. Akrich and Latour’s (1992) metaphors of technology as script became particularly relevant when exploring the semiotic nature of computational tools. As described in chapter 4, there is a fundamental difference, although more of a degree than essence, between the nature of analogue and digital tools. In acoustic instruments, mappings, timbre and even musical affordances are practically given for free, whereas everything has to be designed in the digital instrument. The digital tool is a top-down construction where interface design, interaction design, mappings, synthesis and other parameters have to be explicitly determined through a meaningful system. It is therefore of a much stronger semiotic nature than the acoustic instrument and its analysis therefore requires a pronounced degree of hermeneutic interpretation. The sections that follow try to make explicit what kind of semiotics the ixi software makes use of. It should be noted that this is a post hoc account or post-rationalisation of design decisions that were taken in the flow of programming, where the musician-programmer and the programming environment interact and improvise through iterative creativity, one that is of a completely different nature than traditional software engineering [see sections 7.4 and 7.5 on creative programming, and Figure 7:1 on waterfall model of software design].

6.3.1 A Note on Instruments

Both acoustic instruments and music software incorporate and define the limits of what can be expressed with them. As covered in section 4.4.1.2, the struggle of designing, building and mastering an acoustic instrument is different from the endeavor of creating musical software. The acoustic instrument is made of physical material that defines its behaviour in the form of both tangible and aural feedback. These material properties are external to our thought, something that we fight with when we design and learn to play instruments. There is a degree of physical resistance in the materials used, which affect the instrumental performance (Evens 2005). These characteristics of the physical instrument are not to be found in software. Software is per definition *programmed*, its functionality is pre-written or pre-scribed by a designer or an engineer and the decisions taken in the design process become the defining qualities of the software, determining its expressive scope. In the words of James McCartney, the inventor of SuperCollider:

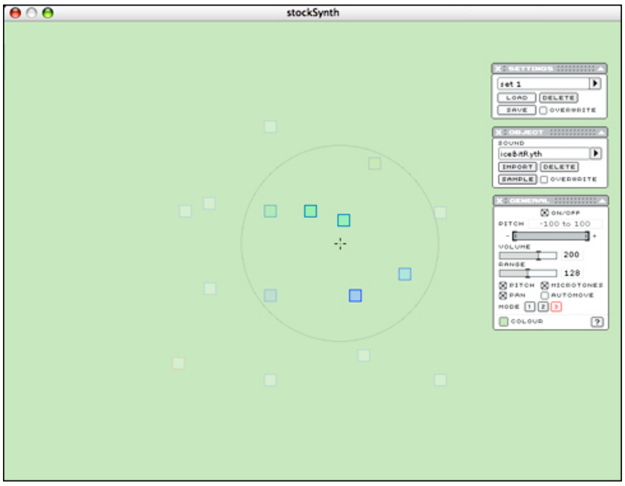

Figure 6:6. StockSynth. Here the crosshair cursor serves as a microphone that picks up sounds from the boxes that represent sound samples. The mic has adjustable scope (the circle). The boxes are moveable and the mic moves by drawn or automatic trajectories or by dragging it with the mouse

> Different languages are based on different paradigms and lead to different types of approaches to solve a given problem. Those who use a particular computer language learn to think in that language and can see problems in terms of how a solution would look in that language.[125] (McCartney 2003)

Here McCartney echoes comments made by Jean-Claude Risset stating that "any language for the description of sounds encourages certain types of manipulation and transformation and suggests trying these first" (Risset 1985) and Barry Truax arguing that "the structure and behaviour of music software provides a framework – a set of concepts and tools for their use – within which the music is conceived and realized" (Truax 1989: 155). We are therefore faced with an important question: what material (instruments) are computer musicians composing for

[125] Try to replace "language" with "instrument" in McCartney's paragraph above and we see that the same applies for musical instruments as well.

and where do they get the ideas from? We are reminded of Baird's material epistemologies explored in chapter 4. The question can also be phrased in other terms: where does the thinking (or composing) of the computer musician or digital instrument inventor take place? It happens most likely in the form and structure of the programming language or the software environment in which he or she is working. The environment defines the possibilities and the limitations of what can be thought. But what does it mean to "learn to think in [a] language"? What do we gain and what do we sacrifice when we choose an instrument or a programming environment? And why are people predisposed towards some environments and not others?

When musicians use software in their work, for example using StockSynth of GrainBox shown in Figures 6:6 and 6:7 respectively, they have to shape their work process according to the interface or structure of the software. As with the rigid body of acoustic instruments, software defines the scope of potential expression. Musicians are already tangled in a web of structured thinking but the level of freedom or expressiveness depends on the environment in which they work. To a great extent, musical thinking takes place at the level of the interface elements of the software itself:

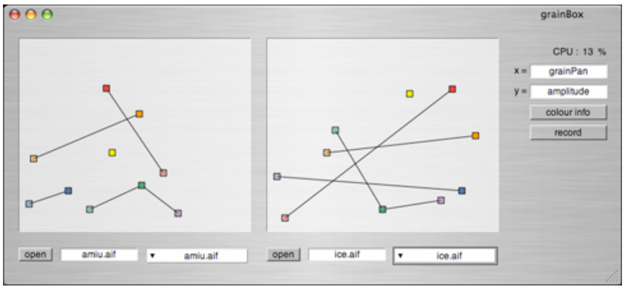

Figure 6:7. GrainBox. It can be hard to create interfaces for granular synthesis. The GrainBox is a suggestion how to represent the complex parameters as boxes with X and Y dimensions in 2D space and with connections to other parameters such as reverb and random functions

> It is misleading then to talk of thinking as of a 'mental activity'. We may say that thinking is essentially the activity of operating with signs. This activity is performed by the hand, when we think by writing; by the mouth and larynx, when we think by speaking; and if we think by imagining signs or pictures, I can give you no agent that thinks. If then you say that in such cases the mind thinks, I would only draw attention to the fact you are using a metaphor, that here the mind is an agent in a different sense from that in which the hand can be said to be the agent in writing.
>
> If again we talk about the locality where thinking takes place we have a right to say that this locality is the paper on which we write or the mouth which speaks. And if we talk of the head or the brain as the locality of thought, this is using the 'locality of thinking' in a different sense. (Wittgenstein 1969: 6)

If here I am attempting to find the "locus" of musical thinking/performing in both acoustic instruments and screen-based digital instruments, it is important to consider the difference in embodiment and incorporated knowledge of the player in those two types of instruments. When learning an acoustic instrument, the skill is incorporated through a process of habituation with the artefact: the learning "happens" through an interaction with the body of the instrument. Due to its material qualities, one can never master an acoustic instrument; due to its expressive openings it always contains something unexplored, some techniques that can be taken further and investigated. With software however, it is more or less a visual and procedural memory that is involved, and these are typically patterns of action that can be described explicitly in user-manuals or in help files.

6.3.2 Semiotics in Digital Musical Instruments

All design is essentially a semiotic endeavour. Designing a digital instrument or programming environment for music is to structure a system of signs into a coherent whole that incorporates some compositional ideology (or an effort to exclude it). The goal is to provide the users with a system in which they can express themselves and communicate their ideas. This has to happen in a way that suits their work methods, and sometimes provide new ways of thinking and working. The nature of the computer as an epistemic tool and a communicative, semiotic medium has to be explored.

6.3.2.1 Interaction Paradigms

Three primary interaction paradigms can roughly be defined in computer software as those of: *computer-as-tool, computer-as-partner,* and *computer-as-medium (*Beaudouin-Lafon 2004). Different research communities address these paradigms. The HCI field investigates the computer-as-tool paradigm but the attention is mainly on how to design understandable and ergonomic software for a receptive user of the tool. What is lacking is a better understanding of creativity itself and how creative people use software in their work (often misusing it to get their ideas across). We have learned from user feedback that there seems to be a general need for better sketching environments that can be modified according to the dictates of the user. An interesting fact here is that many cutting-edge art works are created by hacking or modifying software or simply creating new tools. There are schools of artists that respond to the limitations of commercial software with their own software in the form of software art.[126] (Cramer & Ulrike 2002; Fuller 2003; Gohlke 2003; Magnusson 2002; Shulgin & Gorinova 2004; Goriunova 2006).

6.3.2.2 The Semiotics of a Creative Tool

The most common of semiotic practices is to look at the signifying channel from the sender to the receiver through some medium such as signs, language, text, or film. (An analysis splendidly represented in the work of Barthes (1972) or Eco (1976)). The "work" here is a static

---

[126] The www.runme.org repository is an excellent source for information and examples of what is happening in the field of software art and generative art. It is closely related to the ReadMe festival, which was the first software art festival.

construction that does not change after it has been published or released.[127] By contrast, computer-based works are interactive and can be changed or modified after their release either by users themselves or by updates. Interaction becomes a new sign-feature (Andersen 2001). It is not merely the interface elements that contain semiotic weight and interpretative significance, but also the interaction processes that are often heavily metaphorical or metonymic [see taxonomy in section 5.5.5]. Some studies have been done on this new semiotic quality of the computer (Andersen 1992; Andersen & May 2001; Andersen 2001; Beadouin-Lafon 2000), but very few in the field of music software or other creative software.

In music software, the user is simultaneously *the receiver* and interpreter of information from the designers of the software, and *the sender* of information in the form of the music being composed using the tool. This dual semiotic stance (see Figure 6.5) is important in all tools (whether real or virtual) but becomes vital in heavily theoretically inscribed epistemic tools, such as music software. Music software is a sign system in its own right, but the important question here is: which are the relevant layers of signification and communication and from where do the originate? This could be analysed into strata of different practices, and we might identify the hardware designers, the programmers of the compilers, the language API and the software itself, the designers of the interaction and the programmers of the interface. A creative tool has a history of important design decisions all shaping its scope and potential. This is a complex structure, but the user is faced with the question: what is the meaning conveyed in the interface? And is this system of signification not essentially of compositional nature? Who took those decisions and by which criteria?

The contingency of design mentioned above in relation to the digital medium is one of its most definable characteristics. We do not have this “contingency problem” when designing acoustic instruments as the properties of the material we work with leads us in our design: closing a hole in a flute increases the wavelength in the resonant tube and the tone deepens; pressing the string against the fingerboard of a guitar – shortening the wavelength – produces a note of higher pitch. When designing screen-based computer interfaces we can choose to imitate physical laws as known from the world of acoustic instruments, we can draw from the reservoir of HCI techniques or we can design something entirely new. It is here that interface design, the interaction design, and mapping become important factors in the creation of interesting screen-based instruments for the computer.

6.3.3 Interface Elements in ixi

Most modern operating systems are graphical or allow for a graphical front end. The WIMP (Window, Icon, Menu, Pointer) interface (Apple Computer 1987; Microsoft Corporation 1992)

[127] Post-structuralist thought has rightly pointed out how interpretations of the work change in different times and cultures, but the work itself does not change - only people’s interpretation and reception of it.

has become a standard practice and users have become accustomed to the direct manipulation (Shneiderman 1983) of graphical objects. The most common approach is to translate work practices from the real world into the realm of the computer: thus we get the folders, the documents, the desktops and the trash cans, all representing the office that "we" are all used to.[128] In music applications there are representations of keyboards; buttons, knobs and sliders; rack effect units and cables. This is suitable where the aim is to translate studio work practices into the virtual studio of the computer. But when the aim is to create new instruments using the new signal processing capabilities and artificial intelligence of the computer, there might not exist any physical phenomena that we can use as source for our interface metaphors. The method used in the current work is to aim at minimising references to established practices in the design of metaphors that copy physical artefacts, and rather explore what the computer affords in these terms, in particular with the use of metonyms: strategies for interface and interaction design.

6.3.3.1 Interaction Models

Each of the ixi applications are experiments and they explore a specific mode of interaction. Taken as a whole, our software can be seen as using a specific type of an *interaction model*: a language, a semiotic, or a design ideology that informs and en-forms the work. An interaction model can be defined as more operational than an interaction paradigm (computer as tool, partner or medium) (Beadouin-Lafon 2004). The model can be assessed according to its descriptive, evaluative and generative powers. These dimensions of evaluation are all important when creating an interaction model. The *descriptive* power is the ability to describe a significant range of existing interfaces; the *evaluative* power helps us to assess multiple design alternatives; and the *generative* power is the ability of the model to inspire and lead designers to create new designs and solutions.

The interaction model of ixi software can be formalised and concretised in the following bullet points:

- sounds have visual representations in the form of static or active objects
- actors (rule-based, stochastic or artificial life) can have sonic properties
- spatial properties of the screen are mapped to sonic properties
- objects are dynamic (change size, colour and shape; rotate, move)
- use of metaphors and metonyms from already established practices [section 5.5.5]:
    - human-computer interaction
    - computer games
    - new media/web design

128 Here research programmes that deal with the cultural pertinence of HCI, such as ethnocomputing, observe that only certain sectors of society work in offices and many societies do not even have office cultures.

  - physical actions (such as plucking, stroking, scratching, shaking, etc.)[129]
  - audio channelling through busses (found in mixers and guitar pedals)
- physical manipulation of sample data (e.g., SoundScratcher) through drawing
- non-linear and non-deterministic pattern generators (for example rhythm)
- incorporation of livecoding as realtime performative musical action
- emphasis on microtonality and alternative scales
- explorations of polyrhythms through adaptable software
- modularity (output of one instrument can be used as input in another)
- support of alternative controllers (Wacom tablets, MIDI, now OSC and Muio, soon Wiimote)
- openness to communication with other software or hardware through protocols such as MIDI or OSC

We have aimed to refrain from:

- simulations of hardware
- buttons, knobs and sliders
- simulations of acoustic instruments
- emphasis on timelines (impedes improvisation)
- rigid or linear rhythmic structures
- the necessity of having MIDI controllers or other hardware. The software should run with the most basic setup of screen, speakers, keyboard and mouse.

Many of these points are explained further in section 6.3.3.3. It is important to note that this interaction model rejects a direct simulation of the physical realm, but focuses instead on the affordances and integral qualities of the computer itself as a symbolic, number-crunching, meta-machine. However, although simulations are rejected, inspiration is also derived from physical *activities*, and those are added to actions that have become established in the reservoir interaction design through computer games, new media and other symbolic interactive applications.

6.3.3.2 Interaction Instruments

The generative aspect of ixi's interaction model relates to Beaudouin-Lafon's definition of *instrumental interaction* (Beaudouin-Lafon 2000). The interaction instrument is a tool that couples the user with the object of interest. A scrollbar is an example of such instrument, as it gives the user the ability to change the state/view of the document *through* the use of this representational tool. A pen, brush or a selection tool in a graphics package are also good examples. The interaction instrument is the locus where the user of the virtual tool focuses in order to perform some task, but the instrument itself disappears in its use.

There are three design principles that define the methodology of instrumental interaction: *reification* – the process by which concepts are turned into objects; *polymorphism* – the property that enables a single command to be applicable to objects of different types; *reuse* – the storing of previous input or output for another use. In the work of ixi, reification has been

[129] It should be noted that our design tries to refrain from copying physical hardware or traditional acoustic instruments, but we find metophors of physical actions inspiring in use.

used extensively in a method we could call "interface poetics," i.e., where, especially since simulations are rejected, metaphors and metonyms are used to create novel interactive modes. Polymorphism is used where interface elements/objects are mappable to various actions in the sound engine. The user is able to direct the meaning of the pattern generating interface to different results. Finally, in ixi, reuse is made available when the storing of gestures and settings is allowed, or by the reuse of sound itself (as a material flowing through audio channels/busses), where one instrument can make use of the sound stream coming from another instrument.

Any interactive instrument or element in the user interface is learned and habituated by the user. This happens through a "generate-revise" interpretative process that can be explained by resolving to semiotics [see section 4.3.2.6]. Peirce, for example, called this phenomenon *abduction* (Peirce 1958). Abduction is a process that happens through time and is unlimited, in the sense that its meaning is never fully determined. Therefore, the interaction model presented above should not be seen as a rigid rule model, but one that should rather be viewed from Wittgenstein's language philosophy of family resemblances and meaning constituted through use.

6.3.3.3 The Terminology of ixi's Semantics

In working on *ixi software*, the focus has been primarily on the interaction design as opposed to the interface design. Although semiotics and general "look" are important, the interaction structures are primarily of interest. The design of interface elements is often highly (but not exclusively) aesthetic and depending on taste, whereas the interaction design deals with the fundamental conceptual structure and ergonomic capacity of the software and depends more on work preferences. For example, in SpinDrum, the wheels contain rotating petals determine the rhythmic cycle, the size of the wheel signifies the volume, and the colour accounts for which sound is attached to the object. Here the interaction design clearly influences the interface design (size, number of petals, colour), but the shape of the petals (whether a square, a circle or a triangle) is simply an aesthetic decision and of little musical importance although it affects the general aesthetics of the tool and the feeling its users have. The interaction elements of ixi software can be defined as actors, context and networks. These will be explored here below.

6.3.3.3.1 Actors

The ixi interfaces are pattern generating machines that are mapped to sonic parameters. Due to the emphasis and flexibility of the mapping, there is a separation between the machines' generativity to their sound generation. To sum up the basic design ideas of ixi software we could say that it was the *reification of musical ideas into graphical objects as open control* *mechanisms that act in time*.[130] We call these abstract objects *actors,* as they are graphical representations of temporal processes that act, enact and react to the user, to each other, or the system itself in a complex network of properties, relations and teleology (desired states or end goals). From the perspective of user focus, the interaction instrument described by Beadouin-Lafon is unifocal, i.e., when the user interacts with it, nothing else is happening in the software. This is typical for word editing software, spreadsheets and other production tools. These tools are *ergonomically* "single-threaded" or serial task-processing applications used for painting, text editing, programming, video editing or in architecture. In contrast to these applications, a musical tool is typically multi-threaded or parallel, i.e., there are many processes, streams, layers or channels that run concurrently in every composition or performance, all controlled by the user albeit typically not in a direct mapping.[131]

The interface units that we call actors – such as a picker, a spindrum or a virus – are not instruments that the musician uses for some task and then chooses another instrument for the next task. The actors in the ixi software applications are put into action at some point in time and they continue working in a temporal flow (rotating, moving through a trajectory or interacting) until the musician decides to stop or pause their activities. The musician is therefore both conductor and performer in the same stroke, delegating some of the musical actions to virtual actors.

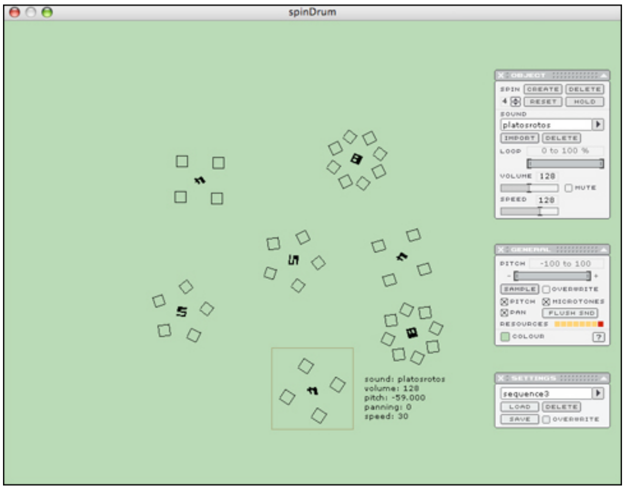

Figure 6:8. SpinDrum. Each wheel contains from 1 to 10 pedals. The wheels rotate in various speeds, and when a pedal hits top position (12 o'clock) it triggers the sample or sends out OSC info to the soundengine. The X and Y location of the wheels can affect parameters such as pitch and panning

This idea of actors as all the elements that affect the interaction in an instrument, can be illustrated by looking at the software *Connector*. Here actors move in a system of connectors (a plumbing-like system) and trigger sound samples or MIDI notes that are properties of the connectors. The connectors are actors themselves as they are the receivers of an action and contain the information that triggers the sound. It is through the interaction of all the actors and their properties that interaction takes place – interaction between elements *within* the instrument and also with the musician using the instrument – and this interaction is simply the automation that controls the various parts of the music set into motion. For example, in *StockSynth* (Figure

[130] Musical idea here meaning any pattern generating structures.
[131] This relates to another fact that divides those types of software is that the painting software, the video software or the 3D package are not packages that are used in live performance.

6:7) the microphone is one such actor (with its properties of trajectory and scope) that interacts with the sound objects that contain the information about the sound and its properties.

6.3.3.3.2 Environmental Context

All actors perform their task in a context. This context is the environment in which they exist, what parameters the space contains in which they live, and how they are affected by them. The actors are graphically represented in a two- or three-dimensional space and their location, interaction with other actors, interaction with non-acting environmental features, etc., might typically influence their properties. The space can have qualities such as temperature, gravity, brightness, etc. which are all qualities that could affect the behaviour of the actor or it can contain other actors of different type that influence the behaviour of the message sending actors. The actors move, rotate or blink in this space and are therefore both spatially and temporally active units. Feedback from users of ixi software has shown us that people find the metaphor of an actor presented in time and space useful to represent musical actions and ideas. What the survey data also indicates is that people intuitively understand the metaphor of having actors on a stage that perform some tasks that they, as the operators of the musical systems, control.

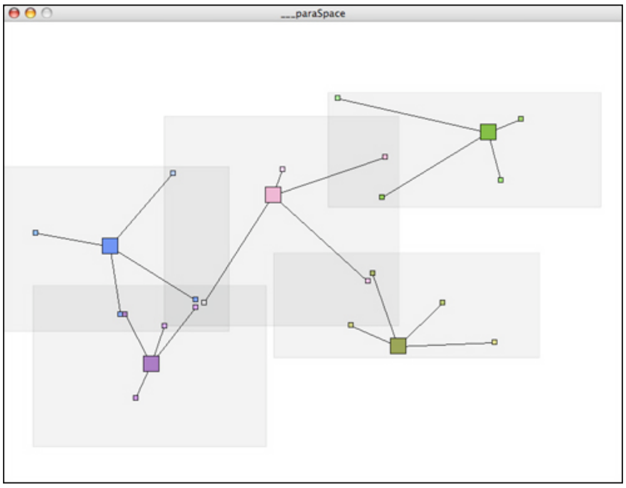

Figure 6:9. ParaSpace. This application interfaces with audio effects written in SuperCollider (but can talk to any software that supports OSC). Each audio effect has variable number of parameters and they are represented as small boxes in the control interface of ParaSpace. The point here is that the parameters interact on the interface level with automation, artificial life and artificial intelligence.

6.3.3.3.3 Network

When talking about the context and the environment of these actors, it should be noted that the interface elements are not the only actors: the user is also an (inter)actor of a specific kind, the control hardware (a mouse, keyboard, sensor or controller), the soundcard, the speakers and other communication such as virtual audio cables, MIDI or OSC messages. The whole context of musical action and reaction is the space of the actor, a space in which the heterogeneous network of musical performance takes place. [132]

[132] It should be noted here that this talk about actors and networks is only indirectly related to the actor-network theory. The actors are obviously made of networks of other actors (or parameters) and the

The network is therefore defined by the context in which ixi software is played; how the user establishes it as a performance tool in either live performance or studio use. This important factor has been studied, for example in an anthropological and ergonomic approach to "performance ecosystems" (Waters 2000, 2007; Bowers 2002). The focus on the network includes the performer, the instrument and the environment in which the instrument is played. The use of a digital instrument is necessarily bound in a social context and in a specific culture of technological use and approval. The performance of the instruments (and often their assembly or modification in realtime), the environment, and the performer's interaction with the audience is here seen as one intricate dynamical system.

6.3.3.3.4 Semiotic elements and mapping

The actors and the contexts in which they function are all elements of a semiotic language. This language has dialects, or rather idiolects (each application is unique), where the meaning of an element can change as in Wittgenstein's concept of the usage as the word's meaning (Wittgenstein 1968) or as in the Saussurian conception of the lack of natural connection between a signifier and the signified (Saussure 1983).[133] In the interaction model developed in the ixi applications, a semiotics is provided, and language games are suggested, where the behaviour of an actor maps onto some parameters in a sound engine. For example, vertical location of an actor could signify the pitch of a synthesised sound or the playback rate of a sample. Size could mean amplitude, rotation triggering, and its direction could mean a tendency for some action. But, considering the open mapping of the interfaces with OSC messages sent out to any sound engine available, it could also signify something entirely different as the controllers are open and it is up to the musician to map the actor's behaviour onto parameters in the sound engine.

6.3.4 Conclusion

This section has attempted to show how the input gesture, the graphical representation of sound and actions, and the complex mapping engine that couples gesture to sound, are the materials we work with when designing digital instruments. Whereas the fundamental scope of acoustic instruments is highly dependent upon the physical material they are made of (wood, iron, strings, etc.), the situation is very different when we create digital instruments, in particular when they are screen-based. This terminology of actors, context and network is presented in order to better understand and modularise the interaction and interface design of virtual

---

general network is also made of various other actors, but it is more of a serendipity than anything else that these terminologies are so similar.

[133] For Saussure, the meaning of signs is relational rather than referential, i.e., the meaning lies in their systematic relation to other signs in the semiotic system and not by direct reference to material things, for example.

instruments. This also implies the possibility of an interface having its own meaning system independent of its relationship to the sound-engine, where the interactive patterns of an instrument can be mapped in many different ways onto the parameters of the sound-engine. Such mapping is always of a compositional nature.

6.4 ixiQuarks: Theory and Practice

6.4.1 Introduction

After this general discussion of the ixi software above, the focus will now be directed at the ixiQuarks as an environment for live performance and audio programming. As mentioned earlier, I developed this software directly as part of the current research in order to explore interactive models, mechanisms of mapping, cultural and aesthetic issues of electronic music performance, but also to serve as an experimental object for user studies and surveys that investigate the aforementioned topics.

The ixiQuarks appear as part of a movement in which, in order to escape from the expressive constraints of commercial or closed source software, musicians increasingly make use of free and open source programming environments such as SuperCollider, ChucK, or Pure Data. Each of these environments have their own affordances and constraints from musical perspective and allow for either textual or graphical programming. The word "free" above connotes not only free as in "free speech," or "free beer," but also free as in "free jazz." The freedom of musical expression when utilising these tools is characteristic of their nature and important criteria their designers kept in mind when designing the environments (McCartney 2002; Puckette 2002).[134] Of course, it could be argued that these audio programming languages are constraining as well, but this is more likely to be at the level of software design and less at the level of musical composition, i.e., it might be difficult to realise some specific musical ideas, but that might be due to the way the language is structured. Virtually everything is possible, but the language leads to particular approaches which means that the system has to be fought at times. Although requiring varying amount of effort, the musician or instrument designer typically has ways of getting around software engineering limitations to reach the goal of the composition or instrument design.

The ixiQuarks software suite was released in June 2007 and has had around 37000 downloads up until March 2009. There are around 1600 users who have registered with us and agreed to participate in user surveys. Of those, 197 people have filled in an extensive internet survey on the ixiQuarks. Furthermore, in-depth user interviews have been administered with both experienced and novice users of ixiQuarks. From this user testing, surveys, interviews and workshops we are able to identify certain defining positive and negative aspects in the use of

[134] See also section 7.5 where I report on a survey made with audio programming language designers.

digital tools as musical instruments. This data is used to extract design strategies for the designer of digital musical instruments, further supporting the theories of epistemic tools and virtual embodiment.

In this section, I will introduce the ixiQuarks as a graphical user interface (GUI) environment of audio tools and instruments for live improvisation that allow for user interaction on GUI, hardware, and code level. The ixiQuarks are written in the SuperCollider programming language and are part of the Quarks repository (3rd party external libraries for the SuperCollider language). I will commence by outlining the general SuperCollider/Quarks environment, then introduce the ixiQuarks instruments and finally talk about the philosophical and aesthetic implications of this unified creative space where code and interface can merge in a continuous performance activity.

6.4.2 The SuperCollider Environment

SuperCollider 3 (or SC Server) (McCartney 2002) is the state-of-the-art audio programming environment of today, written by James McCartney and released as open source software in 2002. Since then, it has become ever more sophisticated and powerful, used by musicians, artists and scientists alike, forming a strong developer and user community. SuperCollider is split up in two independent parts, the SC language and the SC server. The former is an interpreted, object orientated language written in C/C++ that takes inspiration from the design of SmallTalk; the latter is an audio server that supports a powerful C plugin architecture which makes audio digital signal processing effective, fast and easy to program. The language and the server communicate through the Open Sound Control (OSC) protocol (Wright 2003), which makes it possible for all OSC supporting programming languages to talk to the SC server. This split between a language and a server, and the usage of OSC makes SuperCollider an ideal audio programming environment for networked performances, livecoding, interface creation (hardware and software) and collaborative playing. It also means that the language can perform complex realtime calculations without resulting in glitches in the audio as they are two separate processes.

The SuperCollider language and server are open source and anyone can write class extensions for the language or plug-ins (unit generators) for the server. The environment is compiled into a binary file (the distributed application) with a C compiler, but then runs as an interpreted language that has source class files that interface with the C primitives. The SuperCollider class files are written in the SC-language and are an important way to modularise code and compositional concepts when working with it. SuperCollider itself is an extremely broad, flexible and expressive language, easily extendable by writing one’s own classes.

6.4.3 The Quarks System

Authors of 3rd party SuperCollider classes tend to share them with the rest of the community if they have a general scope and are useful in other than a private capacity. In order to extend SuperCollider with new classes, users can either install them manually into the appropriate ClassLib folder or use the dynamic Quarks system. The Quarks system was introduced for the creators of 3rd party class files to simplify the process of creating, updating and distributing their code but also simplifying the updating channel for the users of the classes. This is achieved by using a SVN[135] (Subversion Control) repository system, where authors commit the latest changes in their classes, enabling users to update their classes with a simple two line command:

```
Quarks.checkout( "ixiQuarks" ); // downloads from svn
Quarks.install( "ixiQuarks" );    // installs in sc-classpath
```

The repository is online on the SourceForge website and the Quarks class in SuperCollider takes care of downloading and installing the chosen classes "under the hood" so to speak. The user only needs the two lines above to download new class libraries. The SVN system makes it easy for the author to organise development track changes in the code, but also for the user to follow the changes in the class. As pointed out innumerable times in this thesis, the digital instrument is never finalised, code is dynamic, and a class in a programming language is always a dynamic entity.

6.4.4 Programming and Livecoding with SuperCollider

One of the frustrations for the computer musician performing in a live situation with acoustic instrumentalists, for example in an improvisation band, is the difficulty of carrying out spontaneous and intuitive change in playing [see section 6.1]. Musical software is often more focused on the score or the textural (i.e., timbral) level of a musical performance rather than on the note level. This fact tends to make software instruments less expressive than their acoustic counterparts if conceived of as instruments to be controlled by human gestures in a live performance. But there are various ways to get around the rigidity of musical software. One solution is to develop gestural musical interfaces that aim at instrumental usage (a field represented by the new musical interface research, for example the NIME conference series) but another approach is the practice of livecoding where the instrument/music is created and modified as a performance act (a style of performance well represented by TOPLAP).[136]

SuperCollider has extensive support for creating interfaces on the graphical user interface level, using MIDI, HID (Human Interface Devices), serial or OSC (Open Sound Control) communication. As such it lends itself well to all common interface work, instrument

---

[135] http://sourceforge.net/docs/E09

[136] See www.nime.org and www.toplap.org respectively.

making and installations. But SuperCollider is also one of the most powerful environments for livecoding musical performances (Rohrhuber et al. 2005; Collins 2007; Ward et al. 2004) as it is an interpreted language allowing for new code to be evaluated in run-time without saving or recompiling the environment. Programs can be created that evolve or change according to user input or additional programming. As an example, below is a small JITLib (Rohrhuber et al. 2005) program that generates a simple snare sound every second:

```
~trig = { Impulse.ar(1) }; // the trigger
~snare = { WhiteNoise.ar(1) * EnvGen.ar( Env.perc, ~trig.ar )};
```

Say we wanted to add a low pass filter to the sound without interrupting its temporal continuity, we simply run the latter line again, now with the white noise going through the filter:

```
~snare={LPF.ar(WhiteNoise.ar * EnvGen.ar(Env.perc,~trig.ar), 2000)};
```

There are countless ways of doing these things in SuperCollider, since it is a wide and powerful programming language whose users have different agendas, emphases and programming styles. Some people prefer writing classes and/or graphical user interfaces to be controlled by sensors or controllers. Others work purely in code composing algorithmic music and yet others enjoy the tense experience of coding in front of the audience in a live situation.[137] In contrast to much closed source software, there are as many ways of using SuperCollider as there are people working with it. There is no rigid methodology as all programmers/musicians have their own way of thinking; their own style of writing code/music.

6.4.5 The ixiQuarks: A Subset of SuperCollider

Recently the interface support of SuperCollider has matured to the level that interfaces in the style of ixi software can easily be built with the SuperCollider language itself.[138] As SuperCollider is my programming language of choice, it became more natural to write the interfaces in SuperCollider itself, rather than in Python or Java as previous work with ixi had done. The interfaces are still OSC controllers that can be used with other sound engines as well, but they are now streamlined for use with the SC audio server.

6.4.5.1 The ixiQuarks Environment

SuperCollider is an open and dynamic environment that allows for running many programs simultaneously, using any number of groups, nodes and audio busses. Any process can be started, paused, stopped or freed without interfering with other processes that are also running in

[137] See the TOPLAP manifesto - http://www.toplap.org/

[138] What was needed was a class that detected mouse movements, drawing functionality and hardware interfacing.

the environment. The ixiQuarks toolbox is a collection of tools that perform various tasks that could be time-consuming to code up in a live/improv situation, but easily accessed from a GUI window that contains a list of all the ixiQuarks. The ixiQuarks do not need to be used exclusively as an independent environment, but can be used with any other program written in SuperCollider. As an example one could imagine a performer that is running some process using the Pattern classes, suddenly deciding to add reverb to the output. In this case it is trivial to open a reverb ixiQuark and route the audio from the original process through the reverb widget in ixiQuarks that provides a simple GUI to control the basic parameters.

The ixiQuarks is a modular environment that consists of three different types of tools: *basic utilities*, *audio effects* and *instruments*. The environment is built around audio busses that can be used to patch audio streams into one another. An audio bus can contain the output of many sound-generating processes. Below, the first two types will be explained briefly, and then the focus will move onto what is more relevant to this research: the instruments.

6.4.5.2 Basic Utilities

The basic utilities are tools such as AudioIn, Recorders (of any audio channel), BufferPools (that stores sound buffers in RAM), Players (streaming soundfiles from the hard disk), NodeMixers, and various scopes for viewing the audio (such as an EQMeter, FreqScope, WaveScope, Spectrogram, etc. See Figure 6:10). These are general utilities needed in order to rapidly set up an improvisational environment. The design idea is to let the instruments make use of these utilities rather than integrating them into the instruments themselves, which would create unnecessary complexity and be against the modular design philosophy of the environment. For example sound samples can be loaded into a bufferPool, which is independent of the instruments, but allowing all instruments to access their data. Resources are thus shared in order to save memory and processing cycles.

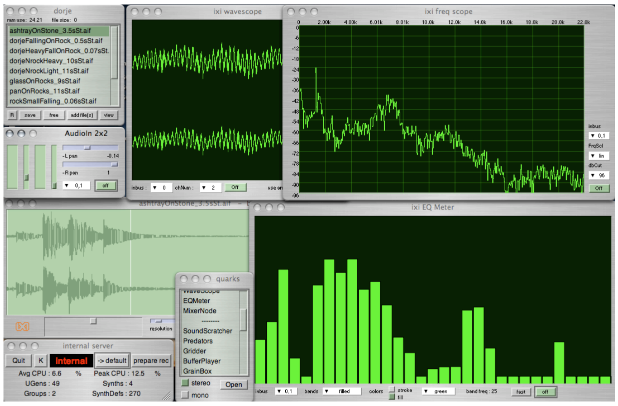

Figure 6:10. A screenshot of some of the ixiQuarks utilities

6.4.5.3 Audio Effects

The audio effects are the typical effects known from most sound editors: delay, reverb, distortion, compression, chorus, flanger, tremolo, equalizer, vocoder, randompanner, and some non-conventional and strange effects such as MrRoque (which is an effect that records incoming sound with reverb and succinctly plays it backwards, also through a reverb – see Figure 6:11), or the cyberpunk. The effects run on the audio channels and can be turned on and off as one wishes. The user can plug many effects onto the same channel or route the sound through one effect into the next on another channel.

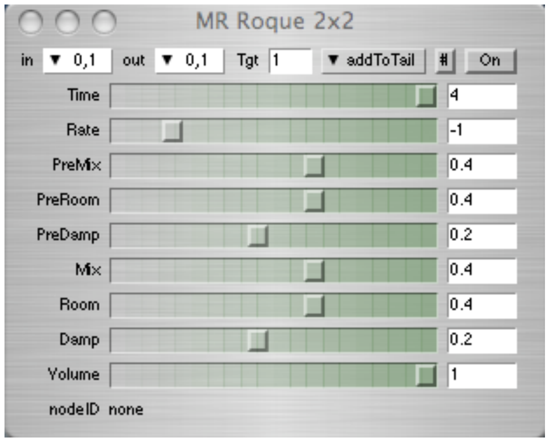

Figure 6:11. A screenshot of a typial ixiQuarks audio effect.

6.4.5.4 Instruments

The ixiQuarks environment is designed for the building of instruments that make use of other ixiQuarks such as the utility tools or the audio effects. At the point of writing there are twelve different instruments available with more in the pipeline, and users can easily write their own instruments that work seamlessly in the environment. In general these instruments are pattern-generators that allow for sample manipulation, synthesis and livecoding. Below I explain five of them.

6.4.5.4.1 The SoundScratcher

The SoundScratcher (Figure 6:12) is an instrument that represents the waveform of a sample buffer in a graphical display. Any buffer stored in the RAM memory can be manipulated by the instrument. The instrument receives input from hardware such as the mouse or a Wacom tablet (where pressure is mapped to amplitude) and the basic idea is that the user can draw on top of the waveform representation and control the playback of the sound that way. There are various interactive modes in which the user can play with the sound sample:

- *warp* – a granular synthesis where the location in the sound file is played constantly using overlapping soundgrains. Here the vertical location of the cursor represents the pitch.
- *scratch* – by moving the pen (or some other sensor input) the user can scratch the buffer forwards and backwards like a needle on a turntable. In the spirit of turntables, the speed of the gestural movement maps naturally to the pitch.
- *random grains* – here grains are represented as dots on the canvas. The sound engine reads randomly though an array of the locations of those dots, creating a granular cloud. The density of the cloud, the envelope length and envelope type of each grain can be controlled from the interface.
- *linear grains* – same functionality as in random grains, but here the sound engine reads linearly through the list of grains. However, there is a randomising button on the left of the interface where the users can randomise the list if they so wish.

- *worm* – the worm is a creature that moves over the space of the sound with variable numbers of grains in its spine. The speed of the worm and the grain duration can be controlled.
- *graincircles* – these are circles of variable size and amplitude (represented as alpha in colour) inside which the sound engine spawns grains according to the set speed. Again the envelope and duration of the grain can be controlled.
- *grainsquares* – as opposed to the graincircles the squares always play the grains from the left point of its location. This makes it easier to create interesting rhythms and periodic sound textures than in the random space of the circles. The speed of the grain repetition can be affected in both the graincircles and the grainsquares by pressing the 1, 2, 3, or 4 number keys, representing the respective time relationships (3 against 2 or 4 against 3, etc.)

The gestural movements of the user can be recorded, stored and played back by the instrument itself. That way the user can draw patterns on the instrument and leave it to perform on its own. The user could then for example open up another instance of Soundscratcher to perform with.

Figure 6:12. A screenshot of SoundScratcher

*6.4.5.4.2 The Gridder*

The Gridder is an instrument that focuses on micro-tonality, tuning and scales. It consists of a scalable grid of nodes (from 5 to 48 squared) which is mapped in equal temperament (implementing the formula *fundamental * 2.pow(i/steps_per_octave))* in octaves (one octave per horizontal line) but wrapping at a set ceiling frequency.

Like SoundScratcher, the Gridder is a broad conceptual environment on its own that affords various interaction modalities. We could divide the analysis of the instrument into two parts: the *environmental part* – where we look at the acoustic properties of the instrument; and the *interaction part* – where we focus on how the instrument can be controlled.

Looking at the *environmental* properties of the instrument [cf. section 5.3.3.3], the most important part is the definition of the pitch resolution in an octave. The grid can be scaled from a 5-TET scale to 48-TET scale. (5-TET = 5 tone equal tempered tuning). Each node on the grid is a note in the scale. The horizontal axis represents the N notes in the octave and the vertical axis is the next pitch range, defaulting to an octave above, but it could be the fundamental scale transposed by a 3rd, a 5th or any other tonal relation. There is a setting that controls maximum frequency on the vertical axis forcing the scale to wrap back to the fundamental note when the ceiling is reached. The user can define scales in any of the pitch resolutions by drawing vertical lines as represented by the columns of boxes in Figure 6:13 and these can be stored in a dictionary. There are 8 different types of synthesized sound that can be selected, and users can also write their own synthesis using code in a special coding window. The nodes are not limited to synthesis, they can also contain sound samples that are triggered when the node is activated. The piano keyboard view is optional and not graphically connected to the grid itself in any way. It shows the notes played, indicated with grey if the pitch maps to the western 12-tone scale but red if it is a microtone.

The *interactive* attributes (related to playing) of the instrument consist of a space where one can play scales or notes with the mouse, the pen tablet or any other interface. There are two types of playing: free playing, where the player can play any note on the grid; and restricted playing, where only the selected notes can be played. That way the instrument can be played as a custom-tuned string instrument. Related projects that implement some of this functionality are the Music Mouse (Spiegel 1993) and MidiGrid (Hunt 2003).

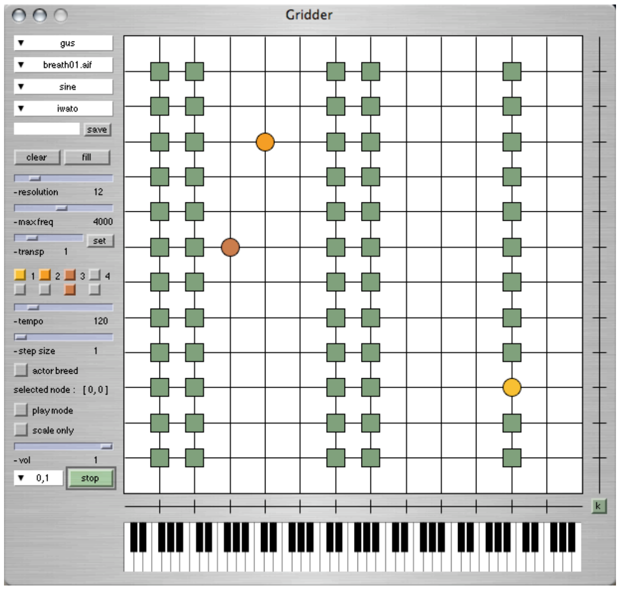

Figure 6:13. A screenshot of Gridder in 12-TET mode

The interface also contains 4 actors that can move through the grid-space, each with its own tempo and step size properties. If the actor lands on a selected node, it triggers the assigned action playing a synthesised note or triggering a sample. The actors also have a mode where they breed note selections if they mate.

6.4.5.4.3 The Predators

The Predators is an artificial life (ALife) instrument where predators and preys are left to interact in a neutral environment. The idea behind the instrument is to create a non-linear note or sample player; one controlled by the properties of an ALife system rather than random number generators. A screenshot of Predators is provided in Figure 6:14.

The predators have properties like energy, speed, restlessness, friction and focus on a prey. They lose energy by not eating from the preys or by fighting about them. When a predator takes a bite of the prey, the prey emits a sound. The sound is either a synthesized sound or a sample from the buffer pool. The pitch of the sound is defined either by the vertical location of the prey or by assigning a special pitch to the prey by choosing a note from a keyboard that pops up. Of course, preys and predators can be added to or removed from the environment.

The ALife properties of this instrument are rather simple. In the design process interaction elements like environmental obstacles and conditions together with properties of the actors like age, death, reproduction, etc. were added, but it turned out that these did not serve any musical purpose in this simple instrument. Had they remained implemented, theoretical complexity would have become stronger than interactive and musical simplicity. This is an important aspect of the design of ixi software: in order to make useable and popular instruments, they have to be inviting and engaging, which means that they are stripped down to bare essentials, where all unnecessary complexity is removed. The experience shows that there is a fine line between overly complex customability and pure accessibility. The former implies too steep learning curve and often constrains the user in terms of work practices, whilst the latter can be too simple, rendering the instrument more of a toy that, due to its lack of depth, is likely to be quickly thrown away.

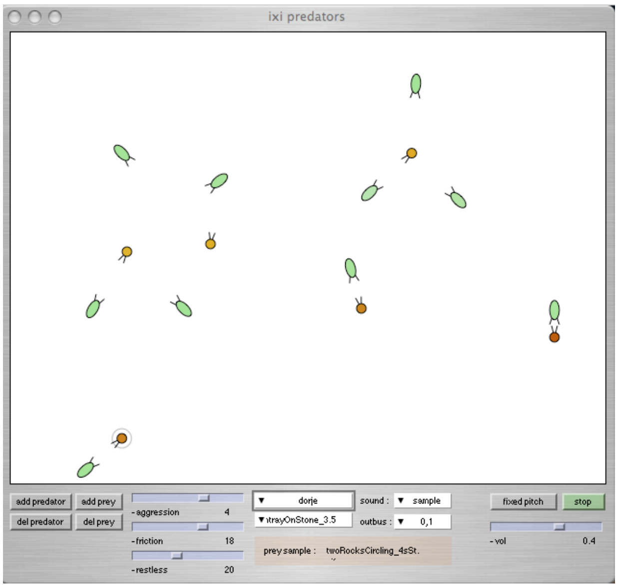

Figure 6:14. A screenshot of Predators

6.4.5.4.4 The PolyMachine

The PolyMachine is a polyrhythmic pattern sequencer that implements four different TempoClocks controlling each channel. The instrument originated from a study of Indian *talas* but turned into a parody of the typical drum sequencer. The PolyMachine consists of four (or more) tracks where the number of steps in each track can be defined individually. One can view the instrument either with a fixed box-size where the GUI window grows in size (Figure 6:15) or a relative box-size where the tracks adjust to the size of the window (Figure 6:16). Both modes

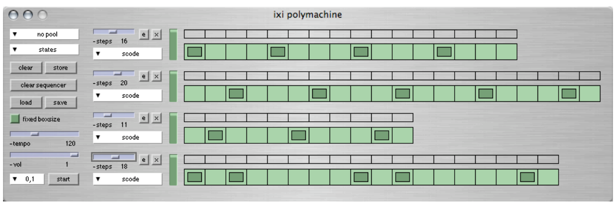

Figure 6:15. A screenshot of PolyMachine

have different qualities to them and it can be interesting to study the conceptual understanding of time and rhythm in the different representational modes.

Each track has a time indicator that travels above the sequence line and triggers an event if the box is selected. The event can trigger a sound sample or any function that SuperCollider can evaluate, such as sound synthesis, or sending OSC or MIDI messages to other environments or applications. The tracks have volume control and an envelope generator. Time can be absolute or relative to the track, which means that it either runs at the same speed for each step (and wrapping at different times) or it goes through the steps at different speeds (but wraps at the same time, resulting in a more repetitive but "out of beat" rhythms).

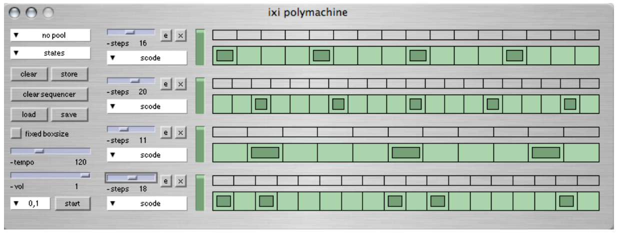

Figure 6:16. A screenshot of PolyMachine with a different representation of time

6.4.5.4.5 The GrainBox

The GrainBox is a two dimensional parameter space for granular synthesis. Traditionally, the problem with granular synthesis has been how to represent it graphically at the interface level as there are so many parameters involved. Here we represent the parameters with coloured boxes in a two dimensional space where boxes with related parameters are connected with lines. This makes it easy for the musician to intuitively understand the state of the sound engine by quickly glancing at the interface, as opposed to the complex analysis of slider positions where one has to read the label of each slider.

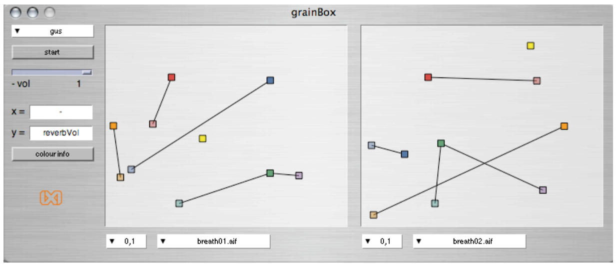

Figure 6:17. A screenshot of the GrainBox instrument

The audio stream of the GrainBox can be output on any audio bus and used as the source sound fed into and controlled by other instruments such as PolyMachine, Gridder or Predators. This way, the sound can be directly adjusted from the GrainBox application but utilised by the other pattern generating instruments that wrap the sound in an envelope. Of course, the GrainBox can also be used independently as a sound texture generator.

6.5 ixiQuarks as Epistemic Tools

6.5.1 The Fundamental Encounter

Picking up a musical instrument for the first time is a special experience. The instrument presents itself as a system of careful thought that one has to engage with on many levels, such as the embodied and the conceptual, in order to understand. Deleuze described such encounters: "Something in the world forces us to think. This something is an object not of recognition but of a fundamental encounter" (Deleuze 2005:139). A fundamental encounter with an object is not something that reinforces our identity or habits. Quite the opposite, it ruptures the stabilised habits of the self. Does the newly encountered instrument change the musical ideas of the user or reaffirm them? The first encounter with a music software and the process gaining an understanding of it is essentially a hermeneutic engagement. It involves at least two conceptualisations of music theory – that of the designer and that of the user – where it is the user's task to mould her habits to the functionality of the software. As the software itself is an artefact on a much more complex and conceptual scale than an acoustic instrument, the process concerns a circular (as in Gadamer's (1989) hermeneutic circle, [see section 2.3.4]) or perhaps corkscrew interpretation of the system's meaning. This meaning can be modified (or "hacked") and appropriated to one's own work methods and musical aesthetics, but in general one could say that the software defines the musician through its interface and interaction structures.

Now, how do the ixiQuarks instruments' design relate to the habituation of music theory and practices established by other software? The ixi instruments are eccentric, limited and focused on certain tasks, unlike much music software that (unsuccessfully) tries to attain full generality in its design.[139] We are interested in constrained tools that focus on certain tasks, are easy to learn, but still provide scope for in-depth study and mastery. The ixiQuarks do hardly relate to any acoustic musical instrument or hardware, but their design does not arise from a vacuum. The design metaphors and interaction models are inspired by computer games and multimedia design on the one hand, but physical actions (such

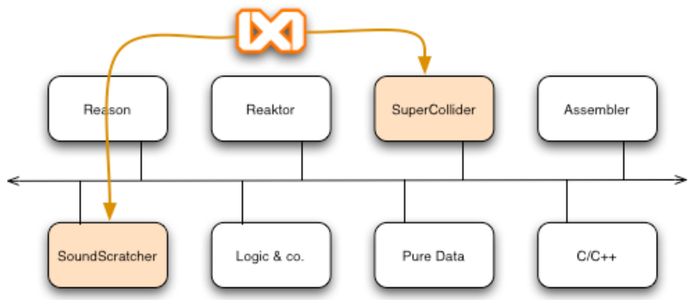

Figure 6:18. A placement of the ixiQuarks on the continuum axis where on the left we have increased constraints and less time investment and on the right we have general and open coding environments, where one has to spend much time in order to create music or expressive tools. The ixiQuarks are doubly represented on this continuum, as constrained instruments, but opening up for the generality and expressiveness of SuperCollider

[139] Not only in design, but in marketing as well: "The studio solution software package".

as scratching or drawing) on the other. These design practices make user-friendliness possible, as they connect to the user's prior experiences with digital technologies or physical actions.

In musical software, there exists a continuum from a narrow scope of expressivity (where the software is a clear personal expression by its author – perhaps allowing for some user interaction) to a wide scope of expressivity (where the tool has few restrictions and allows its users to express themselves more freely. see Figure 6:18). A rule of thumb is that the wider and more expressive the environment is, the more time users have to invest studying it, and the longer it takes them to compose or to design an instrument for their own needs.

The ixiQuarks are limited and constraining instruments with a focused functionality. They are intended for live, improvisational performance and to be played like acoustic instruments as part of musical ensembles. The main criteria for their design is to enable the musician to respond quickly to structural changes in the music, to change directions and to be expressive by bodily gestures, by extension via the supported joystick, tablet, mouse control, etc. However, the focus is not on the gestural part of the tools but on the cognitive or musical implications that a computational system for music always incorporates through its design: i.e., its epistemic nature. This focus involves the emphasis and exploration of non-conventional interaction and design models [section 5.3.3] on the one hand, and the prominence of improvisational fluidity, i.e., allowing for fast reactions, incorporated knowledge, and instrumentality, on the other.

6.5.2 The ixiQuarks as Epistemic Tools

Section 3.3.2 introduced the theory of extended mind, illustrating how the mind uses tools external to the body as part of the cognitive process. The phrase "external scaffolding" was used to explain how, for example, we move Scrabble tiles around in their tray to form new words, rather than memorising the letters and representing the words internally in the mind. Another example is how we rotate the bricks in Tetris to *see* how they fit rather than thinking it through. The theory of extended mind relates to earlier work by Kirsch and Maglio (1994) who show how actions can be divided into epistemic actions and pragmatic actions. Epistemic actions are physical actions that make mental cognition faster and easier. In order to facilitate optimal cognition, intelligent agents adapt their environment to get the most out of their limited cognitive resources.

Chapter 4 demonstrated how epistemic tools constrain, direct and enable certain cognitive tasks on the individual level and on a more historical level we see how cultures produce, adapt and are affected by the technologies of epistemic action. Analogously to Tetris, musical software serves as external scaffolding for the composer. The interface both affords musical potential and stores musical parameters that otherwise would have to be kept in mind. The software engine (rather than the tactile controller) becomes the "locus" of the composition or the musical performance; an extension of the musician's mind. The music theory is carved into the functionality of the software's interface *and* its internal parameters and algorithms, both of which have to be habituated by the user through practise. The interface of a digital instrument becomes a cultural territory; a space where musical ideas can originate and take shape. It is a result of our musical culture and technology, but at the same time it actively influences the contemporary musical soundscape around us. This is not to deny that the acoustic instrument has specific affordances and lends itself to certain musical styles and playing, but more in order to point out how musical software normally resembles the score more than the instrument used to play the score and thus affects the musician on a more formal or theoretical level.

The instruments of the ixiQuarks software suite contain active, affecting, adaptive and automatic elements of various degrees. Musicians "offload" (Clark 2003) some of their cognitive functions to the instrument itself that continues playing or influencing the music in its own way. The tool either encapsulates a mechanism too convoluted to keep in mind at any one time or activities too complex to perform with a bodily gesture. The ixiQuark instruments are limited to a specific design idea but the user is provided with the power to extend the functionality of the instruments by coding extensions to it or creating new synthesis algorithms for it to use.

Figure 6:19. The iLog: a hardware user interface that can be used with the ixiQuarks

The survey reported on in section 5.2 showed that people find the limitations of acoustic instruments fascinating and enjoyable, leading to them wishing to master the tool, and succinctly establishing a relationship of affection. On the other hand, the survey also indicated that people often found problems of acoustic instruments in that they were resistant to change and playing them risked being a cliché-prone activity. In a special part of the survey we stepped out of its general focus and asked questions about ixi software in particular (this was before the release of ixiQuarks). We learned from the participants that their concerns were the same, i.e., that people found joy in the limitations of the instruments and in exploring their scope and expressive depth. However, many reported that after a while they missed the option of being able to extend the instrument and to get it working with other software in a more complex setup. Addressing this issue was one of the design criteria in the creation of ixiQuarks.

6.5.3 ixiQuarks Located in the Taxonomic Analysis

Chapter 5 introduced a theory of digital musical instruments. It illustrated with a survey how people experience the differences between the acoustic and the digital, as discussed in Chapter 4 in the section about material epistemologies [section 4.4]. An ontogenetic account was given on 3 different instruments, the sax, the MiniMoog and the reacTable. Finally, taxonomic structures were given as tools for analysing digital instruments. This section will apply these analytical tools in the context of ixiQuarks.

The ixiQuarks actively make use of the programmatic nature of digital technologies. It is rarely a one-to-one gesture-to-sound generator (although that is also provided in instruments such as ScaleSynth and Quanoon), but more often an environment in which the user can set up a complex musical structure that can be set into action with various generative algorithms, such as stochastic, Markov models, artificial life (Langton 1989, 1997; Ramos 2002) and rule-based AI techniques. This can be seen as a process where the user delegates parts of his cognitive structures onto the epistemic tool. The technology is now acting according to the rules set by the user. However, the aim is always to provide the user with a direct control over these external processes, such that they can be stopped or altered in realtime.

The ixiQuarks can be controlled by various hardware interfaces, such as the mouse, keyboard, pen tablet, MIDI instruments, and custom interfaces such as the iLog.[140] An embodied interaction is provided, but what differentiates ixiQuarks from many digital musical

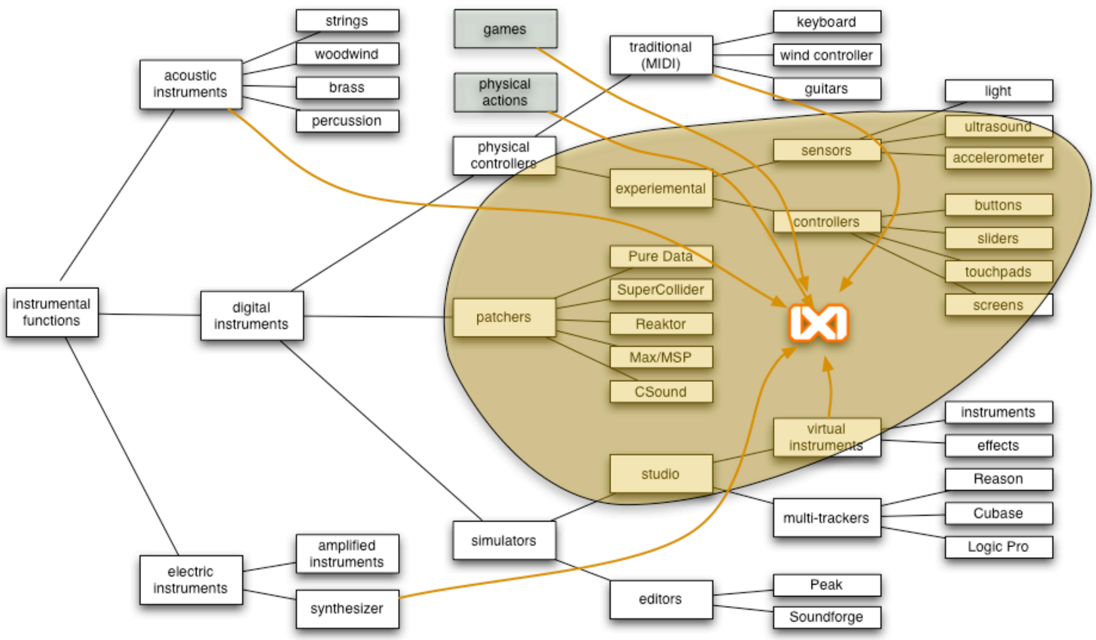

Figure 6:20. ixi software located in the tree graph of instrumental fuctions. The ixi software makes use of patchers in its design and actively encourages the user to work on that level. The design covers experimental controllers and virtual instruments. The work on ixiQuarks relates to the areas that are highlighted by the orange overlay. It is inspired by virtual instruments, synthesizers, acoustic instruments and MIDI controllers, as well as games and physical actions. (indicated by orange arrows)

[140] For a demo, see: http://www.youtube.com/watch?v=OjdacELem_Y

instruments is the visualisation of its sound and mapping engines. Often these patterns are hidden and the user has to build an intuitive understanding of the engine's parameters. The ixiQuarks thus explicitly bridge the divide between the purely automatic and the purely embodied, resulting in instruments that have a strong degree of collaboration between musician and the system. As a type of composed instruments, the ixiQuarks emphasise the musical logic inherent in each and every instrument, pointing the user at the arbitrariness of all such designed tools. Figure 6:20 outlines the scope of the ixiQuarks in the context of other technologies.

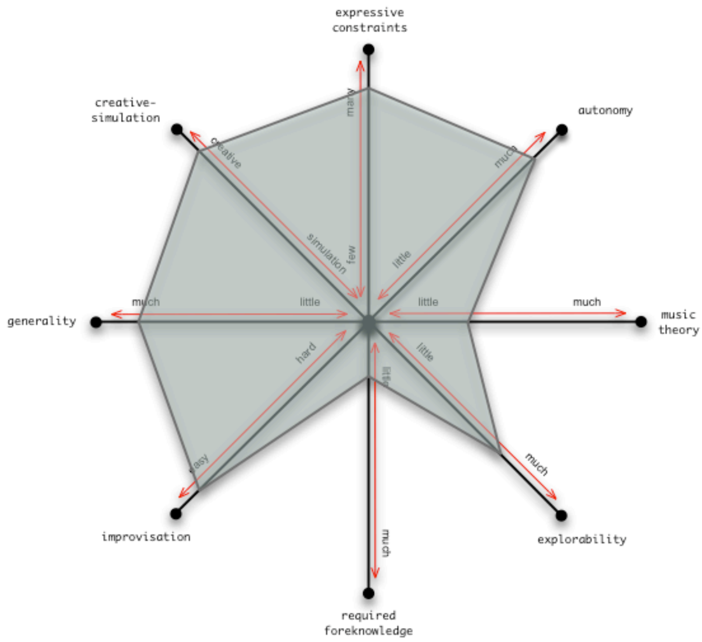

Figure 6:21. The ixiQuarks located on the epistemic dimension space

If we analyse the ixiQuarks in the epistemic dimension space of digital musical instruments, we see that the ixiQuarks do not require a high degree of musical knowledge. They lead the user to perform in certain ways, automate processes, but are highly open for improvisation. They are general in expressivity (particularly when the power of SuperCollider is taken into account) and their interface and interaction design scores highly in the creative-simulation axis. The interfaces are creative as opposed to simulating physical hardware or traditional music work processes. The surveys have demonstrated that the ixiQuarks score sufficiently on the explorability scale, i.e., that behind the simple interface lies a broad field of musical expressivity for investigation. Of course Figure 6:21 should be supplemented with Figure 5:18, of Supercollider, as ixiQuarks is practically these two environments merged into one.

6.5.4 Conclusion

An interface is essentially an abstraction. It is a higher level representation of a structure of further complexity. As such, all efforts to build tools with interfaces are a process of limiting the scope of expression. This section has distinguished two types of interfaces in musical performance with digital instruments: physical and virtual interfaces. The focus has been on software as the conceptual engine of the instrument, the location where parameters of the epistemic tool are set and controlled. This applies in all situations, whether the instrument is controlled by complex gestural sensors or the typical mouse and keyboard interface.

The concept of the interface as limitation applies to graphical user interfaces and programming languages alike. Classes in programming languages are essentially interfaces for potential functionality. The interface is therefore always a limitation and a worldview, but from a musical perspective (as opposed to software engineering), the level SuperCollider represents as a musical environment, is probably the one that is the least limited and constrained, yet expressive, of those available today. We have seen that an interface is a conceptualisation of an action of epistemic nature; a semiotic design that invariably defines the possibilities in thinking and performing whilst using the tool. The ixiQuarks presented above are constrained ideological instruments, designed for specific expression. The aim is not that of musical generality, but rather a focus on particular interactive patterns. However, this happens in a context where anything can be added or built as satellites to the instruments either as part of the ixiQuarks or simply by livecoding it in realtime in a performance. I believe that with the ixiQuarks, and the merging of code and GUI in one epistemic tool, the problematics of constraints and expressive openness have been addressed in one possible way; a path that has served myself well in live improvisations with acoustic instrumentalists where quick response, fast changes and liveliness are common traits of the performance.

6.6 Development of ixiQuarks: The Tool as Thought

In section 6.3.1 there is a quotation from James McCartney, the original author of SuperCollider, where he points out that we learn to conceptualise our problems in terms of the tools that we use. A popular saying captures this well: “When the only tool you own is a hammer, every problem begins to resemble a nail.” We have repeatedly explored the question of inbuilt thoughts in technological artefacts, and this is equally relevant in programming languages for computers. Sections 6.5.1 and 5.5.3 illustrate how different levels of coding means different things to the person creating. What is being created, a musical piece or a musical environment? Where are the boundaries? Or is the system perhaps a musical instrument? If so, where is the dividing line between a composition and an instrument? Since these questions are focal to this thesis I conducted a survey wherein creators of audio programming languages explained their aims and methods in creating a system for synthesis and generative music [see section 7.5].

SuperCollider, my personal choice of audio programming language, initially seemed to fit my primary criteria for an artistic tool: it was an interpreted language, object oriented, it was textual, modular, had an abundance of unit generators and supported various mechanisms for creating generative music. Furthermore, in recent years it has become increasingly good for graphical work. I have found that SuperCollider as an environment has influenced my work, both in terms of what inspires me and the manner in which I implement my ideas. Specific GUI primitives, 3rd party classes, or unit generators generate ideas when I try them out and by exploring a particular idea – which is effectively exploring the affordances of the object under inspection – I typically arrive at some conception that can be developed into an instrument. I would therefore point out that in my case, the technology is the material that I use to think with, the clay that contains properties that can realise themselves as interesting instruments. The question of originality or ingenuity has to be deferred and differed (as in Derrida's concept of *différance*) due to the highly complex network of people, technologies, and ideas that constitute the actor-network that SuperCollider is. To explain this in practice, I will provide a case study here below.

6.6.1 From a Sketch to an Instrument: A Case Study

Looking out of the window on a cold winter day, the snowflakes might yield the desire to create a simulation of snow. In SuperCollider one could draw the snowflakes using a class called Pen, but it would also be possible to check and see if the MultiSliderView could be enough (this would probably be more convenient in terms of processing power as the multi-slider view is implemented as a primitive in C++ and therefore faster). Below is a simple program written in SuperCollider that has four layers of multi-slider views with transparent backgrounds, so one can see through to the next layer of snow. When the snowflake lands, it triggers a bell-like sound.

```
SynthDef(\snowBell, { | freq=440, amp=0.4, pan=0 |
        var x, env;
        env = EnvGen.kr(Env.perc(0.001, Rand(550,650)/freq, amp), doneAction: 2);
        x = Mix.fill(6, {SinOsc.ar(freq*Rand(-10,10), 0, Rand(0.1,0.2))});
        x = Pan2.ar(x, pan, env);
        Out.ar(0, x);
}).send(s);

(
var win, msl, trigAction, snowloc, speeds, speed, layers=4, snowcount = 62;
// fill an array with arrays (number of layers) of locations
snowloc = {{rrand(0.38,1.5)} ! snowcount} ! layers;
// fill an array with arrays (number of layers) of step size (speed)
speeds = {{rrand(0.01,0.018)} ! snowcount} ! layers;
speed = 0.1;

win = Window("snow", Rect(11, 311, 520, 240), border: false).front;
win.view.background = Color(0.14,0.17,0.24);

msl = Array.fill(layers, {|i|
                MultiSliderView(win, Rect(-1, -1, 522, 242))
                        .strokeColor_(
        Color.new255(rrand(22,35),rrand(22,35),rrand(22,35)) )
                        .fillColor_(
        Color.new255(rrand(222,255),rrand(222,255),rrand(222,255)) )
                        .valueThumbSize_(rrand(2.8,3.8))
                        .indexThumbSize_(rrand(2.8,3.8))
```

```
					.gap_(5)
			});
	// when the snow falls. (pitch is mapped to index and amplitude to speed)
	trigAction = {arg drop, amp;
			Synth(\snowBell, [\freq, 400+(drop*20), \amp, amp, \pan, rrand(-0.8, 0.8)])};

	t = Task({
			loop({
				snowloc = snowloc.collect({|array, i|
					array = array.collect({|val, j|
					val = val-speeds[i][j];
					if(val< 0.0, {val = 1.0; trigAction.(j, speeds[i][j]*10 )});
					val
				});
				array
			});
				{ layers.do({|i| msl[i].value_(snowloc[i]) }) }.defer;
				speed.wait;
			});
	}).start;

	// on stopping the program (Command/Ctrl + dot) the task will stop and the window close
	CmdPeriod.add({ t.stop; win.close; });
	)
```

Figure 6:22. *Snjókorn*. A program with 4 layers of **MultiSliderView** triggering sounds

This is a typical work habit of coders exploring the potential of the programming environment in which they work. The act of creation is an interaction between the coder and the creative tool, where ideas are thrown back and forth, implemented and evaluated, and improved or thrown away. For the coder, this process can feel like they are engaging with the creators of the language, but not only them; also enrolled into this structure are the designers of operating systems, hardware and naturally the culture of electronic music production. The process, as described here is fundamentally different from the traditional “waterfall” model of computer science (Winograd 1996) illustrated in Figure 7:1.[141]

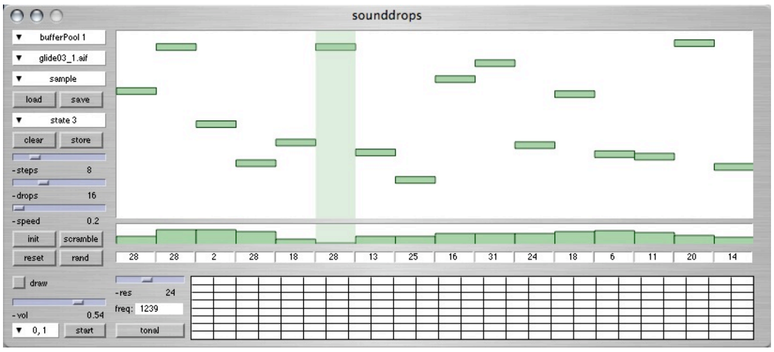

Figure 6:23. A screenshot of *sounddrops* in microtonal mode where each of the drops (there can be from 2 to 48 drops in the view) has properties such as sound function (sample, synthesis, code or audiostream), pitch, amplitude, speed and steps. The microtonal keyboard under the multislider view has 7 octaves (vertical) and from 5 to 48 notes (equal tempered tuning) in an octave.

141 The waterfall model has been criticised even within the realms of hard-core computer science (Parnas & Clements 1986) where it is pointed out that humans rarely have access to all the variables that are needed in order to design a whole system.

The above sketch was found to be interesting enough to explore a bit further. It can now be designed as a prototype for evaluation (Greenberg & Buxton 2008). After adding some functionality, the prototype has turned into a full blown instrument (called *sounddrops*) that is now distributed as part of the ixiQuarks. It makes use of the bufferPool tool of the ixiQuarks system, so when the drops land, they can trigger a sample, various synthesis types, code (allowing for livecoding and morphing the instrument into something else in realtime) or envelope an audio signal running through any audio bus (other ixi instruments could be outputting their signal on say audio bus 20 and sounddrops would then listen to that). As seen from the example above, each slider in the array can have its own speed and the **Task** takes care of recalculating the drop location according to another array, **speeds.** A simple sketch that originated in a ping-pong exploration of the language has now been turned into a powerful tool that can be used for complex sequencing of polyrhythmic temporal structures, allowing for ease of control, pitch mapping and most importantly: a graphical and aural representation of a process that can inspire the musician.

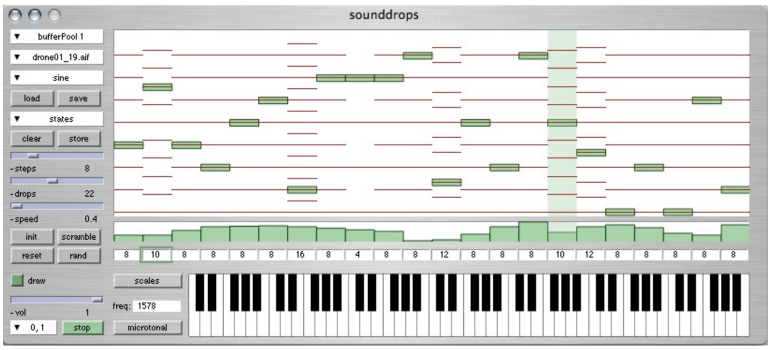

Figure 6:24. Here we see *sounddrops* in tonal mode. There are two pop-up windows as well, one is the coding window for livecoding and the other is a window that contains various scales and chords that can be used to color the keyboard. These keys can then be used to assign the frequencies to the drops.

All SuperCollider users eventually come up with their own design for organising buffers, busses, effects, patterns, tools and instruments. The ixiQuarks is but *one approach* to such organisation. The decision to modularise the code, such that instruments could make use of the same bufferpools, and internal audio routing structure has made the development of new instruments very fast. When the system is modularised this way, the process of turning a sketch (such as the snowflakes here above) into a full-blown instrument that works with ixiQuarks might not take more than a few hours.

6.6.2 Virtual Embodiment in Screen-based Instruments

As we have seen, the interface is a template for how people create music. It imposes a specific way of thinking and an ergonomic as the musician habituates herself with the software. Recent theories in cognitive science (Clark & Chalmers 1998; Dennett 1982; Churchland 1995) have demonstrated how the environment is used as external scaffolding in the cognitive process but they do not address the question of how those technologies *affect* our thinking. In media-theoretic terms we could point out how we now rely on the internet and specific search engines to access much of our knowledge (Manovich 2001). The reference trace in the extended database can thus become more important than memorising the actual content. In general, Clark's argument is that evolved creatures will neither store nor process information in costly ways when they can use the structure of the environment and their operations upon it as a convenient stand-in for the information-processing operations concerned. That is, know only as much as you need to know to get the job done.

Software such as the ixi software – with all its agents, pattern generators and automation – actively encourages or affords such cognitive offloading by the user. Compositional ideas are both consciously and subconsciously constrained by the limitations of the tools we use to compose *with* or the instrument we compose *for*. A composer has to take into account the sonic potential of the instrument he or she is composing for. In the same manner a musician using software tools learns the limits of the software through using it and the composition evolves from the dynamic relationship between the musician and affordances of the software itself. Software defines the creative space in which the composer or musician can think and they learn to think in a way that fits the software. One learns to think in a programming language, a software environment, or within the expressive potential of an acoustic instrument.

Artistic software exemplify tools providing frameworks that can be used as external representation of cognition. As covered in chapter 4, the structure of the tool itself will affect the artists thinking in concealed ways. The software contains the clues, it represents the states of the engine with well known metaphors taken from the physical world and all is presented as neutral and at face value. The way we compose and perform music with such software is through practising – we practise the software just as we practise acoustic instruments – and we master it. The difference can be found in the nature of these practices: instead of incorporating rote movements, the practise digital musicians engage in involves reading manuals, testing setups, exploring possibilities, and stumbling into constraints – in general, acquainting themselves with the conceptual structure of the compositional environment itself. It can still be meaningful to define this mediated interaction as embodiment as there is always some physical structure that controls the engine. In the digital instrument this embodiment is epistemically mediated (see section 7.2.5 further on virtual embodiment).

The icons and graphical metaphors of the screen-based musical instrument are both the control mechanism (the interface) and visual representation of the performative state of the instrument. The user of such a tool does not have to represent the state in her own mind, but projects that thought process onto the instrument in use. In other terms, the instrument performs a considerable amount of the musical thinking and stores the musical structure that in many other systems has to be kept in the composer's or the performer's mind. This automation and cognitive offloading is not as achievable with acoustic instruments as they have to be continually activated in order to get sound and automation is practically impossible.

6.6.3 Conclusion

In this section I have shown how the environment inspires innovation through play and exploration of its affordances. It has been argued that in artistic programming, the ability to work from a bottom-up design is important and the fact that SuperCollider is an elegant, object oriented, interpreted language makes it extremely well suited for such experimental coding. The more protocols, hardware and graphics the environment supports, the more it lends itself for inspiration where the creative mind and the tool interact in a way that can be seen as improvisation, sketching, exploring, building and composing a complete piece or tool.

The ixiQuarks GUIs are built on top of SuperCollider. They can be seen as creative limitations that live on top of an ocean of potential expression as the SuperCollider language is extremely open and expressive. As the survey in section 5.2 shows, people like to have constraints and limitations of all kinds when working creatively. This is true not only of constraints of tools or instruments but also for music theory or notably film theory.[142] The ixiQuarks are based on this understanding of affordances and constraints. SuperCollider is extremely open and expressive environment, perhaps the ideal intellectual partner for the musical free thinker. The ixiQuarks, on the other hand, focus, concretise, and constrain. They prime the user's mind into a certain way of thinking, a way that can be beneficial for musical creativity. It is herein that the power of the ixiQuarks lies for the current author: to be able to sketch and perform intuitively and quickly with tools that one has mastered but still be within the context and expressive scope of SuperCollider itself.

6.7 Surveys

6.7.1 Introduction

In order to get an understanding of how the ixiQuarks are used by other people, all possible gates were opened for feedback. Over the years I have given workshops, taught classes at university level, created on-line feedback mechanisms and directed dedicated user-studies. All this has been invaluable and I am grateful to all those people that have downloaded the software and used it in their own musical practice. However, in the context of this research it was concluded that a more formal online survey and face-to-face interviews were required in order to get more systematic feedback, with both quantitative and qualitative data, yielding a more comprehensive understanding of the tools.

6.7.2 Online Survey on ixiQuarks

6.7.2.1 Introduction

[142] As the example of the Dogme-movement illustrates: http://www.dogme95.dk/

Having implemented many of the ideas of virtual embodiment and interaction from the ixi interaction models (explained in [section 5.3.3]) in the design of ixiQuarks, I embarked upon creating an online user survey that would evaluate ixiQuarks as a software environment for musical expression. Since the primary reason for developing the ixiQuarks is a personal journey into music and musical performance, it was important for me to get an understanding of whether the users feel that they are using another musician's personal environment for expression or if they could use the software in general terms in their own creation. The question of limitations, constraints, and familiarity are therefore germane here.

The online user-survey of ixiQuarks[143] was conducted between June and December 2008. Questions were designed using general heuristics of user studies (Cohen (2000); Schneiderman (1986); Sharp, Rogers & Preece (2002); Perlman (1997); Chin, Diehl, Norman (1988); Nielsen (1993); Davis (1989); Lin, Choong & Salvendy (1997); and Lewis (1995)). Many of those heuristics are general and suitable for this study, but some focus on web-design or more specific applications, so questions were adapted for this study of an expressive, creative, musical tool.

From the 1600 registered users of ixiQuarks (people that willingly submitted their email in order to participate in the user study) there were 197 answers.[144] As the sections of the survey contained both quantitative and qualitative data, an outline of the findings from each type will be given.

6.7.2.2 Quantitative Findings

6.7.2.2.1 Personal Details

The questionnaire begins with demographic questions in order to know the survey participants. The mean age of participants was 35 years, where the oldest participant was 68 years old and the youngest 14. Of those, 41 defined themselves as musicians, 30 as students, 27 as composers, 20 as educators, 16 as artists, 12 as designers, 10 as technicians, and 5 as sound artists. Some defined themselves as more

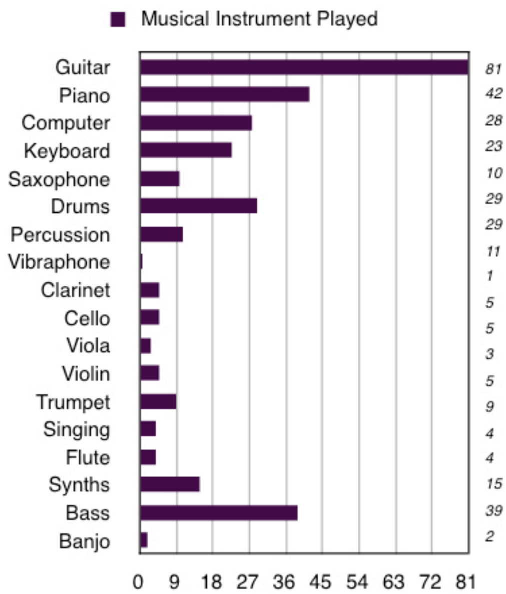

Figure 6:25. 189 participants answered the question of which musical instrument they played. Naturally, some participants played more than one instrument

[143] To be found here: http://www.ixi-audio.net/content/download/ixiquarks/survey/

[144] This means that 5% of users that download the software registered, and of those 10% actually participated in the survey.

than one of the above, and others resisted definitions. The participants were from all over the world, mainly from Northern-America, the UK and Western Europe. There was a considerable number of replies from Asia and South-America, but none from Africa.[145] The participants had spent on average 16 years playing music, and of the 163 people that answered the question of formal musical education, 99 people confirmed that they had received such training. The participants had worked on average 9.5 years with computers in music, which is a relatively long time, considering how primitive the technology was only 15 years ago. They come from all stylistic genres and musical tropes, although 'improvisation', 'electronic music' and 'experimental' were popular defining terms.[146]

Any survey typically attempts to gain a picture of what kind of users are answering it. It was interesting to know what instruments the participants play, as the ixiQuarks relate more to some instruments than others. However, that said, the popularity of some instruments like the guitar or the piano obviously skews that picture too much for it to yield any statistically significant findings. Figure 6:25 shows the distribution of musical instruments.

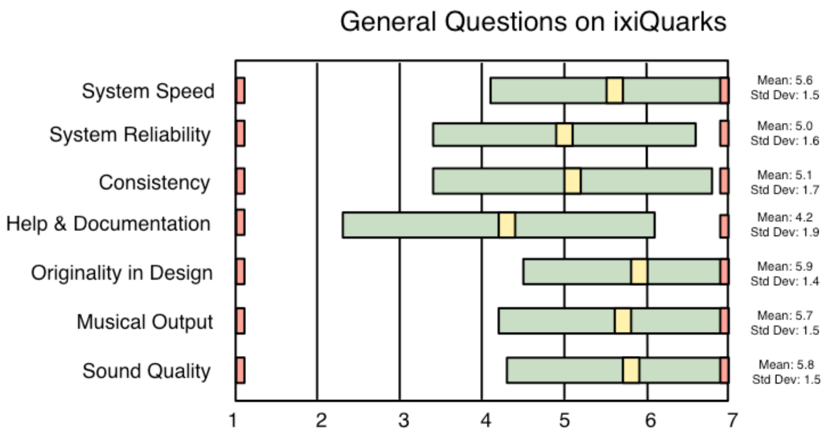

Figure 6:26. Participants answered by ticking boxes from 1 (bad) to 7 (good). The red blocks show lowest and highest marks given, the yellow block shows the means, and the green blocks are the standard deviation.

In Figure 6:26 the results from the general questions about ixiQuarks are represented. Here it is noticeable how unhappy the users are with the help and documentation part of the environment. This is partly due to neglect by the author, but partly also because of the belief

[145] Language is an important factor here as the survey is only in English.

[146] Here are the most common genres specified (and some enthralling ones): Ambient, Drone, Acoustic, experimental, visual, garage, chilled out jungle, wonkey, glitchy, techno, electro, gabber, bleeps, ambient, micro-sound, amazing, dada noise making, folk/minimal/electronica/pop/electro, experimental electroacoustic, techno/electronica, Alternative, Improvisational, experimental/avant-garde/shitty, \"modern classical\" (hate the term), meditative, experimental, ambient, Jazz Fusion, indie/electro, ambient, minimal, dub, glitch, experimental, noise, industrial, electro acoustic, techno, electronica & musique concrete, bossa rock experimental, Outre, weird, folk / post punk / new wave/ post apocalyptic ambient, Post-Genre, improv, abstract headfuckery, Ambient/IDM, contemporary orchestral / jam band / hip hop / house and techno, prosthetic, The music of the water, Free Improvisation and Flamenco, dubstep, Grandcore Circuit Noise ; Freak Noise, electronica.

that all musical instruments might well contain hidden features that make them rewarding to explore. Furthermore, if the instruments make use of the same design models, knowledge of one instrument should transfer to the others. This was one of the questions that we enquired about. Reliability and consistency scored less on average, which can be explained by bugs discovered in some of the filters and instruments. Most of these bugs have been fixed thanks to this survey and other user feedback. However, originality in design, sound quality, musical output and general system speed get a good grading.

Figure 6:27 represents the general *experience* of using ixiQuarks. It is interesting how positive the answers are, considering that many people complained about lack of documentation and tutorials. The values in "terrible/wonderful" and "dull/stimulating" fit well to the comments in Figure 6:27 where we see that people enjoy the experience of working with ixiQuarks, and find it a stimulating, inspiring environment to work in.

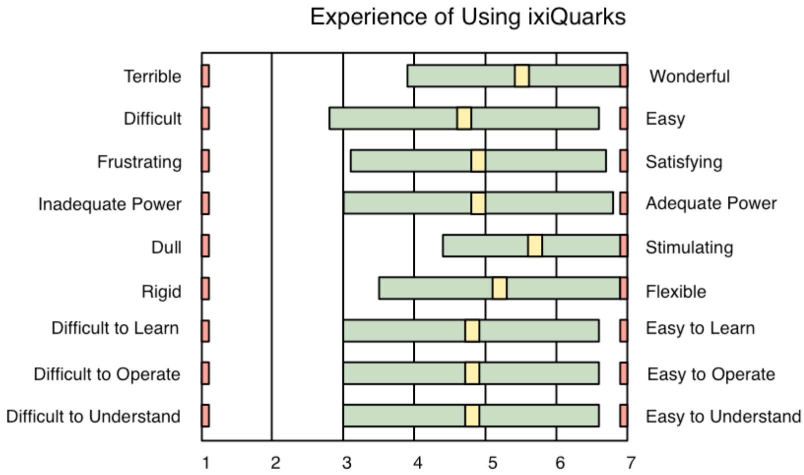

Figure 6:27. The participant rating of the general experience of using the ixiQuarks.

Figure 6:28 shows how people rate the independent sections of the ixiQuarks environment. As the instruments are the main point of the work with ixiQuarks, it is positive to see how well they scored on the satisfaction scale. The utilities are useful for various tasks and do provide features that are not common in many non-programming audio environments, so they get a good grading as well. The effects and filters are useful, but come in a (yet) typical GUI of values controlled by sliders. It is noticeable how bad the spectral filters score and this can be explained by how underdeveloped they are in the version of ixiQuarks used for this survey (version 5) but also due to their technical terminology. This is acknowledged by the author and shows that the survey is indeed working. One participant commented: "Replace all computing based terminology into musical terms, 'spectral

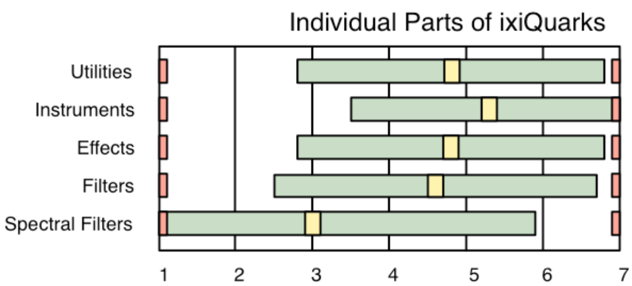

Figure 6:28. This graph shows how people rated the independent sections of the ixiQuarks environment, on the scale from 1 (bad) to 7 (good).

binshift 2x2' makes very little sense." This is something we have tried constantly in the design of the ixi applications, but in the undeveloped spectral plugins, this has not been achieved yet. (There is also a question if there is a "musical term" equivalent to the process of shifting bins in a Discrete Fast Fourier Transform).

In Figure 6:29, we see how 185 people evaluate each of the instruments that make up the ixiQuarks instrument category in version 5. Interestingly GrainBox gets the highest mark. It seems that people enjoy exploring granular synthesis with a graphical user interface. In fact, there are very few available GUI based applications that provide people with the possibility of manipulating sound grains in realtime performance. SoundScratcher also scores high, probably due to uncommon direct access to visually represented samples in the sound, and the multiple interactive modes in which the sound can be manipulated.

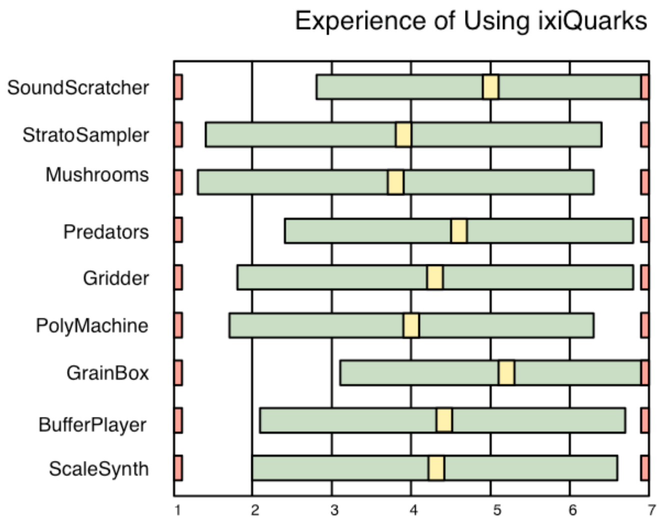

Figure 6.29. Voting favorite instruments on the scale from 1 (bad) to 7 (good).

6.7.2.3 Qualitative Findings

The most important part of this survey is the conversational questions that cannot easily be translated to quantitative data. Since the beginnings in 2000, the ixi software team has been highly influenced and inspired by collaborations and discussions with musicians and sound artists, and other user feedback. We have created software for particular musical pieces or sound installations that are later made general under the ixi name. For us, this is a normal way to design software and we find that our approach fits naturally with many features of situated design, participatory design, extreme programming, etc. although we are never orthodox in our approach, nor following any of the mentioned theories blindly. Below I will address each of the questions posed in the survey and relate that to the design of ixiQuarks in a general manner.

6.7.2.3.1 Context of Use

Roughly how often have you used ixiQuarks? Do you use it regularly? Where does it fit in your workflow? In what context? (live, studio) Do you use it with other software? If so, how?

Here I am interested in the context in which ixiQuarks is used and if it forms an important part of the workflow of the user. The replies reveal that people generally use ixiQuarks in two fundamental contexts:

a) in the early stage of the compositional process ixiQuarks is used in the studio as a tool to explore sonic materials, sound design, or performance (as in a acoustic instrument). ixiQuarks are often used to generate sounds that are later used in a multitrack sequencer. Some people stream the sound directly from the ixiQuarks straight into the sound-track of a digital audio workstation (DAW) such as Logic. In this context, the ixiQuarks provide musical instruments that can be performed and recorded in realtime.

b) the ixiQuarks are also typically used in a live context for improvisation. Here they are often used to sample the sound of acoustic instruments and manipulated by the various instruments, typically simple hardware interfaces like a mouse or a keyboard, perhaps a pen tablet. Examples of the comments:

- I use it mostly for capturing improvisation as sketches or creating samples
- I use it for creating grainy sounds and also use it for processing cello sounds for lush granular soundscapes
- I don't use it at all in my personal music work (I design my own instruments for that). But, I use it for fun, inspiration, and recommend it often to people who need a basic but very free environment to mess around with sound
- I often use ixi for live performances, mostly electronic improvisation with other musicians. On the studio I use it a lot for doing soundscapes, glitches and other sounds that are later used on more "traditional" compositions, mostly done in Ableton Live.
- I don't compose without it.
- Tried once but gave up.
- ixi is my favorite piece of software. Use it in the studio all the time. Use it with Forester and Logic.

There were 190 of these comments,[147] and the quotes above give some idea of the width of the responses. Some users talk about their studio work, some about performance and yet others find no use of it at all as a tool that fits their work process. These responses reflect the feedback given in Figure 6:26.

6.7.2.3.2 Hardware Setup

The hardware setup. What physical interfaces do you use to control the ixiQuarks? (Soundcard, Mouse, Keyboard, Wacom tablet, MIDI, etc)

This question is of interest as the ixiQuarks are mainly designed for "the-computer-as-it-comes" (Armstrong 2006). At the moment, interface capabilities are being implemented through the use

[147] A larger selection of the comments can be read in Appendix II. It did not make sense to print all of the comments however, due to their similarity.

of MUIO devices and WiiMote control,[148] but these are of peripheral interest as the main focus of ixiQuarks is the sound engine, and tools for musical composition. The answers were slightly surprising here as people are happy to perform without the physical controllers they are used to in acoustic instruments, but at the same time we find people highly creative in their use of the software where the following interfaces were used: mouse, Wacom tablet, MIDI, trackball, headmouse, MIDI keyboards, MIDI sax, MIDI knob-controllers. Obviously only limited features of the software can be controlled with MIDI, for example MIDI data cannot currently be mapped to the scratching action in SoundScratcher.[149]

6.7.2.3.3 Initial Encounter with ixiQuarks

Describe your overall reaction/experience when first encountering the software.

With this question the aim is to understand the initial experience, that of encountering the unknown (if so); what Deleuze calls the fundamental encounter [see section 5.6]. The interest is to get some idea of what initial reaction the software triggered, when the mind is fresh and not polluted by the software's ideology.

The answers were flattering, in their positive flavour. However, many complained about the initial complexity of the environment, the hidden features and the (sometimes) difficult journey to understanding a new design philosophy. These were favourable results. I agree with Jordà (2005), Chadabe (1997) and Wanderley (2001) that the HCI requirements of obviousness and conventions in interface design, of clarity and practically no learning curve, do not apply to musical instruments if they are supposed to be interesting and engaging enough for in-depth exploration and practice [see section 3.6.1]. I acknowledge that tutorials and documentation can be improved, but conversely, posing a challenge for the user that requires engagement and dedication is also a fascinating thought.[150] In a way, the digital musical system here draws from the fields of acoustic instruments (of putting in hours of embodied practise) and video games (where the player enters the unknown and the hidden). Although the functionality of the whole system can be described in a manual (unlike the acoustic instrument), it can be argued that this is not necessary as users will "grasp" the thinking behind the system. One participant in the survey stated this: "Once I understood its logic all became easy and intuitive."

- It was a little hard to get a first, but after a few tries it made sense.
- Excellent, very fun environment to play with, very original instruments and effects, for the most part.

[148] http://www.muio.org and http://uk.wii.com/
[149] although this is possible now with the MUIO sensor interface in the current version of ixiQuarks.
[150] Obviously a model not possible in the field of commercial software, but that's not where ixiQuarks belongs.

- Exciting, visually stimulating, especially the scalesynth
- I got a little confused with this inputs - outputs logic and I hade some problems with my soundcard. It seems that ixi does not work when it is on 48.000hz - 24bits, which is the configuration that I mostly use.
- I was always frustrated by the lack of an all-in-one tool for improvisational electronic music composition. The ixi tools were a revelation. The fact that you've put all of the best together in one package has been a godsend.
- About as close as possible to a dream come true. I was trying to write similar software in SuperCollider, but lack the time and expertise. It allows me to focus on writing music instead of spending days on a software component.
- Very user friendly and inventive.
- Stuff to learn but not very difficult.
- A lot seemed very random when experimenting, as you gain experience you learn the purpose of the tools.
- I get high with the software.. I feel that I have found exactly what I was looking for.
- I thought it was the most amazing cool software ever...
- This is fun but useless to my style of music.
- Total frustration.
- What the hell is this??? How do you use it???
- At first I was a little overwhelmed, but now I like it.
- It's very different than anything else I've played with.

These are the reactions that I had hoped for. As the work on ixiQuarks is very much a personal journey – an experiment into music theory, instrument design, composition and interaction design – the interests and needs pursued tend to be personal and not those of other users as customers. Of course, I do try to make the software user friendly, considering all the rules and accumulated knowledge of HCI, but the interest is in creating eccentric software, a musical expression, a personal statement, and I am just as happy when people reject that statement as when people embrace it.[151]

6.7.2.3.4 Difference from Commercial Software

The freedom from having to think in commercial terms, as mentioned in the section above, is very important in the work of ixi. At one stage we entertained the idea of marketing our software, but we experienced that this idea changed our thinking, making it more generic and less creative. Programming ceased to be a fun act of creation and became "work." We therefore rejected this idea and kept on working on our software as a research project. In relation to the "eccentricity of software" question above and its relation to the world of commercial software, the question asked here was:

[151] This is obviously not the case with commercial software, where the aim is to make every customer happy and attract as many customers as possible.

Do you feel the ixiQuarks environment is different from the commercial products? If so how? Can you think of any software (commercial or not) that focuses on the same areas of interactivity?

Unequivocally, people claimed that the ixiQuarks was different from commercial software and that they found its place in the plethora of musical software to be quite unique. The commercial software is general and impersonal, whereas much of the open source and free software (which require the user to develop his/her own expressive environments) are too complex for many people to use. The ixiQuarks exist in a niche but extend into the complexity of programming activities as it plugs seamlessly into the architecture of SuperCollider. The responses here were in general very positive, but below examples of all opinions are given (thus not representative of the general opinion):

- Yes it is. It cuts out all the bullshit and focuses on creation.
- Much less idiomatic, more “play” oriented instrument design - allows for more experimentation.
- Ixi is not based on “recording” and “composing” in traditional way. It seems to me that it is more a performance software (of course you can record what you do, but it is different), which is a good thing also.
- I feel that it really fills a gap in the commercial product range with its flexibility and original set of tools specific to computer music generation.
- Yes. ixiQuarks -- as well as the rest of your software -- feels as though it’s a piece of art in itself, and provides the user with a more visceral and “hands-on” way to do music.
- Yes, it is different. I can’t really put my finger on it, but there is some intangible quality not found in other products.
- Best free instruments for experimental music.

6.7.2.3.5 General or Specific?

We see our work with the ixi software situated experiments of personal interest, yet general in expressive scope; it is a subjective investigation into music and music making, taken from a highly personal and culturally conditioned standpoint. Considering the ideological and musico-theoretical outlook of the software designer, the question of how the users perceive the software in terms of generality or specificity becomes obvious:

Would you say that the software is general or specific? Discuss.

The results were surprising. Half of the answers stated that the ixiQuarks were a general environment for musical expression. Of course, when the effects, the utilities and the filters are considered, the software is quite apersonal with general scope, but the instruments themselves are considerably limited and constraining, to the degree that they cannot be defined as general anymore. They are strongly inscribed with music theory and specific logic. Furthermore, what might have deflated this question is its vagueness in design with regards to whether it is asking if the software is general/specific in purpose or general/specific in expression. An extract from the replies:

- Well both in some ways, it fuses a lot of common and experimental stuff together.
- I'd say its general as it can be used for a whole host of things.
- I would say it is general, it covers many areas of sound creation.
- Specific with regards to the processes and actions available, a slight understanding of basic synthesis terms and methods, also with using audio samples
- I think it's general in that it has a lot of uses. Perhaps it is specific in terms of the type of music or sounds you'd have to be interested in creating. I suppose it is also specific in terms of the experience of using it – the user would have to be interested in experimental compositional and sound generation tools.
- Both I think. The textures can be applied to a number of different styles, but some of the instruments seem a bit rigid. I need to spend more time with it.
- Specific. It's oriented toward experimental musicians (which is great).
- Working within constraints is refreshing. Music is so big - of course ixi is specific.
- I think both terms apply. General - there are many tools that have many functions and the sounds generated can be relatively 'normal' or can get pretty far out. Specific - each tool has a character of it's own and can manipulate the sounds in a unique way.

6.7.2.3.6 Question Concerning the "Script" of ixiQuarks

If the ixiQuarks are specific in scope, if they are coloured by the musical ideas of its author to a certain degree, how do other users relate to that? As covered later [in sections 7.4 and 7.5] there is a subjective difference in what can be defined as a composition and what as a tool. Often the ixi instruments begin as a composition, as an exploration into a musical problem, but if I see that the composition is considered to be general enough for other people to use as well, an effort is made to depersonalise it and a user interface is created for the musical pattern. The following questions were intended to get the participants to express their relationship with the epistemic tool, how much it colours their thinking and musical activities:

Does the ixiQuarks environment affect your compositional or performative patterns? Do you think differently about your music? Does the environment encourage new directions in your music? If so, how?

The replies illustrate that most people found that the tool influenced their musical activities although many also claimed that it did not. In fact, some people claimed that it does not influence their thinking at all, as they were thinking this way before they started using ixiQuarks and that was indeed the reason they use the environment. I believe this question needs a further investigation as this is a complex topic. (User interviews [see section 6.7.3] showed that artists tend to want to appear to be in control of their creative processes and not lose their authorship to a tool).

- It makes me think more about microtones for sure.
- It encourages a totally abstract way of thinking which can only heighten your compositional process.
- I'm an improvising/spontaneous composition type of performer... I find the tools open up a world of possibilities for sound manipulation.
- Any software, in my opinion, restricts the composer, though this is not always a negative. I've also found that software cannot make up for lack of artistic vision and also, for audio synthesis and processing, good source material, live or pre-existing.
- No, ixiQuarks is only a tool for use.
- No. I want to be in charge of things.
- In my case was the other way around, I had very clear in my mind what I wanted to hear but needed the tools to manipulate sound with such freedom.
- I do think differently about my music now - and it does absolutely drive me in a whole realm of thinking - so it does both effect my compositional and live performance.
- Yes! It is NOT a sequencer! It is more focused on my inabilities than on my skills.
- Working within constraints is refreshing.
- Yes. it makes it difficult to just add...drums...bassline...it makes you LISTEN more....
- Sure, it opens up for new ideas. Different ways to conceptually see problems.

6.7.2.3.7 Embodiment and ixiQuarks

Considering the focus of the current research on embodiment, virtual embodiment and the phenomenology of musical instruments in general, it was germane to ask the question of how the users of ixiQuarks experience themselves as embodied beings when using the software. In version 5, the software does not support any gestural controllers apart from the mouse, the keyboard, a pen tablet and rather limited MIDI control (where the effects, filters and spectral effects can be controlled, not the instruments themselves).[152]

Regarding embodiment and bodily control over music: to what extent do you think you use your body to control ixiQuarks? How is it different from acoustic instruments? Or other software?

The replies were quite interesting, at times humorous. Some found the question ridiculous, as "we don't use our body when using software," others did not understand the question even, as, arguably, it is of a rather academic concern.

- It's not the same as beating on drums but I find I can get to a space where I feel the instruments/tools are expressive... I groove along quite physically while I'm playing/improvising.
- I only use my fingers to control my track-pad on my laptop. Acoustic instruments allow for a much more direct relationship to the melody and rhythm.
- As with any software, bodily control remains rather academic, but youngsters seem to

[152] Although, obviously, people with SuperCollider knowledge can write their own support for any interface they like, so in that sense, the software is completely open.

think different. For me my keyboard gives more bodily control than fumbling with mouse on scale synth. In general, I think bodily control needs a hardware controller, indeed something like an acoustic instrument.
- Acoustic instruments will always be the best choice for musicians who desire god-like control over every sound, but this software comes as close as software can to giving the musician a similar feeling of god-like control.
- I feel like this is controlled more by creativity and the mind. It seems a bit tougher, but more pure, more true.
- I use my body to sit on a chair. It is a software!
- I don't understand the question.

6.7.2.3.8 Frustrating Constraints

The question asked here was:

When using the ixiQuarks, do you often find that the software does not allow you to perform certain basic musical things? What is lacking? If not, are these limitations perhaps inspiring?

Considering that the ixiQuarks do not provide many of the basic audio workstation setups, like timelines, how do the users of ixiQuarks engage with the musical ideas of the software? Of interest here is whether constraints result in a feeling of liberation (from too much freedom) or of limitations (of too little scope of personal expression). This question is therefore an appropriation of the question in section 6.2.6.3 specifically to the ixiQuarks environment. Interestingly enough, people were not calling for timelines, step sequencers, tonal or harmonic instruments,

- Not enough opportunities for control and personalization. This is the reason that, as it stands, I would never use ixi for music I would claim as my own - the instruments, to varying degrees, always sound like ixi instruments.
- I cannot find any limitation.
- Sure you can't do everything on that software. But it is only a software, not a miracle. And it does very well (and even better) what his purpose is.
- ixiQuarks steers away from straightforward composition and playing- i.e. no keyboard, chromatic scales etc.
- I use ixi in its own philosophy and it's good for me.
- Limitations in positive way can be inspiration.
- I think that ixiQuarks is for a non-usual music, is a way to find some interesting elements for musician. If you want play the melody - you can choose from thousand programs.
- Naturally, because the software's strength (to me, anyway) is to create novel ways of approaching sound and music creation.
- I could say sequencing but that exactly why I find it so interesting, because it's not based on a time grid or anything like it... Continuous results that emulate the performance of an acoustical instruments.
- Most of the limits are mine and not the software.
- Every software has its own philosophy, and so it's specific lacks which makes it sometimes special and sometimes inspiring. That's why I always look for different software, which gives me new ideas of creating sounds.
- I have other software for more conventional music. It's not an issue.

People seem to embrace and enjoy the constraints of ixiQuarks. The instruments are open enough making the tool attractive for personal expression. The users do not find that they are expressing the music of the ixiQuarks designers but their own. This means that the system, although presenting rigid constraints, allows people to engage with it on personal terms.

6.7.2.3.9 (Un)Familiarity of ixiQuarks

Personal background is important when choosing a technology to work with. Operating system, sound cards, software packages, tools, programming environments; all these things have specific qualities which are hard to fit to an objective evaluative scale. People choose their tools by all types of reasoning and some technologies fit some personalities better than other. We were therefore interested in knowing how familiar the design ideology of ixiQuarks (its interaction model – [see section 6.3.3]) was to the users of the software and if it was hard to understand it initially. We were interested in knowing if people with backgrounds in video games and web use found the design intuitively understandable.

Do you find ixiQuarks easy or difficult to understand? To what degree do you feel that your way of thinking matches that of the designers of ixiQuarks?

- Very easy to understand. Its design is very close to my mode of thinking and working.
- On the beginning it was a little bit complicated. But when you start to use it, it is a very simple structure with very complex instruments and effects. That's satisfying.
- Quite difficult, essentially because I'm French.
- For me, it is easy to understand and difficult to master, and there are always improvements to be made.
- Ixi is complicated because many things are preset, some things are not. Ixi is a strange mix of theoretical academicness and intuitivity which may indeed work for some.
- It has a learning curve, for sure, but I think that it is infinitely easier to grasp once one has used some of your simpler software.
- It's easy when I JUST use its interfaces, but if I use them as SC classes then it's difficult.
- to some aspects yes. I would need a step-by step guide to get started. Too many things are assumed.
- Easy to understand. i feel that the designers are tapped in on many levels, my favorite is that they are offering their labor of love for free, and are really into feedback.
- I love the fact that there are tutorials and video representation. I would agree to some degree that my way of thinking may match the developers though one may never know. I know to some degree . . . Absolutely.
- I feel like this software was made for me.

It seems that people understood the environment rather quickly after having used it for some time. Again, this is a question of engagement versus boredom, which has to be balanced clearly in digital musical instruments [see section 3.6.1]. A user interface so simple that one immediately understands it, will hardly serve as an engaging musical instrument. The questions of design relevant here can be concretisised into the simple comparison of playing music versus reading an e-book or filling in a spreadsheet. The act is not passive, nor is it one of production; it is a personal expression that typically involves some kind of embodiment, whether virtual or not.

6.7.2.4 Conclusion

There were more questions in our survey, but they were specific to the design of the software, on positive and negative features, what other musical software people use, knowledge of programming, and what is lacking in ixiQuarks. The replies to these questions were very useful for me concerning future development of the software, but they are of little academic value here.

The quantitative questions indicated clearly that people are generally happy with the software and use it on its own terms. People were less enthusiastic about some areas of the software, like the spectral plugins or the documentation, but showed much affection for some of the instruments. A feature that might partly explain the general positivity towards the software is that the ixiQuarks are free and people might have different expectations and demands to free “gifts” than with commercial software in which they have invested.

The survey demonstrates clearly how well the software suits some people in both their compositional process and performance practice. One can conclude that the goal with ixiQuarks – to be an improvisational environment for live electronic music that is fast enough to play with acoustic instruments – has been achieved. The survey showed that people are happy with using it only as a screen-based instrument with minimal gestural input, but also that there is a strong interest in gestural interfaces and embodied musical expression. It shows how software based on personal investigations can benefit people from all over the world and how limiting the conventional and commercial software strategies really are. There is much more room for explorations in the design of musical software. Additionally, the survey also shows the power of the internet as communication phenomenon. The development of ixi software would hardly have happened before the internet, if we were simply distributing the software locally to friends. It needed the thousands of downloads from all over the world, supportive feedback and the establishing of discourse between us (the designers) and the users to make obvious to us that it was a worthwhile activity to be engaged with.

As mentioned before, and like any creative explorations into unknown domains, the development of the ixiQuarks have been a highly personal journey. They were written without users or particular musical styles in mind (apart from aspirations for good usability of course). I therefore believe that the instruments are limited and constrained tools that necessarily present a specific picture or attempt to a certain musical problem. The software is not general in expressive scope nor does it attempt to be. However, the survey participants were equivocal on whether the software was general or specific as people found that they could perform a great variety of things with the tool, yet directed by its functionality. With the regards to the "script" of the technology, most participants agreed that the software has a style or scope that encourages them to take certain paths in their musical composition (such as working with microtones, the aleatoric, automated performance, and the focus on timbre rather than notes). The participants found ixiQuarks well suited for musical improvisation and some even claimed that it was the "heart and soul of my improvisations" – a statement that makes the mission accomplished for the designer of ixiQuarks.

Documentation of ixiQuarks exists for every feature of it, but it could be improved. From experience of distributing software for nine years I can confidently assert that, people do not generally like to read help files or documentation. Considering this, it was important for me to know if people understood the interaction model that is used in ixiQuarks. Happily, people found it very easy to understand, after the initial hurdle of gaining that understanding. This means that the interaction model is generic with a model of interaction that conforms to Beaudouin-Lafon's polymorphism, reification and reuse [see section 6.3.3.2].

6.7.3 User-studies Through Interviews

The user evaluation of ixiQuarks also involved user interviews that were recorded (sound only) and analysed. There were six participants who were interviewed for ca. 50 minutes. Four of them were experienced musicians, two were visual artists that did not consider themselves as musicians, although they worked with sound. The control of the set-up was: 20 minutes interview that focussed on general questions of software use for music creation; 20 minutes a "think-aloud" session (Rubin 1994; Tullis & Albert 2008) where the participant tried ixiQuarks and explored its potential for musical creation; and finally a 15-20 minute interview where questions focusing specifically on ixiQuarks were asked.

These interview sessions gave qualitative data corroborating the results received in the online surveys. As interviews are more time-consuming, in-depth, and enable supplementary insights, there were certain nuances which the interview technique provided compared to the online surveys. Some of those additional points will be addressed below.

In the general interview, the discussions extended from acoustic instruments to general digital audio workstations (like Logic [see Figure 7.4]) to more constrained instruments like the ixiQuarks. The participants found that digital technologies do set the user up with an agenda. An initial idea that is executed through a particular *new* software might become very different due to the way the software colours the work process. Through realising that musical idea in the particular software, the tool is learned and its inscriptions incised. The musician thus learns to think in that particular software and from then on novel ideas will be framed within the context of that tool. It was agreed that this process is different in acoustic instruments as they are approached from an embodied practice and the instrument will have been explored and learned sufficiently before one starts express oneself through it. It is thus more a question of training the body into a particular musical culture, which will reciprocally feed the musical creativity. However, there is a similarity with regards to *composers* of acoustic music, as they have to contextualise their composition/musical ideas with regards to the instrument chosen. The consensus was that when software acknowledges its constraints and limitations for expression, such as in simple trackers, it is a joy to work with. The situation changes if the software claims broad generality and "all-purposeness," leaving the user feeling bereft and frustrated when stumbling into its limitations.

The think-aloud session proved an invaluable method of user testing. Traditionally, in our work with ixi, we have demonstrated the software and helped people using it, gotten verbal or email feedback, but the quiet observatory stance required in the think-aloud method was extremely educative as a software designer. Here, users stumbled into problems that they would have had to read the manual for solving, but the think-aloud did not allow time for that. This was considered a design fault in many cases, such as the (then) impossibility of double clicking an instrument name to load it up (there is an open button and one could press the ENTER key to launch it). In general, however, people found the ixiQuarks intuitive and after the initial problems of understanding the environment, they proclaimed that there was a consistency in its design that made it easy to understand. A hurdle in the learning curve of ixiQuarks were the concepts needed for digital audio. As this system is lower level than most commercial software in terms of digital signal processes, it provides access to features that are normally hidden. For example, understanding what buffers are (an array of sound samples stored in the random access memory of the computer), the bus system for routing audio, and the function of *many* of the synthesis algorithms used.

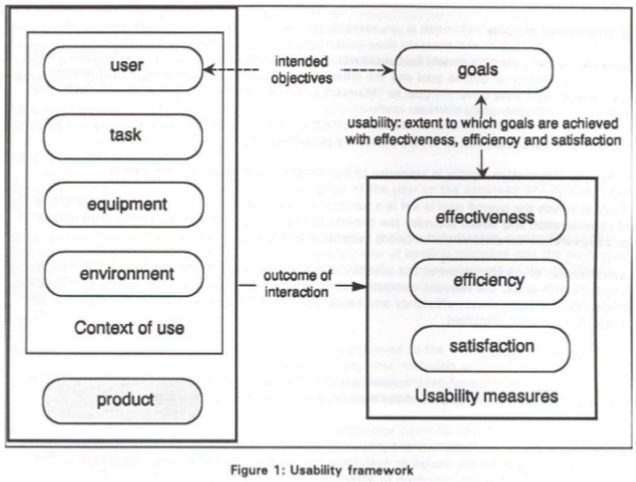

Figure 6:30. Taken from International Standard Organisation (1998) section 5.3.1.

A topic that repeatedly came up in the interviews was the resistance artists have against being defined by their tool use. Most artists seem to abhor by the idea that large parts of their creativity might derive from a tool. Engineers or software designers seem to have other ideas regarding the role of creativity in their field [as seen in section 7.5]. They are perhaps less conditioned by the Western art tradition that often seeks more transcendental or romantic explanations of creativity (Weber 1992; Meyer 1994; Bürger 1984). This would be an interesting topic for further research, but it is of too marginal relevance for the current study.

I will not elaborate on these interviews further, apart from mentioning that as a usability technique, interviews mixed with think-aloud technique are strong methods to use not only in the early stages of systems design but at all times. At the stage ixiQuarks are currently, usability comments and criticism has been received for some years through online feedback and bug report mechanisms, which has smoothed out all major problems, yet I found the interviews enabled me to understand issues in the design of ixiQuarks that were hidden from view.

6.7.4 The SUS Test Applied to ixiQuarks

Apart from learning about people's experiences of using the ixiQuarks, I was interested in a more objective and general usability testing of the system. Usability is obviously not something that exists in an absolute sense, and it differs hugely according to which areas of application the system is made for. The International Standard Organisation's ISO 9241-11 document on usability with screen-based computer systems defines three areas of usability measure:

- effectiveness (how easy it is to use the system, and the quality of its output)
- efficiency (how many resources the system takes)
- satisfaction (subjective reaction to the system)

A few usability evaluation questionnaires exist that aim at objective evaluation methods of computer systems, most taking consideration of the general areas listed above in Figure 6:30. In a response to the requirements of the above, Brooke (1996) developed the SUS, a simple usability scale system for general usability testing of various systems. SUS has proved to be a valuable system-independent evaluation tool. Brooke states that it correlates well with other subjective measures of usability, such as the general usability scale of the MUSiC project (Kirakowski 1988).

I chose to use the System Usability Scale (SUS) questionnaire above questionnaires such as Questionnaire for User Interface Satisfaction (QUIS); or Computer System Usability Questionnaire (CSUQ). The reason is its independence from specific applications (such as design for the web, mobile phones, video games, etc.). Furthermore, as Tullis and Stetson (2004) show, it is one of the most simple and effective usability questionnaires available.

The appropriateness of this scale in the context of ixiQuarks was clear: it was general and easy to conduct as parts of other surveys (such as interviews). Following the SUS requirements, the survey was taken after the participant had been briefed about the functionality of the system and had tried it for considerable amount of time. However, the test was taken before any further introduction or tutorial sessions with the software developer/survey conductor.

Each question could be answered by ticking boxes from 1 to 5 where 1 is "strongly disagree" and 5 is "strongly agree" (the Likert scale). The means and standard deviation for each question can be seen in Table 6.1. The questions are ordered such that each alternate question is negative, which lessens the likelihood of rote replies. The SUS scale has clear instructions how to interpret the data and below are the results:

SUS Questions	**Mean**	**Dev.**
1. I think that I would like to use this system frequently	3.95	0.83
2. I found the system unnecessarily complex	1.55	0.60
3. I thought the system was easy to use	3.7	0.57
4. I think that I would need the support of a technical person to be able to use this system	2.1	1.02
5. I found the various functions in this system were well integrated	4.3	0.47
6. I thought there was too much inconsistency in this system	1.6	0.5
7. I would imagine that most people would learn to use this system very quickly	3.3	1.03
8. I found the system very cumbersome to use	2.05	0.83
9. I felt very confident using the system	3.65	0.75
10. I needed to learn a lot of things before I could get going with this system	2.35	0.99

Table 6:1. Descriptive statistics for individual SUS Questions.

SUS scores, after having been calculated according to the appropriate algorithm (Brooke 1996), have the range of 0 (extremely poor usability) to 100 (excellent usability). This algorithm is not directly linear as every other question of the survey reverses the positive/negative meaning of 1 and 5. From this we can derive at the general SUS means of the system, see Table 6:2.

Mean	73.125
Standard Deviation	10.634
Range	35
Minimum	52.5
Maximum	87.5

Table 6:2. Descriptive statistics for the SUS score.

Good applications typically score between 60 and 70 while bad applications score between 40 and 50 (Brunelly et al. 2007: 60). From this we can derive that the ixiQuarks have been successful as both musical environment and as an experiment in HCI. It shows that our interaction models have proven to be understandable and useful in musical performance and composition. Obviously, section 6.7.2.3 gives more qualitative and informative information, but it is useful to know from a rather more objective measuring scale (the SUS test) that the software works effectively as a product for musical expression.

6.8 Conclusion

This chapter described the digital musical instrument primarily as a system of algorithmic structures that define users in their use of the tool. It explored how interaction models and design semantics can establish a coherent system of virtual objects to which the user can have an epistemically mediated embodiment; one in which the awareness of the objects' arbitrary functionality can be changed at any time. It was also described how such interaction models derive from a personal, yet consistent, system of musical thinking, and entice users to understand the semiotics underpinning this ideology. The ixiQuarks are the result of the evolution of such interaction models. They were introduced as the system used for the exploration of the theory of epistemic tools, in the field of digital musical instruments, and are the outcome of years of experimentation and development, focused into one musical system. Through extensive user studies: an on-line survey, interviews, teaching, and a SUS usability test, a broad overview of ixiQuarks' reception was gained. It was proved that a highly personal form of musical exploration, in terms of building instruments for musical expression, can benefit the general public. The feedback proved that the goal of this software, to facilitate live improvisation with digital musical instruments, has been successful.

Teaching, workshops, collaborations, show cases, user studies, interviews, and online surveys have all been useful ethnographic methods for studying the relationship between the designer and the user of digital musical instruments. Much of the ixi software (pre-ixiQuarks) is a result of a long and intensive process of participatory and situated design, often through extreme programming methods with my colleague Enrike Hurtado Mendieta, or in workshop contexts. The development of the ixi software has made extensive use of contemporary methods in HCI, both by accident and by design.[153] This has resulted in minimising the software's eccentricity (although, by choice, not entirely) and maximising its usefulness to general public, although within a strongly constrained scope, as already explained. The ixiQuarks originate in this context. They build on the design language, semiotics, and interaction models developed in previous work but make those more focused and concrete through more methodological design.

[153] We were doing extreme programming, situated and participatory design long before we knew that these existed as methodologies. However, as theoretical methods they have been useful to this work and yielded a higher level of self-criticism and awareness than before.

The ixiQuarks have been used for teaching in higher education context and feedback from the teachers of those courses has been invaluable.

In section 5.7, I illustrated with a practical example how the relationship between the programming language and the designer of creative software can be symbiotic, where the affordances of the language inspire and engage the designer in ways impossible to obtain if not through the "material" qualities of the programming language (its classes, algorithms, methods, etc.). The programming language becomes the coder's *material*, it has properties and outlines for procedural events, directing the designer to think in particular ways. As discussed in section 4.3, computers are good at fast calculations, repetitions, rigid timing, following rules through conditional statements, etc. They prime the designer to think in ways that work with the code's materiality of rational divisions, conditional statements and abstractions. Designing differently means fighting the materiality of code, which is perfectly possible as well. The next chapter explores these issues more profoundly, where surveys with users and designers of audio programming languages shed light on what might seem to be a unusual way of making art, i.e., coding instructions for a machine to follow.

Chapter 7

Designing Digital Tools: Instruments as Systems

After the investigations in the preceding chapters, it is now possible to revisit and reinforce the concepts of epistemic tools and virtual embodiment. Digital musical instruments are seen here as epistemic or semiotic machines onto which we delegate parts of our cognitive process, but, as opposed to provisions in Andy Clark's work or in enactivism, the designer's role in these machines for thinking is emphasised. External technologies are recognised as cognitive engines that influence and inform the thinking of their users. The above perspectives are then succinctly investigated in two surveys, the former with users of audio programming languages, and the latter with their designers.

7.1 Introduction

The first part of this thesis explored the relationship between technology and culture [chapter 2], technology and the body [chapter 3] and how computer technologies afford the creation of complex epistemic tools [chapter 4]. This part has focused more on the *practical* implications of designing digital tools. I extracted a general outline of the phenomenological and epistemic "nature" of digital technologies, as opposed to analogue tools [chapter 5], and illustrated how the aforementioned theories have been implemented in the design of the ixiQuarks [chapter 6]. In this chapter I am interested in exploring the question of *designing* digital tools. What does it mean? How does it differ from other types of design, and what can be learned from the creators of digital musical systems and audio programming languages? With this in mind, two surveys were conducted:

- A survey with *practitioners* of audio programming. These are people that typically use open source programming languages to express themselves artistically through code.
- A survey with *designers* of audio programming languages. An email questionnaire was sent out to the main language designers in the field. Of interest were the views these creators have of their own work and how they relate to the culture of usage that establishes itself around their creation.

Before interpreting these surveys, it is important to engage with the digital musical instrument in the context of the first part of this thesis. If, in Part I, chapter 2 was a theoretical excursion in the philosophy of technology (or the ontology of tools); chapter 5 was a more concrete and practical engagement with those ideas as they appear in the context of digital musical systems. Furthermore, in part I, chapter 3 focussed on embodiment and human-machine relationships (or the phenomenology of tools) and I intend, in section 7.2 of this chapter, to engage with the way digital technologies provide us with the possibility of virtual embodiment. Finally, in part I, chapter 4 introduced the theory of epistemic tools (or the epistemology of technological artefacts) and section 7.3 of this chapter deals with digital musical instruments as digital tools [see Figure 1:7 of thesis structure]. I am then able to finish the chapter with the exposition of the above mentioned surveys and put their findings into context with the whole theme of this research.

As shown by technomethodology [section 4.3.2.5], the choice of a system will affect work processes and the generation of ideas. This is particularly relevant in the field of creative technologies in the arts. A simplified account of this process could be outlined like this: the musician defines the task and then chooses the best instrument to perform it with.[154] This instrument necessarily provides all kinds of affordances and music theories to work with. It provides patterns of work, which the musician has to follow in order to get the job done. When the musician has finally learned the instrument well enough for the creative task, the project will have become redefined as an effect of the process of learning the tool. The tool affects the expression. The musician has now learned to think in terms of the tool; it has become the cognitive extension of the mind in which the musician represents her musical expression.

Character traits, and often chance, determine which musical systems a musician chooses to use. Just as with acoustic instruments, this decision is rarely based on strong rational calculations, but rather influenced by cultural context, economical factors and the technology at hand. When a relationship with some software tool has been established, we witness how musicians become programmed through the complex mechanisms of engaging with the tool's scripts. However, as covered below in the exegesis of Barry Eaglestone's work [section 7.3.4], different minds work in different ways. Section 6.7 presented an empirical study of people's relationship with the ixiQuarks, but a general psychological study is needed of how technology colours the creative process, although some work has already been done in this direction (McCullough 1996). It is important that this study is empirical and *in situ* through ethnographic accounts as people are prone to rationalise their actions according to the culture of practice they participate in or the categorisations inherent in the discourse they inhabit. Lucy Suchman (1987) has pointed out how fuzzy our activities are in general and how the attempt to conceptualise them often leads to a skewed picture of the actual process being described:

> [O]ur actions, while systematic, are never planned in the strong sense that cognitive science would have it. Rather, plans are best viewed as a weak

[154] I say "simplified" as the musician rarely "defines" a task and "chooses" the instrument through such a rational process. In reality, people typically start playing instruments at a young age where there are haphazard reasons for why an instrument is chosen.

> resource for what is primarily *ad hoc* activity. It is only when we are pressed to account for the rationality of our actions, given the biases of European culture, that we invoke the guidance of a plan. Stated in advance, plans are necessarily vague, insofar as they must accommodate the unforeseeable contingencies of particular situations. Reconstructed in retrospect, plans systematically filter out precisely the particularity of detail that charaterizes situated actions, in favor of those aspects of the actions that can be seen in accord with the plan. (Suchman 1987: ix)

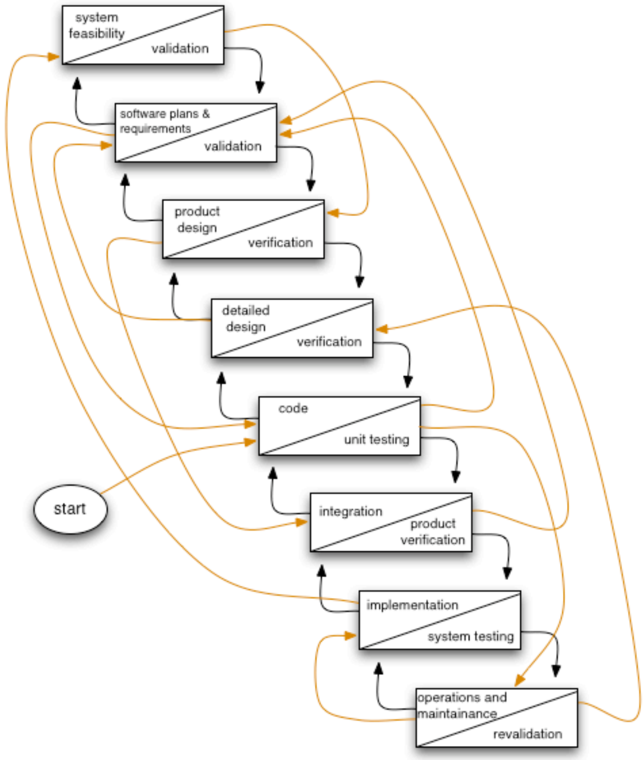

Figure 7:1. The black lines show the traditional "waterfall" software development model (Winograd 1996), but this thesis sees programming as an artistic activity, following a fuzzier model, such as that represented by the orange lines.

The design of a system storing patient information in a health system, a spreadsheet, or an interface for a database (production tools) is a highly divergent activity from that of creating tools for creative expression, such as a paint program, a video editing suite or a musical software (expressive tools). While the software engineer designing the implementation of work practices in the health system can reside upon facts and plans from which she can design a conceptual or UML model of the system, the programmer-artist is exploring the material of code, exploring algorithms and various ways of getting the computer to perform novel, interesting and personal tasks. It is therefore much easier to generate a top-down schematic model of a software in traditional computer engineering, as contrasted with the situation in which computers (and programming languages) are used as expressive tools. Here the approach is typically bottom-up, as it is impossible to create schematic models of ideas that are not formed yet. These ideas become real *through* the process of explorative coding where the affordances of hardware, operating systems, programming environment, and classes are pushed to their limits [see section 5.7.1 for an example of such process].[155]

The aim of this chapter is to explore the creation of creative environments, i.e., *how* system designers (almost unfailingly musicians and/or artists themselves) devise a set of tools

[155] Some interesting work is taking place on the issue of creative coding (or artistic coding), see for example Fuller (2003) or Mackenzie (2006) for a theoretical account of production from the inside (programmers) or Casey & Reas (2007) for a more artistic and educational account.

that enables other artists to express themselves. The "how" in this question is very important as the system can range from being highly personal, idiosyncratic or stylistic to a more general and open environment. Different systems can be placed as points on this continuous scale. Many forces come into play and we will, in section 7.5, explore the context in which a number of audio programming environments come into existence. Furthermore, it is important to explore how the inventors of these expressive environments establish their tools in a social context, in other words, how their personal and often idiosyncratic inventions become innovations established in social practice.

The large topic of inventions and innovations is too peripheral for the scope of this thesis to be covered with any substance. Nevertheless, it is worth pointing to how actor-network theory [see section 2.4] describes innovation as shifting of energies and establishing of power structures. Latour sees technological objects as containing "built in users and authors." The user subscribes to the technology's script by taking it into use:

> The fascinating thing in text as well as in artefact is that they have to thoroughly organize the relation between what is inscribed in them and what can/could/should be pre-inscribed in the users. Each setup is surrounded by various arenas interrupted by different types of walls. A text, for instance, is clearly *circumscribed* – the dust cover, the title page, the hard back – but so is a computer – the plugs, the screen, the disk drive, the user's input. What is nicely called "interface" allows any setup to be connected to another through so many carefully designed entry points. (Latour 1999: 239)

Latour talks about anthropomorphism in interface design and technology in general and points out that etymologically, "*anthropos* and *morphos* together mean either that which *has* human shape or that which *gives shape* to humans" (ibid: 235). This comment epitomises the actor-network theory, as it shows that there never is a unidirectional programme of action from the human to the nonhuman or vice versa. Energies flow both ways and every case needs a grounding analysis; generalisations will not do (In section 6.4 a genealogical analysis of the origins of three different instruments was performed).

In the actor-network theory, innovation is understood as a process of changing energies between social and technical networks. Relationships are shifted between domains resulting in a new product that becomes stabilised. At stake are practices, identities, social and technical protocols, machines, material resources, and the ever-present power of the market. In *Science in Action*, Latour (1988) suggests that innovation is a rhetorical controversy where the claims of the innovator are accepted by other actors who, as a result, become enrolled into the network. This can happen through:

- attempting to include other people's interests in the development of the innovation (such as plugging into an existing market, gaining the belief of an investor, creating a user group with demands)
- persuading users that they need this innovation to achieve their ends
- creating the user group that *needs* the innovation
- convincing the user group that the technology is driven by their interests
- making sure that the innovation's success is attributed to the innovator and not the user group or competitors' lack of success in the domain field

In section 5.4.3 we explored how markets are created around new inventions of musical instruments. This is no trivial task, and one that requires a good product, beneficial economic context and strong effort in public relations. In order to contextualise the focus on creative software with regards to the theoretical issues in Part I, the next two sections [sections 7.2 and 7.3] will explore the role of virtual embodiment and epistemic nature in our expressive digital technologies. It then becomes possible [sections 7.4 and 7.5] to study how the creators and power-users of creative computational technologies relate to these questions of embodiment and epistemic tools, on the one hand, and the topics of invention and innovation, on the other.

7.2 Virtual Embodiment

Virtuality is the cultural perception that material objects are interpenetrated by informational patterns. (Hayles 1999: 13)

Large sections of chapter 3 focused on embodiment and the relevance this concept has to the field of musical instruments. A whole research field has emerged that studies the problems of digital musical instruments; problems that are related to engineering, ergonomics and phenomenology. In this section we will explore the relationship between the body and the digital instrument, and its unique qualities in comparison with the acoustic instrument. N. Katharine Hayles' quote above is pertinent: the epistemic tool does not give the perception that one is performing merely on a physical artefact, but one that is "interpenetrated by informational patterns," i.e., the result of the analysis and representation of what the software designers have defined as the problem field in their structured analysis of the musical task.

7.2.1 Introduction

An *interface* is a field of abstractions where two systems interact with one another. The word is typically used for the locus where a human and a machine communicate. The interface can be as simple as a two-state button or it can be multimodal: a mixture of different types of hardware used to input commands into a system that responds through the use of screens, speakers, motors and haptic feedback. The pressing of a button could result in the initiation of a complex machinery, but, by the same token, many complex actions at the level of interface can result in a small irrelevant systemic change.[156] The computer is a meta-machine with no *natural* interface, unlike physical machinery where gears, buttons and wheels are natural extensions of the mechanism itself. This fact is problematised when the computer is used for music, as we have innumerable arbitrary ways of representing an interface to the audio system of the computer. It could be anywhere from a simple play-button to a custom written class that encapsulates the digital signal processing of an audio unit generator. The question here is that of purpose: what intentional bandwidth do we – as software designers – give to the users of our system? What degree of control do we provide and which interface elements do we present as affordances of the system, or rather: what mechanisms are provided for users to delegate their thinking, and by which logic?

It is generally recognised that the more control parameters an acoustic instrument has, the harder it is to master. The sophistication of an instrument entails particulars of fine control designed through an iterative process that results in multiple versions of the instrument. Consider the difference in the learning curve of the kalimba or kazoo versus the violin or the piano (Jordà 2005). When designing a musical system on the *computer*, we are obviously concerned with the musical parameters of the composition or the tool, but we also have to decide which ones to make controllable to the user through some interface or another. In creating the interface, we decide upon the abstraction level and intensity of the system. Often it is quite arbitrary which parameters are made controllable, as such decisions could be dependent upon the piece, the hardware, or the specific group of people for which the system is made. A problem arises when the situation changes and the designer, or the user, of the system decides to change its internal variables. Not only would the piece be different, but the hardware would behave differently, causing regression of performance in the trained player and general distress by the user group.

The megalomaniac's dream musical interface might be one where all human gesture could be translated to music. It would consist of some amazing tactile, haptic-feedback, motion capture device that would map movement to sound. The problem with this notion is that of bandwidth. Making full use of the capabilities of this interface would require a dedication to embodied practice that is rarely found in the field of computer music. In fact, there exists a good example of such an endeavour, the *Meta-Instrument* (de Laubier 1988) that has 54 sensors, sampled at 500Hz in 16 bit resolution that can control trillions of parameters (de Laubier 1988, de Laubier & Goudard 2006). The designers, Laubier and Goudard, admit that this can create confusion as the control possibilities are "a billion more than the numbers of atoms in the universe," and that this creates a sensation of "vertigo" in the instrumentalist who seldom knows "where he/she is, and which musical space, which algorithm runs under the tip of the

[156] This was discussed in section 6.2.1 on mapping one-to-many, one-to-one and many-to-one.

fingers" (de Laubier & Goudard 2006). Instead of creating the situation of such bewildering madness, it is obvious that we are forced to face the dynamic nature of the digital epistemic tools and how their design is essentially a *process of defining constraints*.

In this section we are interested in the question of what it is we are creating when we design a musical system on a computer. It has become obvious that we do not want to limit ourselves by imitating the world of acoustic instruments. The history of computer music shows that its strength comes from the unique qualities of the computer as a fast and general number cruncher with access to memory and procedures to predict and determine the future. As opposed to humans, computers excel at calculations, complex pattern recognition, analysis and creativity from generative rules. The unique strength and innovative power of the computer in music lies elsewhere than in simulations. It can be found in its nature as an epistemic tool: a platform where we can think about music – and think about ourselves thinking about music – due to its logical and self-referential nature. An environment like ixiQuarks, built on top of SuperCollider, is in its own way a system of thought in which one can think about music; experiment, compose and perform. It is an example of a musical environment where musicians can externalise their thoughts, to sketch through the creation of musical systems that could become compositions (deterministic or generative), co-players (intelligent and adaptive) or instruments (for live or studio use).

7.2.2 Instrumental Relationships – Embodiment

The phrase that the acoustic instrument can become an extension of the body is well known. It reports of a possible relationship with an instrument through an *embodiment relation* [see section 3.2.4, and Figure 3:1 in particular]. The instrument is ready-at-hand and we ideally express ourselves through it in a state of flow. Conversely, the digital instrument can be generally defined as one that contains a pronounced *hermeneutic relation* to the world. It presents itself to us as a symbolic structure in the form of software that we have to start, tune parameters, get inputs working, etc. before we can start performing. Of course, digital instruments have tangible user interfaces – many of whom are highly sophisticated and intuitive – but the core of the instrument is a conceptual structure encoded in a programming language. The tangible interfaces are to a certain extent peripheral to the instrument. They could be swapped out with other interfaces, yet the instrument would be defined as being the same. The interfaces are controllers and it is only when combined with the logic of the software that they become an instrument. The performer relates primarily to the structure of the software and not to the physical sensors in her hands. The sensation is often of interacting with algorithms of adaptable parameters and variables as opposed to the rigid resistance found in the structure of physical materials.

Referring back to Figure 3:1 in section 3.2.4.2, we can define the NIME approach to the design of digital musical instruments as one that tries to pull the digital technology into the realm of embodiment relations. The aim is to create the condition where the performer feels at one with her instrument, where it becomes an extension of her body, thus fulfilling the performer's need for an embodied expression, but also satisfying the audience as people may then intuitively understand the relationship between the gesture and sound.[157] However, this has proved to be a difficult task. The sensors change over the years, the software evolves and operating systems improve. The performer (or instrumentalist) is dealing with an instrument that is most appropriately defined by its state of flux, of eternal change and adaption. Variables can be changed just before (or even in the middle of) a performance, resulting in a sensation of *alienation* in the performer. However, this alienation does not necessarily have to be one of estrangement, but rather of a novelty and freshness. As such, it can result in novel performance and avoidance of cliché-playing [see survey in section 6.2].


Figure 7:2. Leo Theremin and Waisvisz.

Michel Waisvisz, the late pioneer in digital musical instruments, is an interesting case of study. Unlike many of the researchers and engineers presenting their inventions at the NIME conferences, or the many ingenious inventors of the street-DIY cultures, Waisvisz was an active performer of electronic music. He used his musical instruments in all his performances, but interestingly worked with "frozen versions" of his instruments, persisting in the use of the same sound and mapping engine for longer time periods. Waisvisz knew well the problems of the symbolic nature of digital instruments:

> About my own experiences with gestural controllers I can only say that I fight with them most of the time. That's something that almost every instrumentalist will tell. But if you are in the position to be able to design and build your own instruments, and so many interesting technologies pop up almost weekly, you are tempted to change/improve your instrument all the time. This adds another conflict: you never get to master your instrument perfectly even though the instrument gets better (?) all the time…. The only solution that worked for me is to freeze tech development for a period of sometimes nearly two years, and than exclusively compose, perform and explore/exploit its limits. (Waisvisz 1999)

[157] This is a topic we have taken up in sections 4.4, 3.3.1.2 and 6.2.1, but consider also Cascone (2001, 2004) and Jensenius (2006, 2007) for two contrasting views on the dilemma of human gesture in computer music interfaces.

For Waisvisz the only way of mastering his instrument was to resist the temptation to change it continually (Waisvisz, personal communication in 2005). At the STEIM studios, he created a sophisticated controller called *The Hands* which was connected to their in-house sample based software *LiSa*. The Hands is a controller that uses various types of sensors (such as accelerometers, bend sensors, ultrasound and knobs) and sends that data as MIDI information to the sound engine. This sound engine is programmed in LiSa and is therefore a "project" or a "patch" that constitutes the instrument in the particular performance (and in Waisvisz's case for longer periods of time). Of course, he would switch between patches and thus get a "new" instrument to be controlled with his *Hands* controllers, but the argument here is that without deep embodied interaction with the instrument that establishes non-symbolic motor memory in the performer, there will never be a sense of mastery or perfection of performance.

7.2.3 Algorithmic Machinery as the Instrument's Body

In section 6.2.1 mapping was introduced as a defining feature of digital musical instruments. The theory of mapping is an important area within the NIME research field, focusing on how human gestures or indeed any arbitrary parameters are mapped onto sound. This research field studies the material, ergonomic, psychological and sonic mechanisms that are involved in this endeavor. Wanderley (2000: 66) defines the study of mapping as a twofold project:

- Mapping is a specific feature of a composition
- Mapping is an integral part of the instrument

Wanderley proclaims that he is mainly interested in the latter. His research (and of many of those working in the IDMIL Lab at McGill University, and of NIME in general) focuses on this issue. In this thesis we are interested in the merging of the two areas – i.e., of mapping as the conceptual engine of the instrument, the epistemic and musico-theoretical body of the instrument. The aim here is to emphasise the fact that mapping in digital musical instruments is essentially a process of composition. I argue that digital musical instruments contain increased compositional element in their functional mechanism than acoustic instruments, due to their conceptual origins. This is has been described as the epistemic nature of digital tools in this thesis. However, the approach has also been phenomenological, as the two parts (body and mind, interface and engine, phenomenology and epistemology) are never easily separated: it can be beneficial (for composers and performers) to view the digital instrument as one unity, but it is important to realise the contingent nature of the tool, especially due to its inscribed compositional elements.

As already mentioned, there is a tight coupling in *acoustic instruments* between the energy exerted by the performer and the mechanical interface that produces the sound. Some of that energy feeds back to the performer as a haptic response, where the performer couples with the body of the instrument through its material resistance and vibration (Evens 2005; O'Modhrain 2000; Serafin & Young 2004; Behrdahl 2008). The sound energy and the haptic energy are directly and naturally related to the energy exerted by the performer and the mapping of gesture to sound is implicitly written in the material's nature. In *electronic/electric instruments* (such as the synthesizer or the electric guitar) the origin of the sonic energy is electricity and not human biomechanical energy. As in acoustic instruments, there is a direct relationship between the performer's movements and the sound, but this coupling can be mapped in complex and arbitrary ways through the electrical mechanisms of the instrument (e.g., by switching the wires into a capacitor, a touch sensitive keyboard will emit a loud sound when pressed lightly and a soft sound when pressed hard). Nevertheless, the electronic instrument is still highly dependent upon direct energy-logical relationship between human action and sound. In *digital instruments*, this mapping becomes significantly more complex as one can switch behaviour on the fly, adapt it through an interface, apply artificial intelligence to interpret/map human energy onto the sound, or make use of adaptive audio effects (Verfaille et al. 2006; Zølzer 2002). By the click of a mouse, the tangible interface (a keyboard, joystick, data glove, etc.) can be turned into another instrument with completely different functionality. This fact posits mapping as the primary phenomena that *defines* or *constitutes* the digital instrument.[158] If the body of the digital instrument is a symbolic construction in the form of code, its *interface* is primarily made of a mapping engine (be it graphically represented or not), secondarily of the tangible hardware that plugs into the mapping parameters. The hardware can be exchanged without many complications, but a change in the mapping would result in a new instrument.

This complexity of mapping (the arbitrary relationship between gesture and sound) in the digital instrument can be frustrating for both the performer and the audience. Some researchers have explored haptic feedback in digital musical instruments (Chafe 1993; Marshall & Wanderley 2006; Smyth & Smith 2003; Birnbaum & Wanderley 2007) but in general the focus in tactile digital musical instruments has been on the problem of mapping and other ergonomical considerations (Ryan (no date); Wanderley & Battier 2000; Wanderley 2002; Paradiso & O'Modhrain 2003; Miranda & Wanderley 2007). Focused studies have dealt with the problem an audience faces when there is a lack of corporeal gestures that explain the sonic energies (Godøj 2003; Jensenius 2007; Leman 2008). However, those gestures do not have to be corporeal, as exemplified by livecoding where performers project their code/thoughts onto a screen as an invitation for the audience to participate (Collins et al. 2003; McLean 2004; Nilson 2007). The general situation is that the performer wants to feel in control over the instrument

[158] Ryan (no date) and Trueman (2007) explore these issues. Trueman finds room for ironic potential in the right musical contexts.

and the audience wants to be able to make sense of the circumstance.

Until now, I have resisted the temptation to define precisely what the "digital instrument" connotes. I have treated the term as signifying various things in multiple contexts, a stance perhaps best explained by Wittgenstein's concept of "family resemblance" (Wittgenstein 1968). It is not necessary to limit our definition as the epistemic tool and the digital instrument are used here almost as synonyms. They stand for expressive tools that contain much theoretical knowledge of the task domain, a property that analogue tools do not normally have. Wanderley points out that interaction in a musical context might mean many things, such as: instrument manipulation; device manipulation at the score level; other interaction contexts (dragging, clicking, scrubbing); device manipulation in the context of post-production activities; interaction in the context of installations; dance as musical interaction controlling devices; and, computer games (Wanderley 2000: 2). These activities are all inclusive in our definition of interacting with a digital musical instrument. The most dividing difference in these situations are in terms of *production* and *expressive* contexts [see section 7.3.2]. The production instrument is typically not intended for realtime use, whereas in the performative expressive context the musician is under the strain of realtime interaction where dexterity, control and mastery of the instrument is important. Here the performer demands an expressive and ergonomic instrument with no latency or faulty mapping design.

Looking at Figure 7:3, we can see how the performative digital instrument consists of a gestural controller, a mapping layer and a sound engine. It should be noted that not all features of the gestural interface need to be used. The mapping engine and the sound engine do not use all available parameters at all times; the use of control parameters and their semantic meaning can be set for each individual performance, piece or instrument. This describes well the digital instrument's nature as being in constant flux. Here the instrumental model is defined as the primary instrument; consisting of the mapping engine and the sound

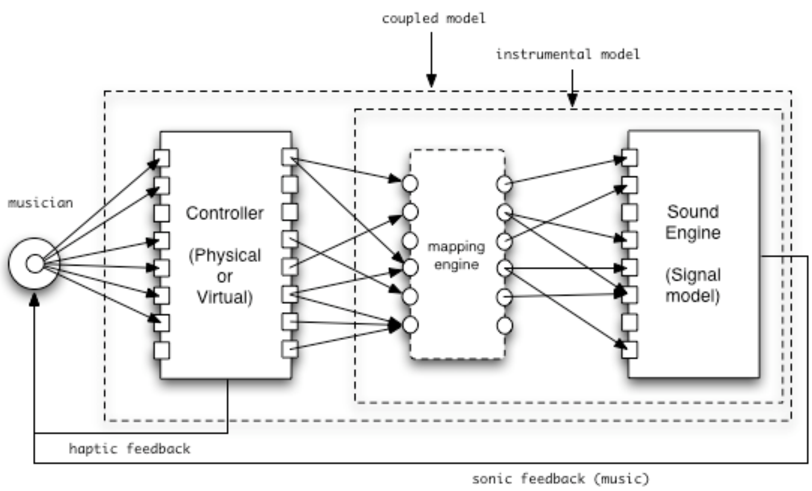

Figure 7:3. A model of the digital musical instrument. This model is related to similar models by Wanderley (2000, 2001), Wessel & Wright (2001) and Leman (2008), but adapted to the focus of the epistemic tool and emphasises the mapping layer and the instrument model as the sum of the mapping engine and sound engine

engine. This is where the character of the instrument is defined, and where the expressive scope and algorithmic functionality is set. The empirical question would then be which physical controllers would fit best to the functionality of the primary instrument.

It should be noted that the physical gestural controller can be hugely inspirational as a compositional factor "calling for" certain software based applications. Developing a digital musical instrument can therefore happen from two different approaches: one could start with a controller and create the mapping/sound engine with the controller's affordances in mind, or one could have a solid and expressive mapping/sound engine that calls for the making of a specific controller. A mixture of both is normally the case, where the relationship between interface and software is continual flux. The design of the digital instrument is never "finished." This does not mean that the gestural interface becomes in any way primary, as the performer is first and foremost interacting with the software engine (or the instrumental model).

As already covered, the structure of the mappings from interface to sonic parameters can be of three principal kinds: *one-to-one* mapping (as found in many acoustic instruments), *one-to-many* (typical for complex tasks such as flying, and in the case of musical instruments, this means a design of an epistemic dimension) and *many-to-one* (good for musical expression, where many embodied parameters define one complex sound). For Wanderley, the design of this type of instrument involves four different parts: a) the definition of gestures and their typologies; b) gesture acquisition and input device design; c) the mapping of gesture parameters to synthesis parameters; and d) synthesis algorithms (Wanderley 2000: 3). To these we could add the design of instrumental feedback; haptic, sonic and visual. This mapping of a bodily gesture can take the form of:

Filtering (time and frequency domain manipulation of an audio signal)
Sonic texture generation (layers of audio generated through synthesis or sampling)
Single musical notes (where pitch, envelope, amplitude and timbre is controlled)
Continuous feature modulation of both note and phrase (timbre, amplitude, pitch)
Musical gestures (glissandi, trills, grace notes, etc.)
Simple scales and arpeggios (of various speed, range and articulation)
Phrases with different contours (from monotonic to random)
Control of sampled material (loop-points, rate, granulation, pitch, filtering)
Synchronisation of musical processes
High-level control of recorded material (as seen with DJs)[159]

Inspecting the above list, we see that only first half is possible with acoustic instruments.

[159] Wanderley (2001: 53) covers few of these areas in his thesis.

Although the digital musical instrument makes use of all these controls, it is in the latter half where it excels. This is where we inscribe the musical instrument with our music theory, our models of what a musical performance should consist of, to the degree that we start to talk about "composed instruments" (Bahn & Truman 2001; Schnell & Battier 2002; Momeni 2005). What makes the digital musical instrument so profoundly intriguing for analysis in man-machine relations is that it constantly transcends all known boundaries. Mumford echoes Marx by pointing out that "the difference between tools and machines lies primarily in the degree of automatism they have reached" (Mumford 1934: 10); this distinction can be applied onto our digital musical instruments as well, but in the same stroke, the division has to be *rejected*; our modern devices of expression should be viewed as both tools and machines; as both instruments for manual dexterity and mechanisms for automation; both extension of our bodies and cognitive scaffolding for our mind.

7.2.4 Mapping as the Interface's Arbitrary Mechanism

In section 3.6.1 Csikszentmihalyi's ideas of flow were introduced. Figure 3:1 illustrated how the ideal musical instrument would provide a clear balance between challenge and skill. With increased sophistication in the way the musical instrument affords expression, the more it challenges the player and the longer it takes to master. The instrument has to engage the player and introduce in her the feeling that it has depth and mystery, which makes it worthwhile for the performer to spend time mastering it. Sergi Jordà (2004: 188) defines the *efficiency* of the musical instrument as a relationship between the output complexity of the instrument and the control input complexity. However, as some musical devices (such as the CD player) might have high output complexity but relatively low input complexity (play/stop buttons), another parameter needs to be added, namely the diversity control. The equation thus ends up looking like this:

$$\text{efficiency} = \frac{\text{musicalOutputComplexity} * \text{DiversityControl}}{\text{ControlInputComplexity}}$$

Jordà succinctly makes claims that a good instrument "should not be able to produce only *good* music" (Jordà 2004: 188) but also bad or terrible music. He is making the point that an instrument should not be strategically designed to output only notes in chosen scales, in perfect timing and with a beautiful timbre. He thus operates with a more confined definition of the digital instrument than we do in this thesis. In this work, we include installations, sequencers, toys and other tools that output music. To a certain degree, Jordà's definition effectively removes the digital instrument from its epistemic nature and "naturalises" it, such that it resembles the way acoustic instruments behave.[160] For Jordà, it is a distinction between *playing* music and *playing with* music, thus echoing frequently raised arguments that digital musical instruments render the performer an operator of the technology and not the active creator of the music. Adaptability (Arfib et al. 2005), or macrodiversity (as Jordà calls it) becomes an important evaluative factor as the instrument should be capable of playing different musical styles in varying contexts, just as acoustic instruments can. A good test for a digital instrument that aims to be expressive is therefore to ask if it can be used in most musical genres.

Chadabe (2002) takes an entirely different approach to the question of mapping. For him, mapping gestures directly to sound parameters (which for Chadabe equals the "limitations of mapping") overlooks the computational potential of the digital musical instrument. Chadabe, as opposed to Jordà's argument above, emphasises the epistemic nature of the composed instrument and he introduces the "fly-by-wire" metaphor used in aviation as an explanatory model of what the musician of such a composed instrument is doing in performance. Chadabe emphasises the interactive nature of the digital instrument and he defines interaction as "mutually influential" and thus different from instruments of deterministic control. For Chadabe, only electronic technologies afford interactivity to a degree worth the label. Examples of such instruments would therefore be Rowe's virtual listeners (Rowe 2001), Lewis's *Voyager* system (Lewis 2000) or Chadabe's own *Solo* (Chadabe 1997).[161] Instead of limiting the question of mapping to a specific ideology or methodology, I propose that we look at it as points on continuous scales as explored in section 5.6 and shown by Figure 5:14.

Mapping is therefore of an utmost importance: both ergonomic and aesthetic. For an instrument maker, the design of the mapping engine is a musicological, aesthetic and culture-political activity. It can influence which paths music takes in the future, what social structures will emerge and thus have important implications for society at large. For example, if we go back to the early 1960s we have Bob Moog in New York and Don Buchla in San Francisco both working with electronic oscillators and synthesis. Moog, the engineer, did not hesitate to add the keyboard to his sound engine, whereas Buchla, the artist, resisted the keyboard interface (he thought it had nothing to do with electronic sound generators).[162] The musical implications of these decisions are now clear. The Moog synthesizer became more popular due to its conformist

[160] We should note that this argument about Jordá's work only applies to this particular section of his PhD thesis, as it is clear that many of his instruments are highly epistemic, the reacTable being a good example. The aim here is not to reduce the complexity of his thesis into the simplicity of this argument.

[161] These works point to a rough distinction can be outlined between *playing* (a one-to-one control of an artefact) and *operating* (setting processes into motion, or engending movements that are partly or fully automated) instruments. A related distinction is that of tool and machine. However, I do not believe such divisions are prouctive in the current context, as what is really taking place is a blurring of domains.

[162] Buchla even rejected the term "synthesizer" for his instruments, as it implies artificially copying or reconstructing real instruments. For Buchla, the electronic instrument was an entity with its own sonic characteristics and he experimented widely with the design of gestural controller for those sounds (Pinch & Trocco 2002).

design whereas the Buchla synths were always more avant-garde. Experiments in tuning and microtonality that were popular in the musical culture of the 1960s were halted by this development. Popular musics have therefore been directly affected by synthesizer design.[163] The point is this: a seemingly innocent design decision taken at one point in time can have cultural and political implications, marginalising some cultures whilst centralising others. This can be clearly seen today in how dance music is the main focus of commercial computer music instrument manufacturers at the expense of, say, instruments for country or folk music. The reason is obvious: dance music is mechanical and repetitious, often devoid of the characteristics that exemplify human performance. The digital instrument is a production tool that affords the organisation of tracks (drums, bass, synth, etc.) in a linear and repetitive manner [see section 4.5.2]. Contrarily, folk music is performed live by a group of people that have mastered their instruments. This musical style would require digital musical instruments that were intuitive, flexible, and lending themselves to strongly embodied interactions.

7.2.5 The Body in the Realm of Metaphors

The virtual is not an opposite to the real, but rather an opposite to the concrete or the physical. Virtual musical instruments *are* real. They have interfaces (both tangible and screen-based), properties, and parameters that can be manipulated in realtime in order to change timbre or musical structure. As we saw in chapter 5, interactions with virtual instruments can be *physical-based*, i.e., using tangible devices that map human action to sonic parameters through musical-instrumental gestures such as plucking, stroking, blowing, scratching, pulling, pressing, hitting, etc. The device becomes a *simile* – something that is alike something else and works in the same way, but is essentially of a different nature. A sensor pad designed for hitting could be mapped to a drum sound, thus making use of a simile type of interaction. Here the interaction is a simulation of an original musical action.

Interfaces can also be *virtual-based*, most typically screen-based, but auditory (Frauenberger 2008) or haptic (O'Modhrain 2004, 2006) displays have been researched and implemented as well. Musical gestures in the screen-based instrument are typically implemented through mapping strategies that make use of *metaphors*. Metaphors are the principal device used in the interface design of modern computer systems (Erickson 1995) and they are used extensively in musical software. As Erickson points out, metaphors in computer design can be problematic, a view already stated by Halasz and Moran (1982) who claimed that they could actually be harmful. Halasz and Moran point out that there are three problems with the use of metaphors: the target domain has features that do not exist in the source domain; the

[163] The best and most obvious example is the synthesizer's incompatibility to the tuning of Arabic Maqam scales (where the octave is divided into 24 notes, or micronotes). We therefore witness performers of Arabic music constantly operating the pitch-wheel on synthesizers, practically limiting the keyboard control to one hand.

source domain has features not in the target domain; and some features work in both domains, but very differently. Metaphors can therefore confuse more than they help. Bertelsen, Breinbjerg and Pold (2007) assert that metaphors frequently make digital instruments too simplistic and thus limit the flow of ides for both the designer and the composer/musician using the device. Nielsen and Gartner in "The Anti-Mac Interface" (1996) point out that metaphors are overused in software to the degree that they dumb down or unnecessarily simplify operational tasks. Fuller (2003) amplifies this criticism by stating that "[s]oftware will need to be seen to do what it does, not do what something else does" (Fuller 2003: 100). In the field of music technology, Hamman maintains that "strongly referential interfaces can block creative activity" (Hamman 1999: 93).

Another representational mode – used extensively in the ixiQuarks – is possible where the mapping strategy is to make use of *metonyms,* where an element in the virtual interface represents an element in the conceptual (sound) engine, but it does so through contiguity rather than similarity (Jakobson 2002: 90). Here the object representing the sound, or, more importantly, the gestural action generating the sound (the mapped action), does not have to be based on any real or metaphorical object or activity. The object or the action can be derived from a new realm, inspired by anything or nothing. This is the sphere of interface poetry, of composing interfaces as art, of writing interaction as screenplays. For example, a sound can be represented as a box in the two-dimensional space of the screen. The "realness" and "embodiedness" of the situation derives from the sense in which the manipulation of the box affects its sound (the movement of the box to the left on the screen might pan the sound to the left speaker, moving it up might result in a higher pitch and increasing its size might increase its amplitude). Virtual embodiment can be defined as a trained interaction with a body (in this case the virtual body of the epistemic tool) whose properties are defined by their virtual physique and environmental context. However, this entrainment will always be liable to revision as the properties of the object are never real and based on material or physical necessity; neither is the mapping between the human gesture and the virtual object.[164] Again, the physical object is "interpenetrated by informational patterns" (Hayles 1999: 13), it is an artefact in an epistemically mediated embodiment.

For the designer of the digital instrument the main question regarding the character and nature of the instrument is to outline its affordances and constraints. Which control parameters does it have and how does the performer interact with them? Where are the expressive limitations of the instrument drawn, and what music theory underlies those limitations? As our survey showed, such decisions are double-edged: if the instrument is too general, it can be

[164] The person who retunes his/her computer's operating system such that mouse movement or keystrokes are different from the default settings knows intuitively what is meant here. The movement of a finger on the trackpad is not mapped "correctly" to the movement of the cursor on the screen. There is no ideal or correct function in this mapping.

uninspiring, whereas if it is too constraining, it can result in an early boredom. Brian Eno phrased this idea clearly a decade ago:

> The trouble begins with a design philosophy that equates "more options" with "greater freedom." Designers struggle endlessly with a problem that is almost nonexistent for users: "How do we pack the maximum number of options into the minimum space and price?" In my experience, the instruments and tools that endure (because they are loved by their users) have limited options. (Eno 1999)

It is clear that, for Eno, the constraints are inspiring.

> Although designers continue to dream of "transparency" – technologies that just do their job without making their presence felt – both creators and audiences actually like technologies with "personality." A personality is something with which you can have a relationship. Which is why people return to pencils, violins, and the same three guitar chords. (Eno 1999)

Here in a joyfully anthropomorphising mood, Eno talks about the relationship we have with our tools, and personalises the musical instrument. It is the focused instrument that provides clear constraints, complex enough to be engaging, yet not too difficult which would result in the loss of interest. The question of instrumental learning-curve becomes important. A learning curve can be defined as "the amount of time it takes a novice to gain enough skill with an instrument so the experience of playing it becomes rewarding" (Levitin et al. 2002). It is also a question of mastery. To master an acoustic instrument, such that one can play compositions written for it, can take over ten years (Lehman 1997; Nilson 2007). Ten years is a long time for a human to be in a relationship with a technological artefact. This is where designers of digital instruments bring forth a new attitude towards mastery: instead of the time consuming (and often rote, boring and frustrating) practice of the acoustic instrument, the digital instrument should be inviting, exciting and expressive enough to keep the player engaged with it for a longer period of time. The mastery of the instrument becomes a mastery of the conceptual, representational engine that underlies the instrument's body and a sophisticated understanding (based on experience) of mapping. There is a shift from a principally embodied mastery of an acoustic instrument to the mastery of the design, mapping and performance of the digital instrument:

> Traditional instruments are hard to play. It takes a long time to [acquire] physical skills which aren't necessarily the essential qualities of making music. It takes years just to get a good tone quality of a violin or to play in tune. If we could find a way to allow people to spend the same amount of concentration and effort on listening and thinking and evaluating the difference between things and thinking about how to communicate musical ideas to somebody else, how to make music with somebody else, it would be a great advantage. Not only would the general level of musical creativity go up, but you'd have a much more aware, educated, sensitive, listening, and participatory public. (Tod Machover, quoted in Rowe 2001: 5)

Machover, a leading inventor of digital musical instruments, brings us a clear message: instead of spending time on incorporating rote motor movements with the acoustic instrument, the musician could be listening, thinking and creating or adapting their musical systems for new musical ideas and performances. The presumption in new media technologies is that novelty is positive and tradition is conservative, a value that became centralised in 20th century modern art. Practising the virtual instrument is typically a conceptual activity where the performer engages with musical ideas from the level of timbre to formal structures. The practise is therefore essentially of a compositional nature, enlarging the creative role of the performer from being a mere interpreter to an active creator.[165]

I have no interest in passing judgement on the two highly different practices of performing with an acoustic or a digital instrument. Suffice to point out, in accordance to our findings from the survey presented in section 5.2, that the acoustic instrument tends to afford deeper embodied relationship, a continuity through time and therefore more dexterity or finesse in expression. The digital instrument, on the other hand, affords a constant re-interpretation of itself, a flexibility and the incorporation of compositional elements into its structure.

7.3 Digital Musical Instruments as Epistemic Tools

Informatics is a technique and a technicizing of language. (Stiegler 2009: 110)

Chapter 4 established how digital technologies can be viewed as epistemic tools. In this section I intend to analyse the epistemic nature of digital musical tools from the perspective of their design, in terms of both implications and applications.

7.3.1 Production versus Expression Environments

Traditionally, music software designers have seen it as their task to simulate the musical processes found in the physical world. They create tools with work habits analogue to those taking place when composing on paper, in the studio, and in the recital room. The graphical user interface is almost always a realist simulation of a physical hardware such as the stave page, the tape deck, the mixing board or effect racks. This makes it easy for musicians to move or translate their practice into the virtual realm. However, as we have seen in this thesis, this "translation" should rather be seen as transformation, as the working practices change

[165] Some composers and system designers have actively created systems for non-expert performers, such as David Tudor's *Rain Forest* and Cornelius Cardew's *The Great Learning.* Greg Schiemer (1999), Tod Machover (1989), Tina Blaine (2005) and Gil Weinberg (2006) have all developed interactive systems that effectively are musical instruments for the non-expert user. However, the argument here is not that systems should not demand in-depth knowledge and years of practice, it is rather pointing out how this training moves from the domain of embodiment to systems knowledge.

considerably and the conceptual and abstract design of the tools impose new work-habits and influence compositional thinking.

These digital audio workstations (DAWs) are, as Douglas Kahn (1995: 212) defines them, "word processors" for audio. Instead of the "phonographic" writing of the score or the midi sequencer, the modern audio workstation (with both note and audio recording capabilities) becomes a processor of sound. The composer does not write for other humans who will generate the sound, but instead writes directly into the machine that generates the sound either with virtual instruments or by recording audio signals. The work of composing becomes that of processing in a sense akin to how Derrida describes word processing activity by comparing it to writing with a pen:

> It's a different kind of timing, a different rhythm. First of all, you correct faster and in a more or less indefinite way. Previously, after a certain number of versions (corrections, erasures, cutting and pasting, Tippex), everything came to a halt – that was enough…. With the computer, everything is rapid and so easy; you get to thinking that you can go on revising forever…. All in all, it's getting a bit too easy. Resistance – because ultimately, there's always resistance – has changed in form. You have the feeling that now this resistance – meaning also the prompts and commands to change, to erase, to correct, to add, or to delete – is programmed or staged by a theater.[166] (Derrida 1995: 24)

Derrida claims that the transformation that happens in the shift from the pen to the computer is "economical, not structural" (ibid: 26). This argument is spurious when thinking about music technology, as evidently, when focusing on more complex technology than the word processor, the affordances of the technology are bound to colour the creative output. Pitch correction, time stretching, quantisation, tonality, meter, overlaying and re-taking are but a few obvious candidates that prove the difference from the score writing or the analogue studio to the more complex digital audio workstation.

The score is the most obvious way to represent music on computers but it has been pointed out that the focus on representing music at the score level can be questioned. Barry Truax (1989) asserts that the score is merely a representation of the music, separated from the sound itself, but also from the thinking that generated the score. "The problem is that the score is a description of the surface level of the resultant composition, not a guide to the musical thinking that produced it" (Truax 1989: 160). For Truax there is an underlying system of thought that could be seen as the music and computers could easily engage with that structural level as well. The score is a historical way of storing music, but with new media technologies, such as the computers, there is no need to store music as deterministic and linear score. Attali's (1985) point about the accidental nature of musical representation (and, indeed, its concretisation due to the demands of its commodification) since the middle ages is relevant in

[166] The resistance Derrida talks about is what I call *constraints* in this thesis.

this context.

The production tools have until now been limited to being score simulators. Their aim is to streamline the musical production process (see an example of such an environment in Figure 7:4). Referring to the epistemic dimension space [of section 5.6], they score highly on the *expressive constraints*, *music theoretical mass*, and *required foreknowledge* axes, but less on the *improvisation* and *creative* dimensions. All kinds of physical interfaces have been created to augment the limited input bandwidth of the mouse and ascii keyboard, such as MIDI keyboards, mixers, or other instruments. The software thus allows for writing music (scoring) through performing. The goals of these tools are straight forward and open: to simulate the physical environment in their interface design and to copy the studio work-processes in their interaction design. From a simplified perspective the virtual has replaced the real. But as the argument developed in chapter 4 shows, such a design always involves an abstraction, generic categorisations and normalisation of behaviour. The system is now a rigid and disembodied symbolic system that conducts the musician. It changes his thinking, behaviour, and introduces an epistemically mediated embodiment. Instead of a blind movement reaching for a knob on a mixer (a movement incorporated in motor memory) the producer now has to shake the mouse to find the cursor, peer at the screen to read the labels under the knobs, and make a mouse movement that rotates the knob.[167]

Figure 7:4. Logic Pro with virtual instruments.

It is not merely the manner of composition that has changed. Brian Eno, having spent half of his life in recording studios, and written about them as musical instruments (Eno 1983), reports on the changed social situation of the digital studio, of the politics of software, and of change of embodiment when the studio became digital:

> My engineer kept saying “Wait a minute” and then had to duck out of the musical conversation we were having so he could go into secretarial mode to execute complex computer-like operations. It’s as though a new layer of bureaucracy has interposed itself between me and the music we want to make....

---

[167] And rotating a knob in music software is not without complexities. Some software requires you to move the mouse in a circular movement, like the index of the knob, whereas other software wants you to move it up or down or even left or right. In some software this can be set, adding yet another layer of parametric knowledge between the gesture and the result.

> But now I'm struck by the insidious, computer-driven tendency to take things out of the domain of muscular activity and put them into the domain of mental activity. (Eno 1999)

Eno, with all his experience of studio work, is able to articulate the change that happens when the studio becomes an epistemic tool. Formerly the studio was a space where everybody knew what was going on – a space full of sliders, knobs, cables, and hardware of all kinds – where the musicians involved would understand the gestures of the producer/engineer when he moved a slider to increase the volume or tweaked a knob to increase the bass. Such an action was a social gesture, of a symbolic nature, understood and shared by everyone in the studio. An action (such as moving a volume slider) had a one-to-one, and thus intuitively understandable, relationship to the sonic result. In the digital studio, the politics of control change. There is only one person that controls the mouse, and the screen representing the "tools" or the "equipment" is so small that only those sitting next to it can *read* what is happening.[168] The recording engineer therefore gains an increased power, the knowledge of the epistemic tool, and this knowledge is hard to share through intuitive (and often theatrical) bodily gestures (Moorefield 2005).

7.3.2 Expressive Tools: Process Engendering Machines

As remarked earlier, musical software has traditionally taken an approach to design that simulates the physical equipment. There are various reasons for this, one being the relative ease in marketing the software ("look, the multi-track recording studio has now entered your bedroom inside a small computer"), but arguably also due to a more complex fact that has to do with our education systems, segregated work practices, and attitude towards creativity. When musical software became possible on computers, the people who were able to write musical software were mainly software engineers. This is an important fact as engineers and designers have very different attitudes to problem solving (see Winograd 1996).[169] Whilst the designer looks at the problem with openness to alternative solutions, with a horizontal scope into other disciplines, the engineer is more prone to define the problem as it is currently positioned and define, from a top-down approach, how it should be solved as a system. In the last decade there

[168] Literally *reading*. In [section 4.3] we looked at what happens when we switch form an analogue clock to a digital one. Instead of *seeing* the time, we now have to *read* it and calculate it. Extra cognitive load is added.

[169] Most of the authors in the volume edited by Winograd stress the difference between the ways designers and engineers look at the world. The designer takes a horizontal approach where she re-interprets the problem, and provides a new solution. The engineer's approach is more vertical, the problem is broken into logical chunks and then implemented in the system. Very little re-interpretation takes place. Denning and Dargan write: "The standard engineering design process produces a fundamental blindness to the domains of action in which the customers [we would rather use the word *users* here] of software systems live and work" (ibid: 107). Crampton Smith and Tabor write: "There is no logical reason why the skills of the artist-designer and engineer-designer cannot be combined in one person. A few people are totally ambidextrous in this way, and many are partly so. But the time needed to gain education and experience in either domain, not to mention the difference in temperament suited to the two domains, means that most designers specialize in one or the other" (ibid. 41).

has been a change in this situation, as people have grown up with computers, and programming has become a natural way to interact with them.

Production tools and expressive tools would both be defined as being of an epistemic nature. They define the cognitive and physical processes of their users. They embody traditions or cultures of work and production, that the user subscribes to or rejects, through various strategies. This distinction is loosely analogous to a distinction that Newton Armstrong makes between functional interfaces and realisational interfaces, following Andrew Feenberg's (1999: 202) theory of primary and secondary instrumentalisations:

> The functional interface (primary instrumentalization) serves a predetermined function; it is structured around a finite set of interactions which are known in advance of the task's execution. The well-designed functional interface conceals the specific mechanics of the task, and presents the user with possibilities for action that draw on familiar and often rehearsed patterns of experience and use. The realizational interface (secondary instrumentalization), on the other hand, brings with it the possibility of continuously realizing new encounters and uses, and, in the process, of re-determining the relationship between technical objects and their human subjects. The realizational domain encompasses the contexts of meaning and signification in which human and medium are embedded, and is conducive to dynamic and indeterminate forms of interaction. In short, realization is a form of play. (Armstrong 2006: 4)

We are getting a picture, not only of different design strategies with regards to the function and meaning of the tools made, but also of how musical instruments can be either characterised by interpretative openings or closure. In section 4.4.2.2, an account was given of the openness to be found in the material properties of acoustic instruments. This was contrasted with the abstracted, defined and explicit nature of the digital instrument. For example, for Simondon [see section 2.5.5], the technicity of a skin drum is higher than that of the digital computer system simulating the drum (Mackenzie 2002: 12). The skin drum is more general and can be applied in a greater number of contexts. The problem lies in the ensemble in which the technical artefact exists. The computer is a concretisation of various technical tendencies and therefore more entangled in a network of technical references, standards, productions and practices. The low technicity of structured technological objects in the form of digital instruments tends to bring a closure of their usage as opposed to the openings of simple acoustic instruments. The skin drum is also a concretisation of a complex network of technical tendencies, but the technical dependencies are much fewer. In a way, the drum is a simple tool that continually offers openings for usage, as opposed to computer software, which is a concretisation of an extremely complex network of technological practices that indicate a closure. The skin drum can be manipulated in various "unorthodox" ways, such as stroking, blowing, warming up (makes the skin looser and deepens the tone), pebbles can be inserted into its interior and then shaken, etc. The skin drum has infinite "interpretive openings," whereas the limits of software is always consciously designed and defined with attention to affordances and constraints. The computer system is one of "interpretative closure": instead of physics or mechanics of a wooden drum we are dealing with the conceptual engineering of a virtual device.

7.3.3 The Uprooting of Physicality and Singularity – An Evolution Towards Generality

There are composers here who swear by numbers, and those who swear at numbers -- tools, instruments (Austin 2009)

Sections 4.3.1, 4.5.1.2, 4.5.2 and 4.6, described the powerful processes of rationalisation, generalisation, normalisation and standardisation that take place in the design of epistemic tools. This has strong implications in musical instruments as the unique, the specific, and the eccentric are highly valued features of all musical performance. In a recent text, Pinch (2006) shows how the tradition of the synthesizer's emulation of acoustic instruments is one that evolves in a complex sociotechnical environment. As an epistemic tool, the synthesizer strives at generality:

> But the voices given to the synthesizer are only an echo of the voices of real instruments. In the process of electronically rendering a sound, that sound must be torn away from its local situatedness. Something is lost in the process of making the sound generalizable and transportable in the electronic media. The clue to what is lost is to be found in the naming of the voices on synthesizers. Sometimes a whole class of instruments is named, such as "brass," or "strings"; sometimes it gets more specific and refers to a particular instrument. But a "brass" sound is not the same as a trumpet anymore than a violin is the same as "strings." The emulated instrument is always stripped of its context – it is rather like an "ideal type" in social science. Even when it is a violin sound that is emulated it is not a particular violin such as a Stradivarius violin, or a particular Stradivarius instrument. All the little resonances and peculiarities that make an instrument unique are effaced. (Pinch 2006: 63)

Over time, the meaning of the sounds in a synthesizer stabilise and they become musico-cultural constants. The synthesizer gains its voice and a character that is associated to its brand name and model. Although Pinch notes that the Moog sound can also be emulated, he does not go deep into the characteristic of the Moog itself. The Moog is a highly characteristic instrument with strong constraints that colour its expressive potential. It is a situated instrument with an abundance of physical properties, such as those of the oscillators and filters. It is more characteristic as a singular unity, an instrument, than the generality we find in modern audio programming languages. It could be roughly stated that the evolution towards generality and expressivity is an evolution from uniqueness, situatedness, and character. We *can* build these into a digital instrument (as attempted in the ixiQuarks) but such constraints are always designed constraints, not those that come "for free" or derived by necessity from the physical materials used [see chapter 5].[170]

In fact, elsewhere, Pinch and Trocco talk about the MiniMoog as an *instrument* in opposition to a *machine* precisely because the synthesizer would not always "work" as it was supposed to. "It is the departure from theoretical models of instruments – the unexpected resonances and the like – that make an instrument particularly valued" (Pinch & Trocco 2002: 223). They detect two general trends with regards to the analogue synthesizer:

> For some it is an archetypical machine, a way to abstract and analyze the core constituents of sound, a way to render into algorithmical form, and manipulate and perfectly repeat with machine-like precision, the essence of sounds and music. For others it is a musical instrument with all the idiosyncracies and inaccuracies associated with the best acoustic instruments - an object to love, to learn to work with, to appreciate for its ineffable qualities and its own personality. (Pinch & Trocco 2002: 308)

We can assert that the former view has now moved over to digital musical instruments and most musicians talk about the analogue synthesizer with a romantic glance in their eyes and mention its thick, warm and characteristic sound. However, as Goodwin (1988) notes, in the early 1970s this very same sound was considered "cold" and "inhuman," and it was only with the advent of digital synthesizers did this sound transform into one defined as "warm" and "characteristic." We can therefore detect a trend here, finely expressed by Pinch and Trocco:

> Whenever a new mechanical contrivance enters the field of music, it triggers the same set of concerns: is it a proper part of the musical domain, an instrument that can release musical talent, creativity, and art, or is it simply a mechanical device, a mere machine? This debate famously occurred when the piano forte replaced the harpsichord at the start of the eighteenth century. Much less well known is the similar debate that arouse when mechanical levers were added to flutes in the mid-nineteenth century.... The boundary between musical instrument is constantly being redrawn with each new encroachment. (Pinch & Trocco 2002: 306)

In his keynote speech at ICMC 2008 in Belfast, the experimental musician Trevor Wishart[171] bluntly (and provocatively, considering the audience) declared that algorithmic music was "not music." He was referring to music where the machine, following the instructions from the composer (yet escaping his or her definite decisions), removes the interpretative role of the performer. Some people lament the ever increasing human disembodiment in the musical performance. For example, the piano removes certain "closeness" with the string: it cannot be

[170] It should be noted here that the digital system, at the lowest level of machine instructions, *can* show its materiality of processor type. A compiled code might behave differently on dissimilar processors due to the way they interpret NaNs (not a number), endianness, bit depth, availability of vector operations, etc. Sofware developers would usually seek to minimise divergence in behaviour between platforms, and might be inclined to see any remaining differences more as bugs than controllable creative opportunities.
[171] Wishart's 1996 book, *On Sonic Arts* has been extremely important in this field.

picked with subtle details, scratched or muted with the palm of the hand. The increasing machinery between the performer and the sound "engine" increases generalisations and minimises unique control. In the case of the piano it is a physical and natural machinery. With the electronic instruments it is a machinery of electricity, transistors, capacitors and resistors. With the computer it becomes a machinery of symbolic systems. One can model the timbre of the sound, the instrumental properties, the compositional system, and an artificial intelligent system to drive the whole "piece." For Wishart this is removed from "music," here defined as something humans do. This raises questions regarding the different emotional relationships people have with the various abstractions of existent musical technologies. What is clear is that the creativity behind the sound, the instrument, the system and the AI are of course an exemplary testimony of human imagination and inspiration.

7.3.4 Different Minds Working with Transmodal Hybrids

A focus on emotional relationships with musical instruments also applies to the different audio programming languages. Modern musical systems on the computer tend to be relatively complex and can be represented as simulation interfaces, graphical programming interfaces, textual programming languages and command line tools. The gateways into computer music systems make use of various types of input (iconic, graphical, code, or textual). For the digital musician (Hugill 2008), the impressive flora of tools and methodologies to do musical research, composition or performance can overwhelm at the same time as inspire.

Barry Eaglestone's team has done extensive work (Eaglestone et al. 2005, 2007b) investigating people's usage of digital media differs according to personality traits. Through various survey and interviews they have explored how cognitive styles affect people's choice of software and software types (whether high level simulation software, graphical patching environments or textual programming environments). Eaglestone's study focussed on a set of four dimensions of cognitive style: global/analytic, imager/verbaliser, intuitive/sensing, active/reflector.[172] *Global* individuals are prone to look at the general picture and work non-linearly whilst the *analytic* individual typically works in a linear and sequential manner. *Imagers* tend to think visually and like working with diagrams, pictures, charts, etc. whereas *verbalisers* tend to think more in words. *Intuitive* individuals are experimental and innovative whereas *sensual* people like to work with concrete information, real-world data and established methods. *Active* individuals are quick to try things out, often in social context, whereas the *reflective* individual normally prefers to work alone and think things through before acting (Eaglestone 2007b: 467).

[172] They derived these character traits from various sources, from both psychology and HCI. See paper.

It is obvious from the above that the ixiQuarks fit people in the former element of these dichotomies. From the surveys and interviews we have conducted there seems to be a high percentage of global individuals that prefer working with graphical data. It is harder to know where people are on the intuitive/sensing, active/reflector continuums as we did not design our questions to explore these dimensions. The ixiQuarks environment encourages the use of multiple windows, many instruments open at the same time (where automated processes are performing together with the musician), and the active user input is done through exclusively graphical and symbolic (metaphoric and metonymic) representations. This can be supported by Eaglestone's et al. findings that verbalisers are satisfied with available composition software, whereas imagers are "poorly served" by it (Eaglestone et al. 2007b: 471). This relates to and partly explains many of the enthusiastic replies we had in the ixiQuarks survey: the ixiQuarks fit the imager's cognitive style.[173]

The work represented by Eaglestone's team can be used as a supplement to the surveys presented in this thesis [sections 5.2, 6.7 and 7.5] as the current research has not been concerned with the psychological factors that determine people's choice of digital musical systems. Furthermore, what they suggest is that, in the design of our musical systems, we should pay more attention to these cognitive styles and provide different interfaces for different people. This does not mean a different interface "skin" but an entirely different interaction modality. It is questionable therefore whether this can be achieved in the same software or whether, indeed, we should talk about a *different* software.

7.4 Code as Expression: Systems to Think With

A programming language that doesn't change the way you think is not worth learning (Alan Perlis quoted in Wang (2008))

This section explores the situation of artists using open source programming environments in their work. It focuses on the layers of potential expression and investigates if an increase in the environment's expressivity decreases the possibility of actually making art. This question is studied through a survey done with artist-technologists, specifically exploring the time/creativity question. Which criteria lie behind the question of expressive strata in production environments is looked at; i.e., the question of finding a platform that enables musical creativity with as few limitations as possible, yet effective (the time factor). Finally, the

[173] We have discovered this through our workshops as well. Normally our workshops have equal distribution between visual artists and musicians, and the musicians are mostly performers as opposed to composers. Initially we were surprised by how many visual artists were attracted to these musical workshops, but after a while we understood what they were after, namely a visual representation of sound or the control of sound through graphical or embodied means. (Art academy students do not normally limit themselves by any medium. Sound is equally interesting to clay or oil paint).

status of open source, artist-created software environments as meta-art is discussed, i.e., expressive platforms that necessarily embody aesthetic ideas and purpose. As such, no environment is aesthetically or politically neutral.

7.4.1 Introduction

Let us rewind to the late 1990s. In the West the internet had become ubiquitous and artists were drawn to the new potentials of artistic expression through the medium of the computer, allowing for interactivity, collaboration and connectivity across time and space. This resulted in the emergence of various fields such as net art (where the properties of the networked medium are used as artistic material), software art (where software is written or commented on from cultural and political standpoints) and robotic art (where computers are used to control hardware, often through telepresence). It also injected new life into established practices such as interactive installations, video art and sound art. A prominent problem was that of tools. There were not many coding environments that allowed for sketching or fast generation of ideas, or which produced complex works that both looked good and worked well. The artist could either learn languages like C/C++ or Java and try to code their own engines for artistic creation or decide to use higher level environments such as Macromedia Director or Flash.

Fast forward to the middle of next decade, and we are now faced with a panoply of free open source tools and programming environments that have been created by artist-technologists for other artists-technologists to use. The explicit aims of these environments are to function as open toolboxes that do not limit or direct the artist regarding form or content (Puckette 2002; McCartney 2002). From the level of analogue input into the machines (Arduino, Wiring) to operating systems (pure:dyne, Planet CCRMA), to programming languages and frameworks (SuperCollider, Pd, ChucK, Processing, OpenFrameworks) to distribution media formats (flac, ogg), we are seeing a proliferation of environments made by artist-technologists to create and distribute their work. The aim is not only to create free and open source (thus collaborative and sharing in nature) production tools, but more importantly to allow for tools that do not necessarily limit the artist's expression in rigid ways.

This is a true paradigm shift and a very positive one for people using the computer for creative work, whether it is music, graphics, video, multimedia, installations or robotics. However, there are a few drawbacks, typically those related to the effort required to learn the environment. Working in open source programming languages can be time consuming and lead to side-tracks in the creative process where the production of the artwork sometimes becomes tangled up in working on the environment in which the artwork is made. In the work process, the distinction can blur between the work and the tool it is made with. This is not necessarily a negative. The problem is that of time-management and how artists manage to situate themselves at a level of code that benefits their work. In this section we will look at the reasons why people find working with open source programming environments for artistic production a meaningful and rewarding activity, despite its perils with regards to time.

7.4.2 Why Open Source Programming Environments?

Above I described the situation in the mid 1990s, where there were few alternatives for art practitioners to create their works in, unless they were creating them in quite time consuming and unfriendly programming languages like C/C++ or Java. The computer had become fast and cheap enough for use in interactive installations, algorithmic composition, realtime synthesis of audio and video, and powerful digital signal processing for manipulation of sound and graphics. As artists are normally inquiring people who want to know why someone designed the tool to perform certain ideas and not others, there was a need for environments where artists could explore the cogs and wheels of programming as artistic material which cannot be done if the source is hidden. This need is expressed by Jaromil the creator of dyne:bolic, a Linux distribution that includes all the main tools the media artist might need:

> **dyne:bolic** is shaped on the *needs* of media activists, artists and creatives as a practical tool for **multimedia production**: you can manipulate and broadcast both **sound and video** with tools to **record**, **edit**, **encode** and **stream**, having automatically recognized most device and peripherals: audio, video, TV, network cards, firewire, usb and more; all using only free software! (Jaromil 2007, my italics)

From the creators of pure:dyne, also a Linux distribution with a different focus, we read:

> The development of pure:dyne can be traced back to the inclusion of Pure Data in the dyne:bolic liveCD distribution. As this addition became increasingly popular, there was suddenly a *demand* to increase its support for Pure Data in a more serious production context… Today pure:dyne gathers a growing user community and has been used in numerous workshops and performances. (Mansoux, Galanopoulos & Lee 2007, my italics)

And the maintainer of Planet CCRMA (a Linux distribution including all the main tools for sound and video production) describes how users outside the Stanford University CCRMA centre began to use the Planet CCRMA as the link to the system disseminated and search engines began pointing people to its existence:

> This changed the nature of the project. As more people outside of CCRMA started using the packages I started to get *requests* for packaging music software that I would not have thought of installing at CCRMA. The number of packages started to grow and this growth benefited both CCRMAlites and external Planet CCRMA users alike. (Lopez-Lezcano 2005, my italics)

All three operating systems – dyne:bolic, pure:dyne and Planet CCRMA – are responding to certain needs and demands in the general culture. All three have become immensely popular and are used all over the world. The situation has changed dramatically: whereas a decade ago an artist working with computers would probably work on a proprietary operating system and with expensive software, users now use increasingly cheap hardware, free operating systems, and many of the best creative environments are open source and free. The impact is huge and more so for cultures whose currency exchange rate is not favourable to the Western valuta, rendering the chance to pay legally for software all but impossible.

Many of the people working in the field of music and the arts today started using computers when they became available in the right price range and interesting enough for creative purposes in the middle of the 1990s.[174] They either had to learn the hard way, with low level languages that required too much time operating on a level that was more computer science than art, or they would be stuck in the constraints of high level packages that limited their creativity. People began collaborating on projects, distribute code and often this resulted in environments that allowed others to jump the step of computer science and go straight into creativity. Here is how the creators of the Java-based programming language Processing construe the raison d'etre of the software:

> Processing relates software concepts to principles of visual form, motion, and interaction. It integrates a programming language, development environment, and teaching methodology into a unified system. Processing was created to teach fundamentals of computer programming within a visual context, to serve as a software sketchbook, and to be used as a production tool. Students, artists, design professionals, and researchers use it for learning, prototyping, and production. (Casey & Fry 2007).

The FLOSS (Free Libre Open Source Software) movement in the arts has become strong and well functional (Mansoux & de Valk 2008). Free and open source software now exists for almost everything one would like to do, and in many cases it is impossible to find equivalent tools developed in the commercial sector. However, learning a programming language is not an easy task. One can become conversational in a month or two, but a fluency of expression is something that might take over 2-3 years, with many hours put into the practise every day. Furthermore, the logic of programming is not the only thing that has to be mastered: working at this level means that the artist has to understand the nature of sound and visuals from both physical and psychological perspectives. (In the case of music one would learn the physics of sound, digital signal processing, synthesis theory, and psychoacoustics). This takes roughly as much time as mastering an acoustic instrument and has to be justified somehow as a meaningful activity.

[174] The precise timing here is obviously debatable: some might say with the Atari computer and the invention of MIDI but others would argue that it was when one could perform real-time manipulation of audio and video in high resolution.

7.4.3 Time and Expressivity in Artistic Progamming Environments

In order to understand the question of time management and expressivity, i.e., the question of making the tool itself versus making something *using* the tool, developers, programmers and artists were asked about their work patterns and attitudes towards time. How do creative people justify their spending time on building tools and not the artwork itself? Why the fascination with code? How can this technology represent our thinking? From the replies to the survey, we can detect a few threads originating in the idea of code as creative material. They are here presented as thirteen subsections for the sake of clarity.

The participants in this survey were: Marije Baalman, Ross Bencina, Tom Betts, Andrew Brown, Graham Coleman, Alberto De Campo, John Eacott, Tom Hall, Enrike Hurtado-Mendieta, Aymeric Mansoux, Click Nilson, Julian Rohrhuber, Andrew Sorensen, Matthew Yee-King and IOhannes m zmoelnig, and a few that chose to be anonymous. All are active users (both developers and performers) of audio programming languages such as SuperCollider, Pd, Impromptu and ChucK.

7.4.3.1 Coding as Self-Understanding

What I cannot create I do not understand - Richard Feynman

One of the participants in the survey quotes Feynman as here above. This quote expresses a stance that most of the participants seemed to subscribe to. The desire to learn about one's music or visual art through building the tools is a common characteristic of all the practitioners. A crucial distinction is whether one is building the tool for a personal or public use (the question of specificity or generality asked about in the ixiQuarks survey). The epistemic dimension of digital musical instruments implies that patterns, forms and structures that should arguably be defined more as of compositional nature can be inherent in the instrument. It hardly makes sense for the "digital luthier" that inscribes compositional elements into her instrument to distribute the software as a general application. However, some composers distribute software as musical pieces that can be performed, with or without human performance or intervention.[175] There are many factors that define the "politics of open works," such as artists wanting to protect their work, the belief that eccentric software is not relevant to others and simply the lack of time required to support a software release. If we compare personal software and released software, the latter seems to be time-consuming on an exponential scale.

> Tools have such a strong impact on the artistic process that I like to know mine well and there is no better way to know a tool than to build your own. Paul

[175] The work of Joshua Parmenter is a good example: http://www2.realizedsound.net:8080/josh/Music/Music.html

> Lansky once said that he didn't really distinguish between building the tools and making the artwork. I have a lot of sympathy for this idea. [as][176]
>
> Where this question becomes tricky for me is in relation to developing tools for other users. On the surface this really is a black hole. You end up spending a lot of time supporting other users needs and this generally has little to do with your own artistic practice. However, end users are really good at forcing you to do the bug fixing, stability and performance improvements that you would probably not do on your own, but which you are really glad you did! [as]

7.4.3.2 No Distinction of Building a Tool and an Artwork

As the survey in section 6.2 showed, many musicians actually experience playing an acoustic instrument as a cliché-prone activity due to how practised and inscribed the instrument is in a specific cultural tradition. Building one's own tool is often seen as a more creative activity that forces the artist to a self-awareness of the compositional processes and ideas that lie behind the work; qualities that are useful for, and often cannot be separated from, the process of making art.

> In software art… some artists strongly refuse to dissociate the art from the tool, to the point where using the words "tool" or "environment" becomes completely irrelevant. [am]
>
> Programming for music is a part of music-making, just as ruling bar-lines on manuscript paper can also be a modest part of music-making, all distinctions less important to my mind than whether or not one is concentrated when working. [th]
>
> I rarely use a piece of software I've written for more than one thing, so use and development are the same in most cases. I'm not really even that bothered about the "music" as I don't really see it as the primary goal or validating factor. [tb]

7.4.3.3 The Artwork as a Process

Software art fits well with the modernist tradition of seeing art as a *process* rather than a finished piece, as software is not written in stone. Software is never finalised. Only the material constraints of older art forms reinforce our habit to see art objects as completed works of art. These are constraints that were introduced mainly with the Gutenberg press and the phonograph (Attali 1985; Kittler 1999). Before these media-technologies, texts and music would normally change every time someone wrote another copy of the text or played the song. Code as immaterial, abundant, copy-able, omnifunctional and executable text is never dead. Instead of finished pieces of art, frozen in time, we have versions and updates.

---

[176] Paul Lansky participates in the survey with audio programming language designers in the next section.

> If the art is truly living in the process, then developing it will not only change the guts but also influence the output as they are the same and sole object. [am]

7.4.3.4 Coding as a Conceptual Practice

Art can be highly formal, mathematical or scientific. Music is a good example of an art form where practitioners have investigated formal and mathematical relationships in harmony, melody, timbre and meter. Programming as a means of formalising one's thought, externalising them (Dix 2008), and testing them in performance using a logical machine is therefore a tempting method of working for many artists.

> Programming problems can be an interesting thing to pursue while musical problems (as defined by Schoenberg and others) are being turned over in one's mind. Especially so as programming problems thus encountered are usually simpler than the accompanying musical problems. [th]

> For me art is not so much about expression, but more about reasoning, so the process is maybe a different one. The environment is not separate from the work, neither as a work of art nor as a work of theoretical research. [jr]

7.4.3.5 Creating the Tool for Originality

The modernist demand for originality in art is still prominent in contemporary culture, although its manifestations have changed. Thinkers in the field of creativity studies consider the highest form of creativity to be creativity where the creator (be it artist or scientist) transforms the cultural space in which he or she works (Boden 1990). For many, it is therefore by artistic and practical necessity that they build their own tools, as using other people's tools might lead to less thought and original design solutions to both aesthetic, formal and technological problems.

> Inevitably tech issues intrude. But I guess you want to come up against difficulties; else you're not pushing the boundaries? [nc]

> Knowing that I'm using an obscure programming language that very few are using makes me excited, as the language allows me to think and compose in certain innovative ways. [an]

7.4.3.6 Coding as an Artistic Practice

Many of the participants in the survey define coding as an artistic practice in itself. In fact, some of them see it as a performative act and do practice livecoding in front of audience with projected computer screens (McLean 2004; Nilson 2007). Coding is here viewed as a way of externalising thoughts, in a manner similar to sketching by drawing or model building. But furthermore, the audience is invited into the logic of the composition, into the creative process taking place in the creator's mind. This is relevant in livecoding where the performance is typically a process of composition and where the machine performs instructions by the coder/composer/performer. Naturally, traditional terms do not work in describing this practice. When programming is seen as a performative action, a choreography of thought, it ceases to be a means to an end and becomes an end in itself.

> I see programming as a part of my creative practice, just like playing the piano or studying books on composition. As such I try to be as fluent at it as I can afford to be so that my creative ideas can be realised as efficiently as possible. This is not at all to say that programming is transparent to the creative process - like any tool it has an impact on the work. [ab]

7.4.3.7 Coding as Craft: An Inscribed Skill

Any search engine will show countless books and articles on the topic of coding as craft; a skill that has to be mastered through time with lots of practice. The programmer learns to think in the language that he or she works in, formulating the given problems in the terms of that language, often conceptualising the world through the means of programmatic paradigms. But coding is not only a craft: it is a performance as well, as the case of livecoding exemplifies. Livecoding requires excellent skills by practitioners, which involves fast thinking, fast typing, practiced algorithms and good knowledge of the creative instrument (the programming language) they are working with (Nilson 2007; Sorensen & Brown 2007). Considering N. Katharine Hayles' distinction between incorporation and inscription [section 3.5.1], we might conclude that livecoding as a process operates more prominently in the inscription domain as opposed to the incorporated domain that we find in the performance of acoustic instruments. The dichotomy, of course, is not absolute, as Hayles herself explains: "incorporating practices are in constant interplay with inscriptions that abstract the practices into signs" (Hayles 1999 : 199).

> I find my expressiveness is limited by the "gestures" my environment can support. I prefer to be able to express things in my library very tersely, so I see working and thinking about changes to the performance environment as relevant part of my time spent. Much as an instrumental performer must spend time learning new techniques, the livecoder must take time to extend their library of musical gestures. [gc]

> A virtuoso computer musician needs to be a decent programmer.... [ab]

7.3.3.8 Creativity Rather than Art

The concept of art is famously narrow and defined by its various cultural/political contexts. Most of the participants of the survey are not concerned with "art" as an isolated cultural phenomenon and see creativity as a ubiquitous human behavior. For some there is not only the absence of distinction between the tool and the work of art, but creating the tool itself is considered more important… and enjoyable:

> Building sequencers rather than sequences seemed to be more fun. [iz]

> For me, the software itself is what I'm focusing on and what I like to present as my work. I sometimes perform with it and have released some recordings of the tool in practice but those almost have the status of a "demo". I find that some people don't understand this attitude, but that's not really my problem. [an]

However, some want to draw a distinction between making the tool and the work that could be made with the tool (as opposed to section 2 above). In fact this seems to be related to the level of coding in which the artist is working. For example in Pd one could write an external (an object encapsulating more complex and lower level code) or in SuperCollider one would write a Unit Generator. This is a stratum down from patching in Pd or coding in sc-lang (the native programming language of SuperCollider) as it involves more debugging and, most annoyingly, having to compile the code before applying it.

> I think it is what it is. If you develop a UGen, you develop a UGen; you shouldn't kid yourself that you're making music, though you might be making an essential technological component. But things blur again, for example, in testing the UGen, which is suddenly so very exciting, musically… [an]

7.4.3.9 The Danger of Spending Time on the System not the Art

Most participants acknowledge, in one way or another, the danger of forgetting oneself in building the ultimate creative system and never having time to use it for creative work. The mindset of the programmer-artist is often *how* music will performed, rather than *what* music will be performed, i.e., the focus can be directed overly on the system in which the performer is able to express herself rather than the musical expression itself.

> Creating your own dream environment to create your art is a dangerous game as most of the time the focus is only on the system, developing it, updating it, add features, document it. On a similar way, the environment can become so important that the art produced with it becomes minor and not important. [am]

> Sometimes I rehearse every week, but this is perhaps for 3 hours, vs. 30 hours working on the environment. [rb]

This distinction, however, is rather superficial as there is as much creative thought put into the design of an environment as the design of a musical piece for example. One of the assertions of this thesis is that composition, design, and performance blur in the digital instrument. The instrument becomes an epistemic tool, with which we engage in a virtual embodiment. Mastery of such instrument is therefore not embodied, but one of system-understanding and epistemic actions. Furthermore, the building of creative tools is increasingly seen as an artistic endeavor, as proved with Linus Torvalds being awarded the first prize in Ars Electronica 2003 for his Linux operating system.

> It is not easy to give an artistic statement about an operating system, because while an operating system can be a work of art (I certainly feel that there is an artistic component to programming), it's not in itself very artful… In more "artistic" terms, you might consider the operating system to be the collection of pigments and colors used to create a painting: they are not the painting itself, but they are obviously a rather important ingredient - and a lot of the great painters spent a large portion of their time on making the paint, often by hand, in order to get their paintings to look just right. (Torvalds 2003)

7.4.3.10 Coding Takes Time

Most of the participants seemed to have a stoic attitude towards the time they put into their work. They see this as the nature of art itself; in order to make good art, you have to work a lot whether that is through practising, reading, preparing materials, learning, going to concerts/events, or talking to other people. Learning the art of programming is no different than learning how to play an acoustic instrument or train the hand in free-hand drawing.

> Well I am still using up the mileage to be gained by convincing yourself that all the time you spend screwing around with open source software trying to get it to work makes you learn things you wouldn't have learnt otherwise. [myk]

> When preparing for a piece, there is always a tradeoff between what is possible within the environment, what do I need to add, and what am I able to implement (and debug!) before show-time. [mb]

> In a way, the 24h to the day limit seems like an inconvenient constraint; but the again it is good preparation for the limited lifetime constraint we all face at some point, and trying to spend that time well. [adc]

7.4.3.11 Programming as Meta-Art

Creating something that creates something else is working at the meta-domain. Many software artists, such as Adrian Ward,[177] see themselves as creating meta-art, i.e., art that is used by artists to create art. It varies how the programmers see their role, from being participant in the creation of the end object (like Ward claims) to rejecting all co-authorship and clearly separating the software as meta-art and the product that it creates when an artist uses it.

---

177 http://www.signwave.co.uk

> As I see it, if I create a tool that is used by artists in their work, I have created something that is of value to them as a product. I see this as an object of art and the artists using my tool not only as users but also as aesthetic "actors" or "perceptors" for lack of better words. (Basically for what "listener" is to music, "viewer" to film, we need a word for the person that engages with software art). [an]

7.4.3.12 What I Cannot Take Apart, I Don't Own

Much commercial music software sets up the whole structure of the music by default and the process of composing is more a question of removing presets than composing from an empty plate. Many people find that what they cannot take apart and understand the bits of, they cannot claim is their own creation. This idea can be found in circles of people working with hardware just as software.[178] But the question is not only that of understanding, but also of archiving as commercial formats are often closed. An artist that produces work in a closed source format or software might not be able to revisit that work in the future.

> I feel I need to understand the technological foundations of my music, what cogs and wheels I am using to implement my ideas and how they in turn change through using the tool. Therefore open source software is important to me, if not I'd be stuck in a situation where I feel I am a mere consumer of the software and not an active participant. Of course, this has to be taken with a grain of salt, as there is obviously code "all the way down" and I'm not interested in looking at machine code or processor design. [an]

7.4.3.13 Oh, Where has the Body Gone?

The problem of embodied skill, the trained musical/artistic body, is something that is ingrained in the question of coding as artistic practice. This feature of the man-machine relationship comes up repeatedly when talking about the digital arts. For livecoders, the haptic interface of the computer (such as mouse and keyboard) might just be an unnecessary interruption between the thought and the implemented algorithm, but for other artists, the body is an integral part of the creative process. The pre-linguistic or unconscious mind might be better expressed through bodily movements than as linguistic thoughts after all.

> Also sometimes I find it quite cumbersome to have to program everything, rather than being able to just play, like I can on a piano or an analog synthesizer; i.e. instruments that have behaviour in themselves. [mb]

The comment above shows that its not only the artist's body that becomes secondary in livecoding or creative programming, but we also find lacking the body of our tools. The paint

[178] http://makezine.com/04/ownyourown/

and the brush or the acoustic instrument contains properties that are given by their materiality [see discussion in section 4.4].

7.4.4 The Question of Strata and Location

Programming is a craft and an artform. It is an inscribed skill that involves complex relationships of various factors: an environment and its logic expressed in a programming language; the programmer and his/her integration with the environment's logic; the programmer and his/her embodied relationship to the hardware used; and various other threads that connect the finer elements of the human-machine network. The question for the artists is *where* to locate themselves in the complex ecosystem of programming languages and libraries that can be taken into use. Where in the machine building process do they want to work?

Here it can be useful to think of software as strata: as concrete layers or sedimentations of varied abstractions. Assembler is an abstracted machine language; Java compiles into a virtual machine that is machine language; SuperCollider is a language written in C++ that is a level above assembler, etc. These are all strata that function on their independent levels, but allow for modifications across the spectrum (or belts of abstraction) if we are working in open source.

> The strata are extremely mobile. One stratum is always capable of serving as the *substratum* of another, or of colliding with another, independently of any evolutionary order. Above all, between two strata or between to stratic divisions, there are *interstratic* phenomena: transcodings and passages between milieus, intermixings. Rhythms pertain to these interstratic movements, which are also acts of stratification. Stratification is like the creation of the world from chaos, a continual, renewed creation. (Deleuze & Guattari 1987: 502)

We have already seen from the thirteen threads above [in section 7.4.3], how some people enjoy the building of a tool even more than performing or composing with the tool itself. The focus ranges from working on operating systems, the source of an expressive language (like creating the source code of SuperCollider, Pd or Processing) or working with code as expressive means and try to shy away from computer engineering tasks. As the quote of Deleuze and Guattari indicates, any one artist can traverse these layers of abstraction in different times; there are indeed transcodings and passages that can be taken between strata, and in some cases the environment encourages such transcodings, ixiQuarks being the nearest example. By the same token, each programming language has their own characteristics and for some it might be better to work in a high level language like Lingo or ActionScript whilst others click with Processing or SuperCollider. For yet others, thinking in a graphical programming language like Pd is easier and more intuitive, and switching to writing objects for it (at a lower level) is only done by necessity. An important question here is that of the economy of time vs. expression. Many people are ready to invest more time in the building of their expressive environments and get in return a personal and unique tool adjusted to their special ways of thinking and methods of production.

However, this is not the whole story. Choosing to work with Pd rather than Max/MSP or Processing rather than Flash or Director is also a political decision. It relates to the freedom of individual expression; who is in power; relations to the capitalistic market; the question of open formats for future archiving; and a clear awareness of what it means to be a user of software (Mansoux & de Valk 2008).

7.4.5 Conclusion - Creators vs. Consumers of Expressive Tools

The open source tools mentioned above are practically all commentaries on the question of software consumption. As they are open source, they invite the artist-technologist to modify and transform the tool by collaborating in its design either privately or as part of the culture around the tool. The idea of collaborative artist-made operating systems, programming languages/frameworks, and creative tools makes sense. They are the people that know and understand how and what is required in the process of making computational art. They are the people whose work involves the reinterpretation of the social ontology of our culture through re-categorisation. The era when artists were forced to buy closed source software and thus constantly being pushed into ergonomical patterns or conceptual boxes by the companies that made them is now over. The "software consumers" of the commercial world become "software co-designers" in the world of open source – users who have the freedom to choose the form of their expression and the manner they express it.

7.5 Creating Creative Software: A Questionnaire for Language Designers

7.5.1 Introduction

No programming environments originate in a cultural void. They are dependent upon the technocultural context in which the are created (Pope 1995). For a programming language designer like Larry Wall, the creation of a new language (Perl) involved deconstructing other languages and select the parts that he liked: "To the extent that Perl rules rather than sucks, it's because the various features of these languages ruled rather than sucked" (Wall, 1999). It is therefore pertinent to explore the origins (or ontogenesis – [see section 6.4]) of audio programming languages and the philosophy of their designers. A survey was conducted that involved audio programming language designers. Responses came from the main players in the field: James McCartney (SuperCollider), Miller S. Puckette (Max/MSP and Pure Data), Andrew Sorensen (Impromptu), Ge Wang (ChucK), Paul Lansky (CMix), Phil Burk (JSyn) Torsten Anders (Strasheela), Peter Stone (Symbolic Composer), Matt Wright (Open Sound Control) and Graham Wakefield (LuaAV). Obviously, not not everyone contacted was able to participate, which explains the absence of some important languages, although I believe these above are the main, cutting-edge tools of today's plethora of expressive coding environments in the arts.

The survey was divided into five principal areas: 1) the *rationale* behind embarking upon creating a new language, 2) the *aesthetic* stance of the designer, 3) the public/peer *response* to the language, 4) the *techno-social context* in which it exists, and 5) a post-hoc *evaluation* of the work. Below, the five areas will be explored, and an attempt made to extract interesting findings from the survey. Full answers to the survey can be read in Appendix I.

7.5.2 Rationale

There were three questions that founded this part of the survey:

- Why did you embark on the project of designing a new musical language?
- What was the design philosophy you had?
- How much do you feel was "given" to you from older environments, operating systems, hardware, code libraries? To phrase it crudely, did you feel that you were a node in a network implementing your vision or an inventor that had to invent everything?

The aim was to try to understand the reasons for starting the work on such a time consuming and complex task as creating a new programming language for music (which is a relatively complex engineering task due to the DSP involved). Any design derives from a specific perspective and an outlook on how things should be conceived, implemented and performed. Given the rationale of this work, it was relevant to ask about the philosophical background and the aims the designers had for their work. The question of originality, whether the desingers see themselves as part of a lineage or inventing something afresh is also explored in the context of both the actor-network theory and the theory of epiphylogenesis.

It might be banal to state this, but the fact is that the designers created their audio programming languages because they found there was a need for it. None of the existing tools allowed them to do what they wanted to do, so they set upon creating the tool themselves. It is important to note that all the respondents created the tool for their *own* use as creative musicians. In fact, it becomes apparent that the act of creating the tool is seen as an equally creative and enjoyable activity as creating pieces of art (of course we can easily define the tool itself as art). The designer has some idea of a process (such as McCartney wanting to write expressions that represented "parametrizable classes of sound processes," Sorensen focusing on livecoding, Burk aiming for "precisely controlled sound," or Anders wanting a system for "constraint-based music composition") and the goal becomes to create a language that affords these processes with as much ease as possible, i.e., allowing the user to operate on the level of musical composition rather than at the level of software engineering. Wakefield explicitly comments on this in relation to what he calls creative computation: "It was important to distinguish the problem space as creativity support rather than productivity support, for a particular orientation of composition that is experimental and engaging with the nature of computation, rather than simply using computers for a priori tasks."[179]

If one of the main pillars in the design philosophy that supports most of these environments is the goal of operating at the high-level of musical thinking, rather than engineering, the other might be the wish to support interactive coding. Nearly all the authors talk about the desire to do realtime manipulation of sound or compositional processes at a high level. In the case of ChucK and Impromptu, the goal becomes directly to support "live coding" or "on-the-fly programming," but SuperCollider and Pd support those activities as well, although historically livecoding was not a prominent performance style when those environment were initially conceived. In fact, it was largely *because* of them, particularly SuperCollider, that livecoding became possible.[180] This realtime focus is clearly derived from the experience of being a musician. Although composing is not a realtime activity, instrument playing is, and it seems that all the designers share this desire to transform the computer into a realtime instrument rather than being merely an off-line compositional pattern generator or score-editor. Rapid prototyping, conciseness and readability of code are all features that are necessary for the computer language to become "instrumental" or usable in realtime performance.

Having dealt with questions of man-machine relationships in terms of the actor-network theory earlier and also seen the evolution of technology as epiphylogenesis in the theories of Stiegler, an interesting question to put forth is how the designers see their inventions: as something part of a larger whole, or as something that they genuinely created from scratch. This question is also raised in relation to recent theories of creativity (Boden 1990). Interestingly most of the inventors see their work as strongly embedded and intertwined in a lineage of technological inventions. They define themselves rather as a "node in a network" than as inventors that create everything from scratch. However, the early inventors are more prone to think that they were creating their systems from the scratch. This distinction is interesting as it can be argued that they were just as much using the technological affordances available to them in their work; the biggest difference is perhaps that they were taking them from one domain and introducing them into another. The reluctance to cite external factors as instrumental in their creativity could also be explained with reference to the cultural acceptance of post-modern ideology, where the romantic idea of the creative and original genius is substituted with the idea of the creator as a filter of influences, emphasising the role of the reader as giving the work its meaning (See Barthes, Derrida, Foucault, etc.).

---

[179] Wakefield here makes the distinction related to the distinction of expressive vs. production software as described in section 7.3.2.

[180] Although one should not omit to mention the pioneering work of slub in this context. Slub worked with their own custom made software, some of it built in Perl (McLean 2004).

7.5.3 Aesthetics

The questions raised in this section were:

- Were your considerations (criteria for design) mainly personal or public user needs? Or both?
- Did you have in mind a certain musical style or styles when designing the language?
- Do you feel that your musical ideas are inscribed in the software structure? (can you (sometimes) hear your "signature" in the music produced by the software?)

As all the participants are practicing musicians whose primary goal with creating an audio programming language is to make a musical tool, it is not surprising that most of them were thinking of their own musical practice when developing it. Some also tried to think about potential users (Puckette, Burk and Wang) but the source for the design is typically based on the working practices, the system knowledge and aesthetic preferences of the designer himself.

However, it is clear that none of the systems were designed explicitly for a particular musical style. Both Puckette and McCartney have written about that in publications (McCartney 2002; Puckette 2002). The goal is stylistic neutrality and they all seem to agree upon that, but there is an awareness that this is never achievable. The technology is always inscribed with an agenda, no matter how much they tried to avoide it. McCartney points out that “with things like Patterns [a system for structuring events in time] it is easier to use them to make rigid sounding music vs. more fluid timing, which I think is unfortunate, but that is the nature of computers,” whilst Puckette shows the institutional politics of his context “IRCAM certainly was more open to some styles than others.” Wang is conscious about the inscriptions inherent in any language construct: “we tried to keep things general, while being mindful that the language can’t help but suggest/impose certain ways of doing things and indirectly the music made. For ChucK, we tried to take advantage of this latter point, rather fighting it.” We find the same in Anders’ Strasheela, which is primarily a note-level AI compositional environment. Anders points out that

> constraint programming is particularly well suited for dealing with musical aspects which are described from different viewpoints. This is the case for the pitch parameter, which is often described from different aspects such as harmony, melody or counterpoint. Consequently, Strasheela is particularly well suited to model music theories on pitches – in contrast to most algorithmic composition approaches. This may lead to certain musical styles.

Although, McCartney points to the Patterns in SuperCollider as one way of using the language, SuperCollider is also designed for low level audio synthesis, providing ingredients such as waveform generators, filters and FFT processing. And here the environments differ, as some are primarily focused on the synthesis level and others at the compositional function and yet others (such as SuperCollider) combine both. As an example, Sorensen’s Impromptu is more focused on the compositional level and offshores the synthesis level to Apple’s Audio

Units, whilst ChucK is strongly timed and can perform control events at a sampling rate timing.

With regards to the question of musical "signature" in music made using their tools, all the designers concurred that their own and personal musical ideas are hard to detect, if not impossible. However, they pointed out that often a particular unit generator could be detected (McCartney), or a certain sound that derives from the way a patch is designed (Puckette). It often happens though that users copy examples that come with the software and as such, musical ideas can be traced in the music of the system's users.[181]

7.5.4 Response

The questions in this section were:

- How do you feel people responded to your software?
- What is the best/worst criticism you've heard about your software?
- Which events do you feel have increased the public awareness of your creation? (media coverage, "famous" users, awards, user communities, etc.)
- Has the usage of the software surprised you in any ways that you didn't foresee? How?

All the designers showed some degree of surprise that their creation was picked up with as many people as proven. McCartney expresses this clearly "I figured that the world market for a new synthesis language was probably 40 people," but he has been proven wrong in that prediction. In general, the designers experienced the users to be highly appreciative of their creations, and the generosity of making it publicly available for others to use. The open source philosophy, which invites any user to become part of the development team of the language, is explicit about this sharing of work for the greater good. All open source work therefore benefits from a certain goodwill from users who do not criticise or demand the same things from them as they would from commercial software.

The typical criticism of these tools were that they were too complex, with a steep learning curve. Coming from musicians, who often have spent years practising one particular instrument, this criticism is interesting. It is as if there is a general belief that all instruments and tools that exist on the computer should be simple; easy to learn and intuitive. People often find the use of the tools to be limited to programmers. This assumption can be questioned. Furthermore, the strong culture of open source can be critical of other work methods as explained by Sorensen (whose software is not open source): "I think one of the most severe and ongoing criticisms of my software, is why it's not open source :) Oh and of course the fact that it's Lisp ;) Oh yeah and also being OSX only seems to be offensive to some people ;)."

The designers were not strongly aware of the PR side of their software; word of mouth, journal publications, someone performing with the software, etc. Wang presented a list of

---

[181] This is a fact that we have experienced with ixi software many times. We can hear the patterns of the instrument, sometimes we have even heard the demo samples distributed with the software used in people's music.

events that might have made ChucK visible. None mentioned events that had much PR value, such as the reacTable team experienced when Björk started using the technology.[182] This is relevant to another dynamic: that of the creative artist. Artists do not tend to talk about the tools influencing them. Although it is obvious that Max and SuperCollider have had strong influence on many musicians of all genres and styles, it is seldom part of artists' world view to see the tool as an influencing factor. It is all in one: a given fact (thus ignored), something neglected (it is an unromantic study of the artistic process), and even a taboo (as it destroys the myth of the artistic genius).

The question whether the designers were surprised by the way their tools have been used had converging answers. Although there is the knowledge that when you build a general tool it can be used in any possible way, there is always usage that genuinely surprises and intrigues the designer. People misusing the software, plugging it into use with other technologies, or simply creating things that are quite idiosyncratic and unique. Burk explains this factor well:

> I did not foresee Larry Polansky using goldfish and video cameras to control vocal processors. I did not foresee Nick Didkovsky constructing a network based lottery where people won the rights to control the spectrum of a sound. Nor did I foresee Matthew Yee-King constructing a web based farm where users bred complex FM instruments using genetic algorithms.

7.5.5 Techno-Social Context

Any human-made artefact, especially high technology, is invariably bound and inscribed in a complex web of intricate techno-social structures. There are traditions of various kinds, institutions, cultural conventions, protocols, legal issues, etc. that all play the role of actors in this equation. It was therefore of interest to know how the designers conceived of the context in which they work. The questions here were:

- Have you had to deal with larger institutions (copyright issues, patents, media, software/hardware companies, academia) in your work with your creation? Has it been positive or negative? How?
- Eventually, did the user-base constrain or ease the development of the language?
- If the software is open source, has that been beneficial? Any negatives? If not OS, why?
- What are the biggest constraining factors? (operating systems, processor speed, users, money, time, etc)

Equipment, time, knowledge, money and other factors are important when developing systems as complex as audio programming languages. Particularly in the early days when access to a computer was costly and scarce. This is a factor that has coloured most work in the field. Puckette talks about the struggles of IRCAM's copyright over Max, which forced him to start developing a new environment, Pd. McCartney is under a contract with Apple that states that

[182] The sound engine in the reacTable is made in Pd, for example.

the company owns anything he creates during employment with them. These situations, however, are atypical in the modern context. Computers are cheap and the open source culture allows people to collaborate and co-develop systems of high complexity and where systems can evolve quickly and effectively, even if many of the contributors are only dedicating a small portion of their working week to the development of the tool.

The decision to release the tool to the public is not an obvious one. With it comes the users' demand for generality and the exorcism of all personal idiosyncracy. The tool could have remained an instrument for personal expression, but is now transformed into a tool that has to support other people's demand. The question of what the user-base means is therefore relevant. All the designers agree that the experience of releasing the software to the public has been positive. People develop the things they need themselves, answer each other's questions on the mailing lists, and, as Wang phrases it "provide feedback and direction for the language development/research." Open sourcing the code has been a positive experience (only Impromptu and JSyn are closed source) for most of the participants. It means collaboration, distribution of workload, and covering of areas that the individual designer might not have interest in, but the general user community might. The negative of open source is expressed by McCartney that the project might lose its "core simplicity," and for this reason amongst others Sorensen has yet no intentions for opening up the source of Impromptu. A common reason for releasing code under open source license is that the developers work for academic institutions or have received grants from public bodies and they therefore feel obliged to give the work back to the community.[183]

The answers to the question of constraining factors were as different as the respondents; learning curves, difficulties in improvisation, lack of time, technical constraints, etc.

7.5.6 Evaluation

Creating a tool and releasing it into the public domain means that the tool starts to take on a life on its own. It is of interest to ask how the designers evaluate their system, how they see the future of it, and, in more general scope, exploring ideas of how they retrospectively see the act of writing music as code.

- What are the problems with your software?
- How do you see the future of your software?

[183] This is a good reason for public bodies (such as universities, academies, art institutions) to make explicit policies of using and supporting open source software whenever possible. Instead of investing thousands of pounds in commercial software, university labs would often better spend the money in supporting individual developers of open source projects to develop particular features that they might need. The argument is that these institutions are receiving taxpayers money and would best spend them on projects that directly could benefit all taxpayers. There are already various universities that support this model.

- Finally, as a programmer and musician, do you have any comments on writing code as musical practice? Is embodiment important in musical performance?

The questions of problems with the software was typically answered with technical details specific to the particular environment and therefore not very relevant to explore here in any detail. The questionnaire has been focused on technology, and probably framed the participants in an "engineering mode." It would be interesting to pose a purely musical set of questions to the same people and ask the same question about the problems with their software.

When asked about the future of their software, the authors of the two biggest environments, Pd and SuperCollider, stated that they cannot change much from here on. When an environment has become open source, with many developers, it becomes harder to make any drastic changes. McCartney talks about creating a new musical language, whilst Puckette admits that Pd cannot change substantially, nor can Max, as it has reached a stage of stasis. The authors of the younger software (thus less advanced and complete) are naturally more open for changes and evolution of their tool. Also, in a world that has a variety of tools, it becomes more natural for people like Sorensen, to focus on his own tool and develop it primarily for his personal musical expression, although he talked about how helpful the user-community is and how it forces him to solve problems that he otherwise might ignore.

The question of writing code as musical practice had very diverse replies. McCartney expresses his interest in "composing music by composing functions," but he stresses that this does not have to be in a realtime performance. In fact, he prefers "tape music" over livecoding practices as he finds it more interesting to listen to the music ("meditate, introspect, look around, do whatever") than watching people program. Puckette emphasises the distinction in designing an instrument and composing. For him, these are two very different activities and the same person can hardly master the two fields properly. (A controversial statement, for sure). However, "there may be a sense in which software embodies musical practice, but if so, it's extremely indirect." For Wang livecoding is a musical practice, a mode of musical performance. Code, and different programming languages, can fundamentally change the way we make music, and therefore perceive music as well (for example when watching/hearing a livecoding performance). Lansky sees the possibility of coding music or instruments "as the modern day analogue of having an instrument shop where I could design, build, and play personal and idiosyncratic instruments." Finally, Anders philosophises about whether composing with code is less "emotionally charged" than composing with pen and paper and finds analogy in how the musical instrument is more abstract and detached than the human voice.

7.5.7 Conclusion

The survey with these influential audio programming language designers corroborates many of the topics addressed in this dissertation. It shows how individuals are able to reject the script of technology and react with their own production. It demonstrates how an actor is able to employ the necessary factors (his or her networks) and produce a novel technological artefact that encapsulates problems (here of musical composition and performance) in novel terms. It also shows how the actor is necessarily drawing from existing technologies – most of the designers saw themselves as nodes in a more complex network than creative geniuses – in his work, being a transformer of systems (taking knowledge from one domain and applying it in another). They are, as Gille, Stiegler and Mackenzie would phrase it, an element in the process of technological transduction [see section 2.5.4]. Furthermore, we see how the desire to create a new tool came from the designers' deficiency in available systems. They needed to create themselves the tool they wanted to use, typically transferring, or bringing with them, knowledge from other domains (computer science, signal processing, algorithms, human activities) into the domain of music making. In other words, their particular background in technology (which computers, languages, systems, models) and compositional technique has primed the designer to think in specific ways, thus inspiring the designs for new languages.

All the designers claim that they aim for aesthetic neutrality, i.e., that all music creation should be possible using their system. However, most of them acknowledge the fact that, in reality, such neutrality is never achievable as each work paradigm colours the methodology and the musical outcome. This is inevitable, but we should recognise that even if every environment constrains to some degree, these constraints are infinitesimal compared to higher level software systems, such as the GUI-based sequencers and virtual instruments (ixiQuarks included). I have dealt with the idea of constraints in creative tools in various sections of this thesis, and [section 7.4.4] further with the problem of where to locate oneself in the spectrum of high and low level languages for artistic creation.

Creating a tool for creative expression and then releasing it on the net for others to use is obviously an honourable act. People are therefore appreciative and substantive in their criticism. There are topics that are up for debate, such as the decision to sell the software or not opening the source, but such criticism rarely takes a meaningful turn, as people are always free to choose other environments. To a considerable degree, users therefore subscribe to the vision of the designer, but they are able to affect the tool by requiring new features, criticising functionality, or, in open source software, implementing the required feature themselves. From highly abstract levels one might describe the situation of music technologies from viewpoints such as technological determinism or social constructivism, but a closer investigation will result in an account that describes the infinite junctions of energy between the actors and the networks enrolled in the machinery of complex technologies, therefore rendering it impossible to point to an author, an independent user, or, indeed, a neutral technology.

As with all creation, often the thing created becomes remniscent of a Frankensteinian monster, taking on a life on its own, and ripping itself loose from the authority of its creator. This typically happens when software is released publicly. Suddenly the reception of the tool influences the way its author conceives of it and he is forced to re-formulate (consciously or unconsciously) the agenda or the design goals. The authors might therefore look back to the days where the tool was a small and private technology for personal expression with nostalgia, although not necessarily a regret, as all the participants expressed a joy and gratefulness for the fact that other people were using their software. In the words of Stone: "At the first release it was really exciting to find out that there are indeed some others, who are interested in the same topics!"

7.6 Conclusion

This chapter closes the loop of this investigation. This thesis started with a questioning of technology from the perspectives of philosophy of technology, phenomenology, actor-network theory and technics as a stratum with an evolutionary history outside the human. The autonomy of technology was established, and the distinct technologies described as systems that not only affect our thoughts, but also present themselves as machines external to our body, something that we control not by simple one-to-one mappings, but through more complex structures. The digital music system is a machine that we operate (rather than play),[184] and it affects our sense of embodiment. Through investigating embodiment and how our learning process is different in acoustic and digital instruments, the definition of epistemic tools was presented as a conceptual model in which digital musical systems can be analysed and understood.

This chapter demonstrated that coding can be an act of both musical composition and performance (as in livecoding). It is an act of high creativity where the artist engages with the materiality of the code itself [see section 6.6.1]. However, artistic coding is of a different nature than production systems design, due to the nature of exploration and investigation that is not common in the more top-down working methods of traditional computer science.

Contextualised this way, and considering the content of the preceding three chapters, it was appropriate to end this enquiry into expressive tools by consulting the real practitioners of audio programming languages, and their designers. What does it mean to create creative software? From where does this desire to work at the meta-level come from? Do these inventors enjoy the creation of conceptual systems for creative expression as much as they enjoy playing music with acoustic instruments? How can they defend spending their time this way? The answers were clear, and can be put into perspective with the way technology was described in

---

[184] Or in other terms, we play *with* them rather than *on* them.

Part I, namely as systems of representations inscribed into material artefacts. Contrary to both technological determinism or social constructivism, the designers and users of these systems are actively making use of technological affordances, yet resisting the technological scripts, to the degree that their resistance becomes a novel creation on its own. In many ways, this means going beyond the accepted search space or accepted technologies and practices (Boden 1991) and extend or borrow techniques from other domains into the current one. These artists, musicians and programmers showed that code is an artistic material with various properties, justa as clay or oil and canvas. However, unlike material properties, the materiality of code is of epistemic nature, one which affects, abstracts, and forms the user's mind in ways that are different from working with physical materials.

Chapter 8

Conclusion

This thesis traverses a wide interdisciplinary field; in this conclusion the central questions and themes are revisited. Future research is proposed and the thesis ends with a note on the cultural responsibility of the systems designer, i.e., an encouragement to programmers and designers to engage with, and question, their philosophical, music theoretical, and aesthetic stance when inventing new musical systems.

8.1 Prelude

This thesis has investigated extensively the effects digital technologies have upon creativity and artistic work processes. It has opened up and explored the question of what happens when creative individuals take technology into their own hands and create their own expressive tools. For this type of analysis, the questioning of technology, and our relationship with it was central. Chapter 2, therefore, looked at the question of whether humans shape or are shaped by technology, concluding that technology is too intricate a subject matter to be served by such dualistic thinking. We saw how the actor-network theory provides us with analytical tools for studying which factors are at play behind the complex structure of a scientific theory or a technological product. In chapter 3 the focus was on the body in relation to digital (non-mechanical and non-physical) technologies and how our cognitive processes can be extended into the environment. The chapter paved the way for the discussion of the digital tool's epistemically mediated embodiment. Chapter 4 followed that thread, and dealt with material epistemologies, i.e., how our digital artefacts can inhere complex models of thought; models that are necessarily culturally conditioned and domain specific. If cognitive processes extend into material objects, it becomes important to explore how epistemic tools, i.e., objects conditioned by a specific world view, influence our thinking. The chapter argued that technology is always of a political, ethical, and aesthetic pertinence; it has ergonomic implications and it structures our work practices. Technology is never neutral, but it does get its ultimate meaning in the way it is utilised by a human actor. An important imperative byproduct of this thesis has been to point to the problems of music software design, and the responsibility of the software designer. It has also been noted how some technologies originate as resistance or antiprograms to the powerful script of the large and omnipotent software companies.

A preliminary methodology for the analysis of the materiality and virtuality of musical technologies is one of the results of this research. This methodology makes an explicit account of technology as a constitutive element of human agency, and as a phenomenon that poses ever new contexts for embodied relationships with machinic artefacts. This description has illustrated the ways in which human cognition is principally non-symbolic at the lowest level of embodiment, becoming symbolic at higher abstractions where much of the interaction with digital musical instruments takes place. From a cognitivist point of view, it can be argued that the cognitive faculties used in the performance of the acoustic and the digital are not the same due to this difference in embodiment and practice. The digital musical instrument was consequently defined as an ideal example of an *epistemic tool*.

This thesis has explored how the computer, as the ideal meta-machine, continually spawns new languages, paradigms, protocols, and conceptual structures that, when used to create musical instruments, pose problems in terms of the time needed for embodied incorporation, or learning. The meta-machine's capacity of being programmable makes it an essential tool in the analysis, categorisation, algorithmisation, and restructuring of the world, such that it can be parsed in the digital domain. It becomes a machine of thinking; of perceiving and interpreting the world. The focus has centred on the relevance of the symbolic machine in the field of music and the effects it has on musical production and cultures. When the computer has been thus defined as a symbolic machine, inscribed with ideologies, it is perhaps appropriate here, in the conclusion, to quote Guattari, who is clear on this issue of how technologies mark and proscribe our expressive scope:

> Desire machines, aesthetic creation machines, are constantly revising our cosmic frontiers. As such, they have a place of eminence in the orderings of subjectivation, which are themselves called upon to relay our old social machines that are unable to follow the efflorescence of machinic revolutions that are causing our time to burst apart at every point. (Guattari 1993: 26)

It is succinctly concluded that virtuosity in the digital domain is a different type of virtuosity, that of the epistemic tool; a mastery of cognitive structures and not that of embodiment. When this virtuosity is coupled with an "aesthetic creation machine," the importance of its musico-theoretical inscriptions becomes incontestable due to the ways in which the symbolic system prime the performer into particular thought patterns.

In these final sections of this thesis, I will reiterate what has been achieved in this journey, and discuss what can be concluded with regards to the design of digital musical systems and their future paths of development. This involves rounding up the phenomenological and epistemological arguments, and pointing at issues in the design of digital musical instruments that are germane due to their strong aesthetic implications for musical culture.

8.2 Problems of Digital Musical Instruments

The current problems of computer-based music production are many and diverse, relating to the human body, music theory, hardware protocols, software design, and the integration between different hardware and software. Below I summarise the main areas identified in this thesis:

Lack of embodiment. As the survey in section 5.2 showed, people see many positive aspects in the use of digital instruments, but they lament the lack of embodiment. In acoustic instruments, the vibration of a physical body emitting sound from its very location, with haptic feedback to the performer's body, is a deep characteristic feature. Training the body through muscular memory allows for a different mode of musical playing than that made possible in most digital systems. Research is currently being undertaken in tactile feedback in musical instruments (Howard & Rimell 2004; Marshall 2006; Birnbaum & Wanderley 2007; Berhdahl et al. 2008; Marshall forthcoming PhD), but it is yet to be seen whether there will be any commercial application.[185] This thesis has argued for the impossibility of recreating the acoustic in the digital realm, and that the nature of the two are so divergent indeed, that the differences should be embraced and made use of, rather than simulate the acoustic or complain about the absence of natural embodiment in the digital realm.

Lack of tradition. When virtuosi musicians play music, in say jazz improvisation, they draw upon the history of their instrument and their own practising of musical themes from that tradition. The flexible nature of digital instruments means that the interface is much more consciously present, and the immediacy of the acoustic instrument (the instrument as *prosthesis* of the body) is substituted for the hyper-mediacy of multiple windows, arbitrary gestural mappings and context dependent functionality. These questions of legacy and tradition in digital musical instruments were explored in the survey in section 5.2, and have recently become prominent on the academic radar (Dobrian & Koppelman 2006; Butler 2008).

Flexibility in improvisation. Related to the question of embodiment is the lack of improvisational flexibility when playing digital instruments. It was demonstrated how the digital often prescribes a usage through an enclosure, as opposed to the expressive openings of the acoustic. Typically, digital environments are systems for organising sound rather than instruments for performing music. There are good historical reasons for this in both HCI (such as the lack of innovative physical user interfaces) and computer science (work with realtime sound has only been possible in affordable computers for just over a decade). There is now a need for flexible environments that can be adapted quickly to each musical situation with as little menu-selecting and screen-gazing as possible.

---

[185] The reason for mentioning commercial success is the implication it has for the general public. With mass-manifacturing of devices they become cheap and there are standards that can be designed for in, say, Pd or SuperCollider. As an example one could mention the popular usage of the Wiimote standard in the last year's NIME conference. However, I do not want to insinuate that commercial success has necessarily anything to do with quality.

Deprivement of the visual. A constant problem for the performers and the audience of computer music is the lack of visual aspects in the performance. Section 3.3.1.2 described some reasons why this might be, for example by referencing the workings of mirror neurons. There are various solutions to this problem, such as building new gestural interfaces, visualisation, or integrated audiovisual composition. Historically, visualisation of sound or sonification of visuals is an old artistic dream, perhaps epitomised in personal synaesthetics of Kandinsky, Messiaen, and Scriabin. The audiovisual work of Oscar Fischinger and Norman McLaren is also pertinent here.

Collaboration. All musical partners create their own ways of working but digital music systems are not providing sophisticated facilitation of musical collaboration. Of course, people can send project files via email, but netbased project reservoirs akin to the CVS or SVN[186] (version control systems of code) ought to be available. In this way, collaborators could follow the changes made by each other, even forking the project, which allows for simultaneous work on various versions of the musical piece. Furthermore, collaboration in a live context, whilst performing, has not reached sophisticated levels. This is not as easy to implement as it might seem. The temporal nature of music poses unique problems in terms of efficient solution to machine syncing. It is not a trivial task to share time-clocks between different instruments, and research in making multi-user performative instruments is really in its infancy, although protocols such as OSC (Wright 2003) have been developed for this very purpose.

Composers and performers. A non-existent relationship. When thinking about digital musical instruments, we do not relate them to the classical setup of composers writing music for trained performers. This is both a positive and a negative fact: positive in that we are free from hierarchical traditions and elitism, but negative as composers rarely write for electronic musical instruments. As described in this thesis, the composer becomes simultaneously an instrument designer and a performer. The act of composition dissolves into the act of instrument design, and often performance as well. A question for the future will be whether composers will ever write for new digital musical systems. Is it possible to create such a standard instrument? How do we get to know their potential, compare them and compose for them? (Butler 2008). For most practitioners, this might not be a serious problem, but it is an interesting question nevertheless.

Embedded music theory. Many of the music production tools available today embody a certain perspective on how music should be made and what it should sound like. This is productive for some musical genres, but less productive for others; consider traditions, such as the Arabic or the Indian, or even simply western Folk. The amount of systematic structures in the digital instrument can also lead to the diminution of expression. Whereas Machover (Rowe 2001: 5) argues for time better spent than practising acoustic instruments, celebrating how easy

[186] http://subversion.tigris.org/

digital instruments can be designed for play, Jordà (2005) stresses that the instrument should also be capable of bad playing; a fact that gives the performer the potential of mastery and in-depth knowledge. This thesis has emphasised the origins and functions of digital musical artefacts as composed instruments, and pointed to their situatedness as cultural products.

Un-intelligent tools. Today most people do not recognise the rigid, inflexible and unresponsive nature of computers as a problem, probably because they have never had intelligent musical tools. It is likely though, that if we fast forward twenty years and look back to the 2000s, we will be bemused by how primitive and unresponsive our tools used to be from the perspective of AI and embodied-epistemic interaction design. The computer's understanding of pitch, harmony and rhythm is still in its infancy: pitch followers and temporal onset detectors are relatively unsophisticated and problematic in their use. With regards to performance, the computer is far too vacuous as a co-player in musical performance, so limited in functionality that the human performer quickly becomes disengaged and bored. This situation differs significantly with regards to academia and commercial software; whilst various academic projects have been successful (Lopez de Mantaras & Lluis Arcos 2002; Collins 2006), it remains to be seen whether this will manifest in the form of successful commercial products.

Sketching. Contemporary music production tools are good for top-down organised production of music, but less suitable for sketching, sonic brainstorming and the exploration of musical ideas. The instruments are simply not open and dynamic enough. This applies to both sound design and pattern generation. The surveys reported on in this thesis have corroborated this, suggesting that there is a need for inspirational and flexible interfaces that support quick intuitive response, bodily gestures, and a conceptual field of a generative nature onto which composers or performers can offload their musical ideas quickly and clearly. The design of ixiQuarks has aimed to address these needs.

8.3 Contextualisation, Implications and Future Scope

If the history of musical instruments is studied, the development from plain tools to more complex technologies can be traced. Initially, instruments were simple, the learning curve was fast, and their expressive scope easily explored. With increased technological sophistication, instruments became more complex (mechanical keys in flutes, acoustic technologies of wood, and complex machinery in instruments such as the piano). Furthermore, it becomes clear how musical content is increasingly being inscribed into the musical instrument. At the moment, we are at a stage where we have created digital musical machines: tools that assist us in arranging and performing our conceptual ideas, our music. These tools range from simple sound generators triggered by some input or another, to sophisticated compositional environments where timbre, melody, harmony, rhythm, and other compositional structures can be defined in great detail. In section 7.2.4 it was suggested that the act of performing the digital instrument does not fit the categories of playing or operating, but becomes rather a mixture of both, along with other elements, such as composing. Closely related to this observation of the performance of digital systems is how the *design* of the digital instrument is also a process of composition, and even performance (as the instrument is often made in realtime, such as in livecoding).

Common commercial music tools have yet to reach the fourth stage where the tool becomes an intelligent co-player, a fellow improviser or a collaborator in the musical process. What I have in mind is a software that would learn from and adapt to the user (and the online general user-base) in both performing and composing. Techniques such as *machine learning* (artificial neural networks – good for system understanding of the user), *genetic algorithms* (for adaptive and evolutionary computation – the system adapts to the users and generates patterns evaluated by them), and *artificial intelligence* techniques (various expert systems for evaluating aesthetic and musicological structures) are useful for creating this artificial musical collaborator. As mentioned above, outstanding research is taking place within academic settings, though very little has appeared for public consumption. Furthermore, users will want to collaborate over the internet and define sounds, patterns and musical styles together. Future systems will contribute to the reservoir of music by uploading these musical data/aesthetic criteria to databases accessible to other (identical or not) intelligent systems to learn from.[187]

Below I provide a bullet point list of areas that I see as prominent in the next generation of musical tools:

- Creative interfaces for improvisation and embodied musical playing
- Modularity in software *and* hardware
- AI technologies helping/guiding/serving the user in sound design
- Online databases with intelligent search (for sounds, styles and instruments)
- Netbased projects with version control for collaborative music production
- Collaborative and distributed interfaces allowing for co-located musical playing
- Scripting possibilities for algorithmic composition and instrument creation
- Understanding of the musical fundaments of the new markets (such as China and India) and producing software that adapts easily to all musical traditions
- Sophisticated mapping engines for physical interfaces (NIME)
- Support for better audiovisual integration (for VJs, sound visualisation, etc)
- Interesting interfaces for multi-channel (surround) audio composition
- Ubiquitous computing: (ex)porting applications to mobile/hybrid devices
- Cross-platform meta-environments for the creation of instruments and generative music

[187] That way novice users in a new musical genre could get assurance that what they are doing is "in accordance" to the genre. And through clever interface design, designers could implement ways that allow the user to "expand" or "break with" the genre in inspiring ways, thus encouraging musical originality.

8.4 Points in the Design of Digital Musical Systems

The lack of real physicality in digital musical instruments (even if their interface is physical it is not their body that resonates) raises questions for their design. The possibilities of mapping are practically infinite. The question becomes that of conventions, language and style. As studied in the various sections in Part II, it is possible to invent a coherent semiotic language for any digital musical instrument. The ixi software project has devised and made extensive use of a loose symbolic language heavily made use of in the various ixiQuarks instruments [section 6.3]. The ixi approach is specific, but from it we can extract a more general orientation towards the arbitrary design of mappings and symbolism in graphical interfaces:

- Designers of musical systems should ideally be aware of the musical and cultural implications of their design. They should question the process in which they analyse, classify, abstract and produce musical affordances in the form of tools. Although it can be good to have a target group in mind, it has to be noted that such a group is never a unity. In terms of artistic expression, there are only individuals. Nevertheless, software is often designed with specific cultures in mind, normalising them further into their aesthetic styles.
- Although a strong case of the distinction between a compositional and performative environments has been made in this thesis (as these categories blur in digital instruments), it is important to understand where on the continuum a specific software fits. The epistemic dimension space provided in section 5.6 makes those parameters explicit. Ideally the instrument should be adaptable to the performer, who, in turn, will always be aware of the instrument's epistemically mediated, or virtual, embodiment.
- Sympathetic as I am towards the conceptualisation of the physical controller and the sound engine as one musical instrument, it is also important to be able to leave this perspective, and acknowledge the digital musical instrument as something that can be distributed in *space* (various gadgets in different places can control the same sound engine), *time* (the performance does not have to be realtime with respects to the performers), between *multiple performers* (as opposed to most acoustic instruments, the digital instrument can be multi-user), or as part of a *larger structure* (one performer can control various instruments). These examples indicate clearly the prominent feature of mapping in digital systems.
- As digital technologies, and software in particular, are easily adaptable (in comparison with hardware in any case), it should be an accepted practice to support open communication protocols, provide APIs (Application Protocol Interfaces), scripting, plug-in architectures, and the subclassing of code elements, such that technology can be adapted to different (sub-)cultural contexts.

8.5 Summary – Research Questions and Contributions Revisited

This research originates from an engagement with music technologies from both theoretical and practical perspectives. The principle point of departure has been to investigate how the digital music software enframes the musician through its scripts. The thesis has provided the following:

- An account of non-symbolic learning was given, and related to the practise of acoustic instruments. The phenomenon of embodiment was questioned in the realm of digital musical instruments, concluding that here one finds a specific type of *virtual embodiment* that is always epistemically mediated. The differences in mastering both types of instruments was covered by using the concept of habituation, establishing the fact that virtuosity in digital musical instruments is of a different kind, one in which design, composition, and performance combine in one activity that is primarily one of conceptual understanding, mapping, and data representation by the use of interfaces.
- The concept of *epistemic tools* was presented. The aim with this neologism is to denote technologies that are effectively machines of abstracted reality, through a complex process of design that is always culturally conditioned. Epistemic tools serve as semiotic machines in that people can delegate parts of their cognitive processes onto the cognitive artefacts, using them to store, represent and manipulate ideas that otherwise would have to be kept in working memory. For all creative activities, music in particular, such technologies are of aesthetic, ethical, and political importance.
- The importance of the concepts of virtual embodiment and epistemic tools in music technologies, points directly to their relevance for the field of *ethnocomputing*. If this research field is to study the interfaces between computational systems and culture, then the fields of art that use computers as part of their creative expression, and music in particular, are the ideal testbeds for investigations in ethnocomputing, i.e., in how cultural systems influence human creativity. This, of course, relates to the field of software studies, where software is analysed in a way similar to how we would engage critically with literature or film.
- This research has contributed to the field of music technology in various respects and from diverse angles. It has been demonstrated that the design of digital musical instruments is based on arbitrary conventions; how the creation of an *interaction model*, or a design ideology, can benefit both designers and users of such systems. The concept of epistemic tools is useful here, for the design and analysis of digital musical systems.
- The *empirical studies*, in the form of surveys and questionnaires, performed in this research contribute to the fields of music instrument design and HCI. These studies have dealt with digital technologies from a phenomenological perspective and through

various approaches, such as taxonomical analysis and ontogenetic descriptions, the covert differences in acoustic and digital instruments have been made explicit. This thesis has covered areas that have not been researched indepth within the computer music research community before, and emphasised unconventional aspects in the design of digital technologies.

- Finally, this research has been successful in that the practical basis that informs its theoretical investigations has been highly popular and lives its own life on the Internet. The *ixiQuarks* get high numbers of downloads every month, and are used by people all over the world for live improvisation, studio use, and explorations in music. Together with SuperCollider, this software proves to be a powerful environment that combines the intuitive directness of constrained interfaces and the expressive power of a textual, state-of-the-art audio programming language. It is my hope that through ixiQuarks, people will discover the worlds of free musical thinking, limitless expression, elegance, and the enjoyment that SuperCollider can provide.

8.6 Future Work

A thesis as interdisciplinary and extensive as the current one inevitably touches on many things which could not be dealt with in sufficient depth. This is the nature of all interdisciplinary research. Out of this current work a long list of potential research directions awaiting investigation. Some of those will be explored in the near future, such as:

- **The instrument: interface and engine.** A comparative empirical study of people's perception of what the digital musical instrument is from a phenomenological and epistemological perspective. This future experiment would study whether, and to what degree, musicians perceive the embodied interaction and the music theoretical parts as belonging to the hardware or software interface, the mapping engine, or the sound engine. Here we would question what the instrument's *body* is considered to be. Naturally, this will not result in a dichotomous distinction between software and hardware, but a clearer understanding of *how* people experience and conceptualise the distinct elements that make up the digital musical instrument [see section 7.2.2].
- **The audio programming language and its affordances:** An ethnographical account of studying a new audio programming language, described from the perspective of epistemic tools: how technological systems encode a specific world view. A language will be chosen, distinct enough from SuperCollider, such as Impromptu or ChucK – the former being more interesting as it is based on Scheme/Lisp which is a programming language family quite distinct from the Smalltalk/C++ architecture of SuperCollider,

thus of a different conceptual framework. This study would involve a group of people, and result in a conglomeration of comparative accounts.

- **The psychology of performance:** Cognitive research on the mental faculties used in the performance of acoustic and digital instruments could be explored more indepth through methods of experimental psychology. The aim would be to gain more quantitative data of the topics explored above, than achievable through the use of qualitative surveys in this thesis.
- **Investigation on Epistemic Dimension Spaces:** For illustrative purposes I mapped various instruments to the epistemic dimension space [see section 6.6]. A more statistical approach would be beneficial, where a number of specialists in the field would participate in an experiment where each participant maps his or her own understanding of a particular technology onto the space.
- **Development in ixiQuarks:** Further support for hardware interfacing with ixiQuarks is planned. There is already a muio support (where analogue data from sensors are converted to parameters within the ixiQuarks environment), but the intension is to open up to other protocols and hardware types, such as MIDI and Wiimote. Various instruments and effects are already in the pipeline and will be part of ixiQuarks v. 7.
- **Cultural adoptions and adaptions of music technologies.** This strand of research focuses on how culturally conditioned technologies travel. The questions here are many: What are the historical, technological and artistic foundations of creative environments for music? How do musical tools embody culturally conditioned musical practice? What are the structures of adoption and appropriation when complex technology is introduced into a tradition? What is gained and what is lost in the transition from a historical technology to a (seemingly neutral) global technology? Furthermore, how do traditions with different metaphysical worldviews deal with embodiment in musical practice through digital tools? Can intelligent tools, musical automata and AI agents be fulfilling musical partners in all cultures? What are the histories of humans performing with technology and how do they differ?

8.7 Postlude

This thesis has traversed a territory that extends over many sectors of contemporary thought, such as philosophy of technology, phenomenology, science and technology studies, computer engineering, human-computer interaction, media theory, software studies, music theory, and instrument design. Whilst reflecting the interdisciplinary nature of contemporary research in digital musical instruments, such as that encountered at the NIME or ICMC conferences, its basis has been the philosophy of technology and the view of technology as a vital structuring element in contemporary culture.

The journey from the phenomenology to the epistemology of musical instruments took the following form in this thesis: first it introduced Ihde's phenomenological modalities of tools. It was proposed that we could define, with some rough generalisations, the acoustic instrument as a channel for an *embodiment relation* with the world, and the digital instrument as medium for a *hermeneutic relation.* Through an analysis of enactivism, the highly embodied nature of skill acquisition and performance of acoustic instruments was identified, a path that was forked in order to introduce the epistemological fundament of designed artefacts. An exploration of the extended mind helped to clarify how things external to the body can become part of its cognitive mechanism, thus rendering technology as an integral element in musical creativity. The thread of embodiment was picked up again by exploring the different phenomenological relationships we have with acoustic and digital instruments, due to the distinct nature of their interfaces, and this divergence was described with regards to the conceptual and system design complexity integral to all digital systems.

This thesis has demonstrated how the mastery of an acoustic instrument is essentially an incorporating process of sub-symbolic skill acquisition. This posed a question regarding the primary characteristics in the design and learning of a digital musical instrument. It was shown how the digital system is more likely to be based on a symbolic structure, and thus include a stronger degree of compositional framework that presents itself as being of the instrument itself. However, all such structures (of mapping and composition) can be easily changed and adapted to situations as they are naturally of an arbitrary design. Composition and/or the design of digital musical systems was therefore defined as a process of mapping. Conversely, mapping in such systems necessarily includes decisions of a compositional nature. This results in the blurring of composition and instrument design, of composer and luthier, and indeed of composer and performer.

Consequently, the digital instrument was defined as an *epistemic tool* (an encoded artefact of knowledge used by an extended mind) and its symbolic nature was described as a designed artefact that affords cognitive offloading by the thinker or the performer. Although acoustic instruments may contain epistemic dimensions as well, a divergent factor in acoustic and digital instruments is the difference in mapping between a gesture that affects real vibrating material, on the one hand, and an action that is arbitrarily mapped to a symbolic system, on the other. Another fundamental difference of the acoustic versus the digital is that although both inhere knowledge of acoustics, the latter is typically designed from the top-down activity of classifications (of sounds, gestures and musical patterns), where nothing is given for free by nature. The digital instrument is an artefact based on purely rational foundations, and as a tool yielding hermeneutic relations, it is characterised by its origins in a specific culture. This portrayal highlights greater responsibilities for the designers of digital tools, in terms of aesthetics and cultural influence, as these are more symbolic, and of a more prominent compositional pertinence than our physical tools.

This research has therefore often focused on differences in the acoustic and the digital at the expense of similarities, and divided into distinct groups phenomena that are best placed on a continuum. This has been done not in order to claim preference for one type or the other, but for the sake of creating an awareness; as creators and users of instruments, of how much intelligence our instruments contain, where this knowledge is derived from, and at what level it resides. The project of exploring music technologies in this manner thus becomes a truly philosophical, aesthetic, and ergonomic investigation that can benefit from the use of a historical genealogy (as proposed by Nietzsche and practiced by Foucault) as method. The question 'why does a particular music sound like it does?' can this way be transposed into a questioning of the conditions in which musicians make music, which tools are used, and the complex origins of those tools. When music making has become a process largely taking place through digital technologies, we should bear in mind that software has agency, and necessarily inheres more cultural specifications than any acoustic instrument.

In this thesis, I have emphasised the philosophical approach – using the threads of ontology, phenomenology, and epistemology to tackle the problems of embodiment on the one hand, and technology as tools for cognitive extensions on the other – since this modus operandi has often been lacking in previous discussion of digital musical instruments. An awareness and analysis of the conceptual inscriptions in digital tools, and the complex networks that lie behind their manifestations, is becoming increasingly important at the current stage of development. It is to this area that this work contributes, with the hope that designers of future technologies for expression will become increasingly aware of the cultural context in which they work, and how their categorisations can affect innumerable cultural practices all over the world. It might therefore be appropriate to end this thesis by reverberating an observation from early in this work that technology has indeed never been neutral.

Bibliography

Ackrill, J.L. (1987). *A New Aristotle Reader.* Oxford: Clarendon Press.

Adorno, Theodor W. (1955), "The aging of the new music." in *Theodor W. Adorno: Essays on Music*. Berkeley, Los Angeles: University of California Press, 2002. pp. 181-202.
----. (1970). *Aesthetic Theory*. New York, London: Continuum, 2004
----. (1973). *Philosophy of Modern Music.* New York: The Seabury Press.

Ahrens, Christian. (1996). "Technological Innovations in Nineteenth-Century Instrument Making and Their Consequences" in *Musical Quarterly.* vol. 80 (3). Oxford: Oxford University Press. pp. 332-40.

Akrich, Madeleine. (1992). "The de-scription of technical object" in *Shaping Technology/Building Society.* (eds.) Bijker, W & Law, J. Cambridge: MIT Press.
----- & Latour, Bruno. (1992). "A summary of a convenient vocabulary for the semiotics of human and nonhuman assemblies" in *Shaping Technology/Building Society.* (eds.) Bijker, W & Law, J. Cambridge: MIT Press.

Andersen, Peter Bøgh. (1992). "Computer semiotics" in *Scandinavian Journal of Information systems*. Vol. 4: 3-30. Copenhagen.
----- & May, Michael. (2001a). "Instrument Semiotics" in *Information, organisation and technology*. Studies in organisational semiotics. (eds. Liu, Kecheng; Clarke, Rodney J.; Andersen, Peter B.; Stamper, Ronald K.). Kluwer: Boston/Dordrecht/London. pp. 271-298.
-----. (2001b). "What semiotics can and cannot do for HCI" in *Knowledge Based Systems*. Elsevier.

Anderson, Michael L. (2003). "Embodied Cognition: A Field Guide" in *Artificial Intelligence.* vol. 149. pp. 91-130.

Anonymous. *120 Years of Electronic Music: Electronic Musical Instrument 1870-1990*. <http://www.obsolete.com/120_years> [accessed July 2007].

Apple Computer Inc. (1987). *Human Interface Guidelines: The Apple Desktop Interface.* Boston: Addison-Wesley.

Arfib, D., Couturier, J-M., and Kessous, L. (2005). "Expressiveness and digital musical instrument design. *Journal of New Music Research* vol. 34 (1) pp. 125-136.

Arkin, Ronald C. (1998). *Behavior Based Robotics.* Cambridge, MA: MIT Press.

Armstrong, Newton. (2006). *An Enactive Approach to Digital Musical Instrument Design* (PhD thesis). New York: Princeton University.

Augoyard, Jean-Francis & Torgue, Henry. (2005). *Sonic Experience: A Guide to Everyday Sounds*. Montreal: McGill University Press.

Austin, Kevin. (2009). "The timeline matter; object and process, instrument and tool". Email to the *Cec-Conference mailing list* on February 8th, 2009.

Ascott, Roy. *Turning on Technology*. <http://www.cooper.edu/art/techno/essays/ascott.html> [accessed May 2005]

Attali, Jacques. (1985). *Noise: The Political Economy of Music*. Minneapolis: University of Minnisota Press.

Bahn, Curtis & Trueman, Dan. (2001). "interface: electronic chamber ensemble" in *Proceedings of NIME 2001* <http://hct.ece.ubc.ca/nime/2001/program.html> [accessed May 2005].

Bailey, Derek. (1993). *Improvisation: Its Nature and Practice in Music.* New York: Da Capo Press.

Baird, David. (2004). *Thing Knowledge: A Philosophy of Scientific Instruments.* Berkeley: University of California Press.

Baker, Ann C.; Jensen, Patricia J. & David A. Kolb. (2002). *Conversational Learning: An Experiential Approach to Knowledge Creation*. Santa Barbara: Greenwood Publishing Group.

Bannon, Liam and Bødker, Susanne. (1991). "Beyond the Interface - Encountering Artifacts in Use" in *Designing Interaction: Psychology at the Human-Computer Interface*. (ed.) Carroll, John M. Cambridge University Press pp. 227-253.

Barbosa, Álvaro; Kaltenbrunner, Martin & Geiger, Günter. (2003). "Interface Decoupled Applications for Geographically Displaced Collaborations in Music" *Proceedings of ICMC 2003*.

Barthes, Roland. (1972). *Mythologies.* London: Paladin.

Barnet, Belinda. (1999). "Machinic heterogenesis and evolution: collected notes on sound, machines and Sonicform". *M/C: A Journal of Media and Culture.* vol. 2(6). <http://www.uq.edu.au/mc/9909/sonic.php> [accessed May 20, 2008].
-----. (2004). "Technical Machines and Evolution" in cTheory (eds. Kroker, Arthur & Kroker, Marilouise). article: a139. <www.ctheory.net/articles.aspx?id=414> (August 15, 2007).
-----. (2004). "Material Cultural Evolution: An Interview with Niles Eldredge" in *Fibreculture Journal.* <http://journal.fibreculture.org/issue3/index.html> [accessed July 2008].
-----. (2006). "Engelbart's Theory of Technical Evolution" in *Continuum: Journal of Media & Cultural Studies.* vol 20 (4). pp. 509-521.

Baudrillard, Jean. (1988a). *Selected Writings.* Cambridge: Polity.
-----. (1988b). *The Ecstasy of Communication.* New York: Semiotext(e).

Beaudouin-Lafon, Michel. (2000). "Instrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces". In Proceedings of the Conference on Human Factors in Computing Systems (CHI 2000).
-----. (2000). "Reification, Polymorphism and Reuse: Three Principles for Designing Visual Interfaces" in *Proceedings of the working conference on Advanced visual interfaces*, Palermo, Italy.
-----. (2004). "Designing Interaction, not Interfaces" in Proceedings of the working conference on Advanced visual interfaces, Gallipoly, Italy.

Bencina, Ross. (2005). "The Metasurface: Applying Natural Neighbour Interpolation to Two-to-Many Mapping" in *Proceedings of NIME 2005*. (eds.) Fels, Sidney & Blaine, Tina. Vancouver: University of British Columbia.

Benjamin, Walter. (1969). “The Work of Art in the Age of Mechanical Reproduction” in *Illuminations*. (ed.) Hannah Arendt. New York: Schocken Books. pp. 217-251.

Berger, P. & Luckmann, T. (1967). *The Social Construction of Reality* . Garden City, NY: Doubleday.

Berhdahl, Edgar; Steiner, Hans-Christoph & Oldham, Collin. (2008). “Practical Hardware and Algorithms for Creating Haptic Musical Instruments,” in *Proc. of New Interfaces for Musical Expression (NIME)*. Genova, Italy.

Bertelsen, O. & Bødker, S. (2002). "Activity theory" in *HCI Models, Theories and Frameworks* (ed. John Carroll). Morgan Kaufmann, pp. 291-324.
-----. Breinbjerg, Morten & Pold, Soren. (2007). "Instrumentness for Creativity: Mediation, Materiality & Metonymy" in *Proceedings of Cognition and Creativity 2007*. Washington: ACM Library.

Bevilacqua, Frederic; Muller, Remy & Schnell, Norbert. (2005). "MnM: a Max/MSP mapping toolbox" in *Proceedings of NIME 2005*. (eds.) Fels, Sidney & Blaine, Tina. Vancouver: University of British Columbia.

Bigus, Joseph P & Bigus, Jennifer. (2001). *Constructing Intelligent Agents Using Java.* New York: Wiley.

Bijker, Weibe, B; Hughes, Thomas P. & Pinch, Trevor. (1987) *The social construction of technological systems: New directions in the sociology and history of technology*, Cambrdige: MIT Press.
-----. & Law, J. (1992). *Shaping Technology/Building Society.* Cambridge: MIT Press.
-----. & Law, J. (1992). "General Introduction" in *Shaping Technology/Building Society.* Cambridge: MIT Press.
-----. (1995). *Of Bicycles, Bakelites, and Bulbs: Toward a Theory of Sociotechnical Change*. Cambridge: MIT Press.

Bijsterveld, Karin & Schulp, Marten. (2004). "Breaking into a World of Perfection: Innovation in Today's Classical Musical Instruments" in *Social Studies of Science.* vol. 34, p. 649.

Birnbaum, D; Fiebrink, R; Malloch, J; Wanderley, M. (2005). "Towards a Dimension Space for Musical Artifacts" in *Proceedings of NIME 2005*. (eds.) Fels, Sidney & Blaine, Tina. Vancouver: University of British Columbia.
----- & Wanderley, Marcelo. (2007). "A Systematic Approach to Musical Vibrotactile Feedback" in *Proceedings of ICMC 2007.* Copenhagen: ICMA.

Bishop, J.M & Nasuto, J.S. (2005). "Second-order cybernetics and enactive perception" in *Kybernetes* Vol. 34 No. 9/10.

Blackwell, Alan. (2003). "Cognitive Dimensions of Tangible Programming Languages" in *Proceedings of the Joint Conference. EASE & PPIG*. <http://www.ppig.org/papers/15th-blackwell.pdf> [accessed August 2008]

Blaine, Tina. (2005). "The Convergence of Alternate Controllers and Musical Interfaces in Interactive Entertainment" in *Proceedings of NIME 2005*. (eds.) Fels, Sidney & Blaine, Tina. Vancouver: University of British Columbia.

Blauner, R. (1964). *Alienation and Freedom.* Chicago: Chicago University Press.

Boden, Margaret A. (1990). *The Creative Mind: Myths and Mechanisms.* London: Wiedenfield

and Nicholson.
-----. (2006). *Mind as Machine: A History of Cognitive Science.* Oxford: Oxford University Press.

Bolter, Jay David & Grusin, Richard. (1999). *Remediation: Understanding New Media.* Cambridge: MIT Press.

Bongers, Bert & Harris, Yolande. (2002). "A Structured Instrument Design Approach: The Video-Organ" in *NIME 2002 Proceedings*. Limerick: University of Limerick, Department of Computer Science and Information Systems.
-----. (2000). "Physical Interfaces in the Electronic Arts. Interaction Theory and Interfacing Techniques for Real-time Performance." In M. Wanderley and M. Battier, (eds). *Trends in Gestural Control of Music*. Paris: Ircam - Centre Pompidou.
-----. (1993). "The Use of Active Tactile and Force Feedback in Timbre Controlling Electronic Instruments", Proceedings, International Computer Music Conference.

Borgman, Albert. (1984). *Technology and the Character of Contemporary Life.* Chicago: The University of Chicago Press.

Born, Georgina. (1995). *Rationalizing Culture: IRCAM, Boulez, and the Institutionalization of the Musical Avant-Garde.* Berkeley.

Bourdieu, Pierre. (1977). *Outline of a Theory of Practice*. Cambridge: Cambridge University Press.
-----. (1984). *Distinction: A Social Critique of the Judgement of Taste.* London: Routledge.
-----. (1986). "The forms of capital" in *The Handbook of Theory: Research for the Sociology of Education.* Richardson, J.G. (ed). New York: Greenwood Press, pp. 241-258.
-----. (1990). *The Logic of Practice.* Cambridge: Polity Press.

Bowers, John. (2002). *Improvising Machines: Ethnographically Informed Design for Improvised Electro-Acoustic Music*. School of Music, University of East Anglia, Norwich, UK. <http://www.ariada.uea.ac.uk/ariadatexts/ariada4/html/> [accessed May 2008].
----- & Archers, Phil. (2005). "Not Hyper, Not Meta, Not Cyber but Infra-Instruments" in *Proceedings of NIME 2005*. (eds.) Fels, Sidney & Blaine, Tina. Vancouver: University of British Columbia.

Bowker, Geoffrey C & Star, Susan Leigh. (2000). *Sorting Things Out: Classification and its Consequences.* Cambridge: MIT Press.

Bowler, I., A. Purvis, P. Manning, and N. Bailey. (1990). "On Mapping N Articulation onto M Synthesiser-Control Parameters." In *Proceedings of the 1990 International Computer Music Conference*. San Francisco, International Computer Music Association, pp. 181-184.

Brandt, Per Aage. (1993). "Meaning and the machine: Toward a semiotics of interaction" in *The Computer as Medium*. (eds.) P. Bøgh Andersen, B. Holmqvist & J. F. Jensen. Cambridge: Cambridge University Press. pp. 128 - 140.

Bradie, Michael. (1983). *The Applied Turn in Contemporary Philosophy: Bowling Green Studies in Applied Philosophy*. Ohio: Bowling Green State University.

Braverman, H. (1974). *Labour and Monopoly Capital: The Degradation of Work in the Twentieth Century.* New York: Monthly Review Press.

Brey, Philip. (2005). "The Epistemology and Ontology of Human-Computer Interaction" in *Mind and Machines*. vol. 15. pp383-398.

Brooke, John. (1996). "SUS: a 'quick and dirty' usability scale" in *Usability Evaluation in Industry*. P. W. Jordan, B. Thomas, B. A. Weerdmeester & A. L. McClelland (eds.) London: Taylor and Francis.

Brooks, Rodney A. (1991). "Intelligence Without Representation" in *Artificial Intelligence Journal* vol. 47, pp. 139–159.

Brown, Andrew R. (2006). "Code Jamming." M/C Journal 9.6. 09 Jan. 2007 <http://journal.media-culture.org.au/0612/03-brown.php>.

Brown, John. S.; Collins, Allan. & Duguid, Paul. (1989). "Situated cognition and the culture of learning" in *Educational Researcher* 18 (1): 32-42.

Brunelly, R. et al. (2007). "Detecting Focus of Attention" in *PEACH: Intelligent Interfaces for Museum Visits*.
Stock, O & Zancanaro, M. London: Springer.

Bull, Michael. (2000). *Sounding Out the City: Personal Stereos and the Management of Everyday Life.* Oxford: Berg.

Burtner, Matthew. (2005). "Composing for the (dis)Embodied Ensemble: Notational Systems in (dis)Appearances" in *Proceedings of NIME 2003*.
<http://hct.ece.ubc.ca/nime/2003/onlineproceedings/TOC.html>

Burnes, Bernard. (1989). *New Technology in Context: The Selection, Introduction and Use of Computer Numerically Controlled Machine Tools*. Aldershot, UK: Avebury.

Burns, Tom. (1969). *Industrial Man.* London: Penguin.

Burzik, Andreas. (2003). "Go with the flow" in The Strad. London: Orpheus Publications. July, 2003.

Butler, Jennifer (2008). "Creating Pedagogical Etudes for Interactive Instruments" in *Proceedings of NIME 2008*. < http://nime2008.casapaganini.org> [accessed January 2008].

Butler, Judith. (1990). *Gender Trouble: Feminism and the Subversion of Identity*. London: Routledge.

Button, G. & Sharrock, W. (1996). "Project work: the organisation of collaborative design and development in software engineering" in *CSCW Journal*. vol. 5 (4), pp.369-386.
-----. Dourish, Paul. (1996). "Technomethodology: Paradoxes and Possibilities" in *Proceedings of the ACM Conference on Human Factors in Computing Systems CHI'96*. New York: ACM. pp. 19-26.

Bürger, Peter. (1984). *Theory of the Avant-Garde*. Minneapolis: University of Minnesota Press.

Bødker, Susanne. (1990). *Through the Interface: Human Activity Approach to Interface Design.* New Jersey: Lawrence Erlbaum Associates.
-----, Ehn, P., Kammersgaard, J., Kyng, M., & Sundblad, Y. (1987). "A Utopian experience" in *Computers and democracy: A Scandinavian challenge.* G. Bjerknes, P. Ehn, & M. Kyng. (eds.). Aldershot, UK: Avebury.
-----. (1996). "Creating conditions for participation: Conflicts and resources in systems design" in *Human Computer Interaction.* vol. 11(3), pp. 215-236.

Cadoz, Claude, Luciani, Annie; Florens, Jean-Loup & Castagné, Nicolas. (2003). "ACROE - ICA: Artistic Creation and Computer Interactive Multisensory Simulation Force Feedback Gesture Transducers" in *Proceedings of NIME 2003*. http://hct.ece.ubc.ca/nime/2003/onlineproceedings/TOC.html (May, 2005).

Cage, John. (1961). *Silence*. Hanover: Wesleyan University Press.

Callon, Michel. (1986). "Some Elements of a Sociology of Translation: Domestication of the Scallops and the Fisherman of St Brieuc Bay" in *Power, Action and Belief: A New Sociology of Knowledge?* (ed.) Law, J. London: RKP.
-----; Law, John & Rip, A. (eds.) (1987). *Mapping the Dynamics of Science and Technology: Sociology of Science in the Real World.* Basingstoke: Macmillan.
-----. (1997). "Society in the Making: The Study of Technology as a Tool for Sociological Analysis" in *The Social Construction of Technological Systems.* (eds.) Bijker, W.E; Hughes, T.P & Pinch, T. Cambridge, MA: MIT Press.

Campbell, Murray & Greated, Clive. (1987). *The Musicians Guide to Acoustics.* London: J.M. Dent & Sons Ltd.
-----; Greated, Clive & Myers, Arnold. (2004). *Musical Instruments: History, Technology & Performance of Instruments of Western Music.* New York: Oxford University Press.

Carbaugh, Donal. (2005). "Silence and Third-Party Introductions: An American and Finnish Dialogue (with Saila Poutiainen) " in *Cultures in Conversation*: http://works.bepress.com/donal_carbaugh/3 [accessed August 2008]
-----., Berry, Michael & Nurmikari-Berry, Marjatta. (2006). "Coding Personhood Through Cultural Terms and Practices: Silence and Quietude as a Finnish 'Natural Way of Being'" in *Journal of Language and Social Psychology*. http://works.bepress.com/donal_carbaugh/7 [accessed August 2008]

Card, S.K., Moran, T.P. and Newell, A. (1986). *The Psychology of Human-Computer Interaction.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Carpenter, Stanley. (1974). "Modes of Knowing and Technological Action" in *Philosophy Today.* vol 18. nr. 2. pp. 162-168.

Carrol, J.M. (2003). (ed.) *Interfacing Thoughts: Cognitive Aspects of Human-Computer Interaction.* Cambridge, MA: MIT Press.

Cascone, Kim. (2000). "The Aesthetics of Failure: "Post-Digital" Tendencies in Contemporary Computer Music." in *Computer Music Journal*, vol. 24 (4). pp. 12-18.
-----. (2001). "The Microsound Scene: An Interview with Kim Cascone" by Jeremy Turner. <http://www.ctheory.net/articles.aspx?id=322> [Accessed January 2008]
-----. (2004). "Laptop Music – counterfeiting aura in the age of infinite reproduction" in *Skriftserie: Center for Digital Æstetik-forskning.* nr. 13. Aarhus: Center for Digital Æstetik-forskning.

Chadabe, Joel. (1997). *Electric Sound: The Past and Promise of Electronic Music*. New Jersey: Prentice Hall.
-----. (2000). "Remarks on Computer Music Culture". *Computer Music Journal*, vol 24(4), pp. 9-11. Massachusetts Institute of Technology.
-----. (2002). "The Limitations of Mapping as a Structural Descriptive in Electronic Instruments" in *NIME 2002 Proceedings*. Limerick: University of Limerick, Department of Computer Science and Information Systems.

Chafe, Chris. (1993). "Tactile Audio Feedback" in *Proceedings of the International Computer Music Conference*, pp. 76–79.

Chandler, Daniel. (1992). "The Phenomenology of Writing by Hand" in *Intelligent Tutoring Media.* vol. 3 (2/3). pp. 65-74.

Chin, John P; Diehl, Virginia A. & Norman, Kent L. (1988). "Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface" in *ACM CHI'88 Proceedings*, pp. 213-218.

Chipchase, Jan. (2006). "Shared Phone Practices" <http://www.janchipchase.com/sharedphoneuse> [accessed August 2008].

Chislenko, A. (1995). "Legacy Systems and Functional Cyborgization in Humans". <http://www.lucifer.com/~sasha/articles/Cyborgs.html> [accessed August 2008].

Churchill, Winston. (1960). A quotation in *Times Magazine*. September 12, 1960.

Churchland, P. M. (1995). *The Engine of Reason, the Seat of the Soul*. Cambridge, MA: MIT Press.

Clark, Andy. (1996). *Being There.* Cambridge, MA: MIT Press.
-----. & Chalmers, David. (1998). "The Extended Mind" in *Analysis* vol. 58(1).
-----. (1999). "An Embodied Cognitive Science?" in *Trends in Cognitive Sciences* vol. 3 (9), pp. 345-351.
-----. (1999). "Mind, Brains and Tools" in *Workshop on Mental Representation*. August 1999, Maine. Also appeared (2002) in *Philosophy of Mental Representation.* Oxford: Clarendon Press.
-----. (2001). "Reasons, Robots and the Extended Mind: (Rationality for the New Millenium)" in *Mind and Language.* vol. 16 (2), pp. 121-145.
-----. (2003). *Natural-Born Cyborgs: Minds, Technologies, and the Future of Human Intelligence.* Oxford: Oxford University Press.
-----. (2007). "Curing Cognitive Hiccups: A Defense of the Extended Mind" in *Journal of Philosophy,* vol. 104. pp. 163-192.
-----. (2008). *Supersizing the Mind: Embodiment, Action, and Cognitive Extension*. New York: Oxford University Press.

Cliff, Dave. (2003). "Biologically-Inspired Computing Approaches To Cognitive Systems: a partial tour of the literature" HP laboratories publication : http://www.hpl.hp.com/techreports/2003/HPL-2003-11.pdf

Clutton, Cecil. (1961). "The Pianoforte" in *Musical Instruments Through the Ages.* (ed. A. Baines). Harmondsworth: Penguin Books.

Cohen, Louis. (2000) *Research Methods in Education.* London: Routledge.

Collins, Nick; McLean, Alex; Rohrhuber, Julian; Ward, Adrian. (2003). "Live coding in laptop performance" in *Organised Sound.* vol. 8 (3). pp. 321-330.
----- aka Nilson, Click. (2007)"Live Coding Practice". in Proceedings of NIME 2007.
----- & Blackwell, Alan. (2005). "The Programming Language as a Musical Instrument" in *Proceedings of PPIG05* (Psychology of Programming Interest Group).
-----. (2006). "Towards Autonomous Agents for Live Computer Music: Realtime Machine Listening and Interactive Music Systems". PhD thesis. Cambridge: Cambridge University.

Cook, Perry. (2001). "Principles for Designing Computer Music Controllers" in *Proceedings of*

NIME 2001. <http://hct.ece.ubc.ca/nime/2001/program.html> [accessed May 2005].

Cooper, Alan. (1999). *The Inmates are Running the Asylum, or, Why High Technology Products Drive Us Crazy and How to Restore the Sanity.* Indianapolis: SAMS.

Cott, Jonathan. (1974). *Stockhausen: Conversations with the Composer*. London: Robson Books.

Costall, Alan. (1995). "Socializing Affordances" in *Theory and Psychology*. vol. 5 (4) pp. 467-481.
-----. (1997). "The meaning of things" in *Social Analysis*. vol. 41 (1).

Costanza, E; Shelley, S. B. & Robinson, J. A. (2003). "Introducing audio d-touch: A tangible user interface for music composition and performance," in *Proc. of the 6th Int'l Conf. on Digital Audio Effects (DAFX)*, pp. 63–70.

Couturier, Jean-Michel & Arfib, Daniel. (2003). "Pointing Fingers: Using Multiple Direct Interactions with Visual Objects to Perform Music" in *Proceedings of NIME 2003*. http://hct.ece.ubc.ca/nime/2003/onlineproceedings/TOC.html (May, 2005).

Cowan, Ruth. S. (1983). *More Work for Mother: The Ironies of Household Technology from the Open Hearth to the Microwave.* New York: Basic Books.
-----. (1987). "The consumption junction: A proposal for research strategies in the sociology of technology" in *The Social Construction of Technological Systems*. (eds. Bijker, W; Hughes, T. P. & Pinch, T). Cambrdige: MIT Press.

Cox, Christoph & Warner, Daniel. (2004). *Audio Culture: Readings in Modern Music.* New York: Continuum.

Coyne, R. (1995). *Designing Information Technology in the Postmodern Age*. Cambridge, MA: MIT Press.

Craik, Kenneth. (1967). *The Nature of Explanation.* Cambridge: Cambridge University Press.

Cramer, Florian & Gabriel, Ulrike. (2001). "On Software Art" in *CTheory*. <http://www.rhizome.org/object.rhiz?2848> [accessed July 2002].
-----. (2005). *Words Made Flesh: Code, Culture, Imagination.* Rotterdam: Piet Zwart Institute. <http://pzwart.wdka.hro.nl/mdr/research/fcramer/wordsmadeflesh/> [accessed December 2008].

Cross, Ian. (1999). "AI and Music Perception" in *AISB Quarterly*, 1999, 102, pp. 12-25. <http://www.mus.cam.ac.uk/~ic108/AISB99/index.html>, [accessed June 2002].

Csikszentmihalyi, Mihaly. (1996). *Creativity: Flow and The Psychology of Discovery and Invention*. New York: Harper Collins.
-----. (2003). *Good Business: Leadership. Flow, and the Making of Meaning*. New York: Penguin Group.

Dahlstedt, Palle & Mats G. Nordahl. (2001). "Living Melodies: Coevolution of Sonic Communication" in *Leonardo*. Vol. 34, No. 3, pp. 243-248.

Daniels, Dieter. (1997). "Art and Media in the XXth Century" in *The Age of Modernism, Art in the 20th Century*. (eds) Christos M. Joachimides, Norman Rosenthal, Ostfildern: Gerd Hatje Verlag.

Dannenberg, Roger B. (2004). "Aura II: Making Real-Time Systems Safe for Music" in

Proceedings of NIME 2004. <http://hct.ece.ubc.ca/nime/2004/NIME04/paper/index.html> [accessed May, 2005].

Dant, Tim. (2005). *Materiality and Society.* Maidenhead, UK: Open University Press.

Danto, Arthur C. (1997). *After the End of Art: Contemporary Art and the Pale of History*. New Jersey: Princeton University Press.

Davidson, Philip L. & Han, Jefferson Y. (2006) "Synthesis and Control on Large Scale Multi-Touch Sensing Displays" in *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression* (NIME06), Paris, France.

Davis, Fred D. (1989) "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology" in *MIS Quarterly*. vol 13 (3), pp. 319-340.

Dawkins, Richard. (1976). *The Selfish Gene*. Oxford: Oxford University Press.
-----. (1982). *The Extended Phenotype: The Gene as a Unit of Selection*. Oxford: Oxford University Press.

De Landa, Manuel. (1997). *A Thousand Years of Nonlinear History*. New York: Zone Books.

de Laubier, Serge. (1988). "The Meta-Instrument" in *Computer Music Journal.* vol. 22 (1). pp. 25-29.
-----. & Goudard, Vincent. (2006). "Meta-Instrument 3: a look over 17 years of practice" in *Proceedings of NIME 2006*. Paris: IRCAM.

Deleuze, Gilles & Guattari, Félix. (1984). *Anti-Oedipus: Capitalism and Schizophrenia.* London: Athlone.
----- & Guattari, Felix. (1987). *A Thousand Plateaus: Capitalism and Schizophrenia*. London: Continuum.
-----. (2005). *Difference and Repetition.* London: Continuum.

Dennett, Daniel C. (1982). "Beyond Belief". in *Thought and Object*. (ed.) A. Woodfield. Oxford: Clarendon Press.
-----. (1991). *Consciousness Explained*. London: Penguin.
-----. (1998a). "Making Tools for Thinking" in *Metarepresentation.* http://ase.tufts.edu/cogstud/papers/maketoo2.htm [accessed August 2008]
-----. (1998b). "Things about Things" in *Lisbon Conference on Cognitive Science.* http://ase.tufts.edu/cogstud/papers/lisbon.htm [accessed August 2008].

DeNora, Tia. (1995). *Beethoven and the Construction of Genius: Musical Politics in Vienna, 1792-1803.* Berkeley: University of California Press.
-----. *Music in Everyday Life.* (2000). Cambridge: Cambridge University Press.

Derrida, Jacques. (1974). *Of Grammatology*. Baltimore: John Hopkins University Press.
-----. (1981). *Dissemination*. London: Athlone Press.
-----. (1989) *Of Spirit: Heidegger and the Question.* Chicago: University of Chicago Press.
-----. (1993). "The Rhetoric of Drugs. An Interview" in *Differences: A Journal of Feminist Cultural Studies.* vol 5. nr. 1. pp. 1-10.
-----. (1995) "The Word Processor" in *Paper Machine*. Standford: Stanford University Press.

de Souza, Clarisse Sickenius. (2005). *The Semiotic Engineering of Human-Computer Interaction*. Cambridge: MIT Press.

Dilthey, Wilhelm. (1990). "The Rise of Hermeneutics" in *The Hermeneutic tradition: from Ast* *to Ricoeur*. Albany: SUNY Press.

Di Paolo, Ezequiel, Rohde Marieke & Jaeger, Hanneke. (forthcoming). "Horizons for the Enactive Mind: Values, Social Interaction, and Play" in *Enaction: Towards a New Paradigm for Cognitive Science*. Stewart, J, Gapenne, O & Di Paolo (Eds), Cambridge: MA: MIT Press.
-----. (2009). "Extended Life" in *Topoi*. vol. 27 (2).

Di Scipio, A. (1998), 'Questions Concerning Music Technology', in *Angelaki: Journal of Theoretical Humanities*, 3: 2, pp. 31–40.

Dix, Alan; Finlay Janet; Abowd, Gregory & Beale, Russell. (1993). *Human-Computer Interaction*. Prentice Hall International, Hemel Hempstead.
-----. (2008). Externalisation – how writing changes thinking . Interfaces, 76, pp. 18-19. Autumn 2008.

Dobrian, Christopher & Koppelman, Daniel. (2006). "The 'E' in NIME: Musical Expression with New Computer Interfaces" in Proceedings of the 2006 Conference on New Interfaces for Musical Expression (NIME06), Paris: IRCAM.

Dogan, Mattei & Pahre, Roberg. (1990). *Creative Marginality: Innovation at the Intersections of Social Sciences.* Boulder: Westview Press.

Dorin, Alan. (2001). "Generative processes and the electronic arts" in *Organised Sound* 6(1), pp. 47-53. Cambridge University Press.
Dorin, Alan & Jon McCormack. (2001). "First Iteration: A Conference on Generative Systems in the Electronic Arts" in *Leonardo*. Vol 34 (3) pp. 239-242.

Doubleday, Veronica (2008). "Sounds of Power: An Overview of Musical Instruments and Gender" in *Ethnomusicology Forum.* Vol. 17 (1). pp. 3-39.

Dourish, Paul & Bellotti, V. (1992). "Awareness and coordination in shared workspaces". *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*. ACM Press New York, NY, USA.
-----. (2001). *Where the Action Is: The Foundation of Embodied Interaction.* Cambridge: MIT Press.
----- & Button, Graham. (1998). "On 'Technomethodology': Foundational Relationships between Ethnomethodology and System Design" in *Human Computer Interaction* vol. 13(4). pp. 395-432.

Dreyfus, Hubert L; Dreyfus, Stuart & Athanasiou, Tom. (1986). *Mind over Machine: The Power of Human Intuition and Expertise in the Era of the* Computer. New York: Free Press.
-----. (1992). *What Computers Still Can't Do: A Critique of Artificial Reason*. Cambridge, MA: MIT Press.
-----. (1996). "The current relevance of Merleau-Ponty's phenomenology of embodiment" in *Electronic Journal of Analytic Philosophy.* vol. 4
<http://ejap.louisiana.edu/EJAP/1996.spring/dreyfus.1996.spring.html> [accessed June 2008]

Duignan, Matthew; Noble, James & Biddle, Robert. (2005). "A Taxonomy of Sequencer User-Interfaces" in *Proceedings of ICMC 2005,* Barcelona: Escola Superior de Música de Catalunya.

Eacott, John. (2000). *Form and Transience: Generative Music Composition in Practise*.
<http://www.informal.org/research/FormandTransience.pdf> [accessed June 2008].
------. (2001). *Morpheus: Emergent Music*.
<http://www.informal.org/morpheus_emergent_music.htm> [accessed May 2002].

Eaglestone, Barry; Upton, Catherine & Ford, Nigel. (2005). "The Compositional Processes of Electroacoustic Composers: Contrasting Perspectives" in *Proceedings of ICMC 2005.* Barcelona, University of Pompeu Fabra, ICMA.
-----; Ford, Nigel; Brown, Guy J & Moore, Adrian. (2007a). "Information Systems and Creativity: An Empirical Study" in *Journal of Documentation.* 63 (4). pp. 443-464.
----; Ford, Nigel; Holdridge, Peter & Carter, Jenny. . (2007b). "Are Cognitive Styles an Important Factor in the Design of Electroacoustic Music Software?" in *Proceedings of ICMC.* Copenhagen: ICMA.
-----; Ford, Nigel; Holdridge, Peter & Carter, Jenny. (2008). "Are Cognitive Styles an Important Factor in Design of Electroacoustic Music Software?" in *Journal of New Music Research*. vol 37 (1). pp. 77-85

Earnshaw, Rae A. (2001). *Frontiers of Human Centered Computing, Online Communities and Virtual Environments*. London: Springer.

Eco, Umberto. (1976). *A Theory of Semiotics*. London: Macmillan.

Ehn, P. & Kyng, M. (1987). "The Collective Resource Approach to Systems Design" in *Computers and Democracy - A Scandinavian Challenge*. Bjerknes, G., Ehn, P., & Kyng, M. (Eds.) (pp. 17–58). Aldershot, UK: Avebury.
-----. (1988). *Work-oriented design of computer artifacts*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Eldredge, Niles & Temkin, Ilya. (2007). "Phylogenetics and material cultural evolution" in *Current Anthropology.* vol. 48 (11). pp. 146-153.

Eisenstein, J.; Rich, C. (2002). "Agents and GUI's from Task Models", ACM International Conference on Intelligent User Interfaces. pp. 47-54.

Ellul, Jacques. (1964). *The Technological Society*, New York: Vintage Books. (the French original published in 1953).

Engelbart, Douglas C. *Augmenting Human Intellect: A Conceptual Framework*. Summary Report AFOSR-3223 under Contract AF 49(638)-1024, SRI Project 3578 for Air Force Office of Scientific Research, Stanford Research Institute, Menlo Park, Ca., October 1962.

Eno, Brian. (1983). "The Studio as Compositional Tool." in *Downbeat* 56–57, 50–52 (July/August 1983).
-----. (1999). "The Revenge of the Intuitive" in *Wired* 7.01, January issue.

Erickson, Thomas D. (1995). "Working with Interface Metaphors" in *Readings in Human-computer Interaction: Toward the Year 2000*. (eds.) Baecker, R & Greenberg, S. San Francisco: Morgan Kaufmann.

Evens, Aden. (2005). *Sound Ideas: Music, Machines, and Experience*. Minneapolis: University of Minnesoda Press.

Ezzl, Karlheins. (2007). *Composing in Cyberspace*. <http://www.essl.at/bibliogr/cyberkomp-e.html> [accessed February 2009].

Feenberg, Andrew. (1991). *Critical Theory of Technology.* Oxford: Oxford University Press.
-----. (1999). *Questioning Technology.* London: Routledge.

Feiner, S; MacIntyre, B; & Seligmann, D. (1993). "Knowledge-based augmented reality" in *Commun. of the ACM.* vol. 37(7). July 1993. pp. 52-62.

Fels, Sidney and Hinton, G. (1998). "Glove-TalkII: A neural network interface which maps gestures to parallel formant speech synthesizer controls". *IEEE Transactions on Neural Networks*. vol. 9 (1) pp. 205-212.
-----. “Designing for Intimacy: Creating New Interfaces for Musical Expression” in *Proceedings of the IEEE*. Volume 92. No. 4. Pages 672--685. 2005.

Ferré, Fredric. (1988). *Philosophy of Technology.* Englewood Cliffs, NY: Prentice-Hall.

Fiebrink, Rebecca; Wang, Ge & Cook, Perry. (2007). "Don't Forget the Laptop: Using Native Input Capabilities for Expressive Musical Control" in *Proceedings of NIME 2007.*

Fishkin, K. P. (2004). "A Taxonomy for and Analysis of Tangible Interfaces" in *Journal of Personal and Ubiquitous Computing* 8 (5), pp. 347-358.

Fitzmaurice, G. (1993). "Situated Information Spaces and Spatially Aware Palmtop Computers" in *Commun. of the ACM*. vol. 37(7). July 1993. pp. 38-49.

Fitzpatrick, Geraldine. (1998). *The Locales Framework: Understanding and Designing for Cooperative Work*. PhD thesis. The University of Queensland.
-----. (2002). "The Locales Framework: making social thinking accessible for software practitioners?" in *Social Thinking, Software Practice*. (eds.) Dittrich, Y; Floyd, C & Klischewski, R. Cambridge: MIT Press.

Fodor, Jerry. (1975). *The Language of Thought.* New York: Thomas Crowell.

Forsyth, Richard & Rada, Roy. (1986). *Machine Learning: Applications in Expert Systems and Information Retrieval*. Chisester: Ellis Horwood Limited.

Foucault, Michel. (1970). *The Order of Things*. New York: Pantheon.
-----. (1977a). *Discipline and Punish.* New York: Pantheon.
-----. (1977b). "Nietzsche, Genealogy, History" in *Language, Counter-Memory, Practice: Selected Essays and Interviews.* Bouchard, B.F. (ed.). Ithaca: Cornell University Press.
-----. (1980). *Power/Knowledge: Selected Interviews and Other Writings, 1972-77*. Brighton: Harvester.
-----. (1988). "Technologies of the self". in *Technologies of the self: A seminar with Michel Foucault*, 16-49. (eds.) Martin, L.H.; Gutman, H. & Hutton, P.H. Tavistock Publications: London.

Frank, Arthur W. (1996). "From Dysappearance to Hyperappearance: Sliding Boundaries of Illness and Bodies" in *Theory Psychology* vol. 6. p. 733.

Frauenberger, Christopher. (2008). *Auditory Display Design: An Investigation of a Design Pattern Approach*. PhD thesis. London: City University.

Fuller, Matthew. (2003). *Behind the Blip: Essays on the Culture of Software.* New York: Autonomedia.
-----. (2005). *Media Ecologies: Materialist Energies in Art and Tecnoculture.* Cambridge: MIT Press.
-----. (2006). "Softness: Interrogability; General Intellect; Art Methodologies in Software" in *Skriftserie: Center for Digital Æstetik-forskning.* nr. 13. Aarhus: Center for Digital Æstetik-forskning.
-----. (2008). (ed.) *Software Studies: A Lexicon.* Cambridge: MIT Press.

Föllmer, Golo. (2001). *Soft Music*. <http://crossfade.walkerart.org/foellmer/text_print.html>

[accessed May 2008].

Gadamer, Hans-Georg. (1976). *Philosophical Hermeneutics.* Berkeley: University of California Press.
-----. (1989). *Truth and Method.* New York: Crossroad Press.

Gallagher, Shaun & Sørensen, Jesper Brøstad. (2006). "Experimenting with Phenomenology" in *Consciousness and Cognition.* vol. 15. pp. 119-134. Elsevier.
-----. (1986). "Lived Body and Environment" in *Research in Phenomenology* vol. 16. pp. 139–170.

Garfinkel, Harold. (1967). *Studies in Ethnomethodology*. Englewood Cliffs, NJ: Prentice-Hall.

Garnett, G., and C. Goudeseune. (1999). "Performance Factors in Control of High-Dimensional Spaces." In *Proceedings of the 1999 International Computer Music Conference*. San Francisco, International Computer Music Association, pp. 268 - 271.
-----. (2001). "The Aesthetics of Interactive Computer Music." *Computer Music Journal*, vol. 25 (1), pp. 22-33. MIT Press.

Gartland-Jones, Andrew; Copley, Peter. (2003). "The Suitability of Genetic Algorithms for Musical Composition" in *Contemporary Music Review*. vol. 22 (3).

Gaver, William W. (1991). "Technology Affordances" in *Proceedings of the ACM CHI 91 Human Factors in Computing Systems Conference*. New Orleans. p. 79-84.

Gärdenfors, Peter. (2000). *Conceptual Spaces: The Geometry of Thought.* Cambridge: The MIT Press.

Gehlen, Arnold. (1980). *Man in the Age of Technology* . New York: Columbia University Press. Original, Reinbek: Rowohlt.

Gerhard, David; Hepting, Daryl & McKague, Matthew. (2004). "Exploration of the Correspondence Between Visual and Acoustic Parameter Spaces" in *Proceedings of NIME 2004*. http://hct.ece.ubc.ca/nime/2004/NIME04/paper/index.html

Gere, Charlie. (2002). *Digital Culture.* London: Reaction Books.

Gentner, Don & Nielsen, Jacob. (1996). "The Anti-Mac interface" in *Communications of the ACM.* vol. 39 (8) pp. 70-82.

Gibson, James J. (1979). *The Ecological Approach to Visual Perception.* Houghton Mifflin.

Gibson, Will. (2006). "Material Culture and Embodied Action: Sociological Notes on the Examination of Musical Instruments in Jazz Improvisation" in *The Sociological Review.* vol. 54 (1).

Giddens, Antony. (1984). *The Constitution of Society: Outline of the Theory of Structuration*, Berkeley: University of California Press.

Gille, Bertrand. (1978). *Histoire des techniques.* Paris: Gallimard.

Godøj, Rolf. (2003). "Motor-mimetic Music Cognition" in *Leonardo.* vol 36 (4), pp. 317-319.
-----, Haga, E. & Jensenius, A. R. (2006). "Playing 'air instruments': Mimicry of sound-producing gestures by novices and experts" in *Gesture in Human-Computer Interaction and Simulation.* 3881 pp. 256-267.

Gohlke, Gerrit. (ed.) (2003). *Software Art - A Reportage about Source Code*. Berlin: The Media Arts Lab.

Goodwin, Andrew. (1988). "Sample and Hold: Pop Music in the Digital Age of Reproduction" in *Critical Quarterly* vol. 30. pp. 34-49.

Goriunova, Olga & Shulgin, Alexei. (2004). *Readme Edition 2004. Software Art and Cultures* Aarhus: Digital Aesthetics Research Centre.
-----. (2006). ed. *Readme 100 Temporary Software Art Factory*. Dortmund: Hartware MedienKunstVerein
----- & Shulgin, Alexei. (2008). "Glitch" in *Software Studies: A Lexicon.* (ed.) Fuller, Matthew. Cambridge: MIT Press.

Gould, Stephen Jay. (1989). *Wonderful Life: The Burgess Shale and the Nature of History*. W. W. Norton & Co.

Greenbaum, J. and Kyng, M. (1991). *Design at Work: Cooperative Design of Computer Systems*. New Jersey: Laurence Erlbaum Associates.

Greenberg, Saul & Buxton, Bill. (2008). "Usability Evaluation Considered Harmful (Some of the Time)" in *CHI 2008 Proceedings*. Florence, Italy: ACM.

Grint, Keith & Woolgar, Steve. (1997). *The Machine at Work*. Cambridge: Polity.

Guattari, Felix. (1993). "Machinic Heterogenesis" in *Rethinking Technologies.* Conley, V. A. (ed.). Minneapolis: University of Minnesota Press.

Habermas, Jürgen (1970). "Technology and Science as 'Ideology'," in *Toward a Rational Society.* Boston: Beacon Press.
-----. (1984). *Theory of Communicative Action*. Boston: Beacon Press.

Halasz, Frank & Moran, Thomas. P. (1982). "Analogy considered harmful" in *Proceedings of the ACM Conference on Human Factors in Computer Systems (Chi 1982)*, pp. 383-386. New York, NY: ACM.

Hamilton, Edith & Cairns, Huntington (eds.) (1989). *The Collected Dialogues of Plato Including the Letters.* New Jersey: Princeton University Press.

Hamman, Michael. (1999). "From symbol to semiotic: representation, signification, and the composition of music interaction," Journal of New Music Research, 28(2), pp.90-104.
-----. (2000). "From Technical to Technological: Interpreting Technology Through Composition," Proceedings of the 2000 Colloquium on Musical Informatics, L'Aquila, ITALY.
-----. (2002). "From Technical to Technological: The Imperative of Technology in Experimental Music Composition," Perspectives of New Music, Vol. 40(1), Winter 2002.

Hansen, Mark. (2000). *Embodying Technesis: Technology Beyond Writing.* Ann Arbor: The University of Michigan Press.
-----. (2006). *Bodies in Code: Interfaces with Digital Media.* New York: Routledge.

Haraway, Donna. (1991). "A cyborg manifesto: Science, technology and socialist feminism in the late twentieth century" in *Symians, Cyborgs & Women: The Re-Invention of Nature.* New York: Routledge.
-----. (1997). *Modest_Witness@Second_Millenium: .FemaleMan(c)_Meets_OncoMouse(TM).* New York: Routledge.

Harkleroad, Leon. (2006). *The Math behind the Music.* New York: Cambridge University Press.

Haugeland, John. (1998). "Mind Embodied and Embedded" in *Having Thought.* (ed.) Haugeland. J. Cambridge, MA: MIT Press.

Hayles, N. Katherine (1997) "The Condition of Virtuality," in Jeffrey Masten, Peter Stalleybrass, Nancy Vickers, (eds.) *Language Machines: Technologies of Literary and Cultural Production*. London: Routledge.
-----. (1999). *How We Became Posthuman*. Chicago: University of Chicago Press.
-----. (2005). *My Mother was a Computer: Digital Subjects and Literary Texts*. Chicago: University of Chicago Press.

Heidegger, Martin. (1962). *Being and Time.* Oxford: Blackwell Publishers.
-----. (1977). *The Question Concerning Technology and Other Essays*. New York: Harper and Row.

Hitch, G. (1978). "The Role of short-term Memory in Mental Arithmetic" in *Cognitive Psychology*, vol 10. pp. 302-323.

Hopkin, Bart. (1996). *Musical Instrument Design: Practical Information for Instrument Making.* Tuscon, AZ: See Sharp Press.

Horkheimer, Max. (1972). *Critical Theory: Selected Essays*. New York: Herder and Herder.

Horning, Susan Schmidt. (2004). "Engineering the Performance: Recording Engineers, Tacit Knowledge and the Art of Controlling Sound" in *Social Studies of Science.* vol. 34(5). p. 703.

Horwood, Wally. (1983). *Adolphe Sax 1814-1894: His Life and Legacy.* Baldock: Egon Publishers.

Hoskinson, Reynald; Doel, Kees van den & Fels, Sidney. (2003). "Real-time Adaptive Control of Modal Synthesis" in *Proceedings of NIME 2003*. http://hct.ece.ubc.ca/nime/2003/onlineproceedings/TOC.html [accessed May, 2005].

Howard, David M. & Rimell, Stuart. (2004). "Real-Time Gesture-Controlled Physical Modelling Music Synthesis with Tactile Feedback" in *EURASIP Journal on Applied Signal Processing*. Volume 2004 (7). pp. 1001-1006.

Huhtamo, Erkki. (1996). "Time Traveling in the Gallery: An Archeological Approach in Media Art" pp. 233-268. *Immersed in Technology: Art and Virtual Environments*. (ed.) Moser, Mary Anne. Cambridge: The MIT Press.

Hughes, Thomas P. (1986). "The Seamless Web: Technology, Science, Etcetera, Etcetera" in *Social Studies of Science.* Vol 16. p. 281. London: Sage Publications.
-----. (1994). "Technological Momentum" in *Does Technology Drive History?: The Dilemma of Technological Determinism.* (eds.) Smith, M.R & Marx, Leo. Cambridge: MIT Press.

Hugill, Andrew. (2008). *The Digital Musician.* London: Routledge.

Hunt, Andy and Kirk, Ross. (1999). Radical User Interfaces for Real-Time Control. In *Euromicro 99 Conference*. Milan.
-----, Wanderley, Marcelo & Kirk, Ross. (2000). "Towards a Model of Mapping Strategies for Instrumental Performance". In *Proc. of the 2000 International Computer Music Conference*. San Francisco, Calif.: International Computer Music Association, pp 209–212.

----- & R. Kirk. 2000. "Mapping Strategies for Musical Performance." In M. Wanderley and M. Battier, (eds.) *Trends in Gestural Control of Music*. Paris: Ircam - Centre Pompidou.
----, Wanderley, Marcelo M. & Kirk, Ross. (2000). "Towards a Model for Instrumental Mapping in Expert Musical Interaction", in *Proceedings of International Computer Music Conference*.
-----, Wanderley, Marcelo M. & Paradis, Matthew. (2002). "The Importance of Parameter Mapping in Electronic Instrument Design" in *NIME 2002 Proceedings*. Limerick: University of Limerick, Department of Computer Science and Information Systems.
----, Kirk, Ross. (2003). "MidiGrid: Past, Present and Future" in *Proceedings of NIME 2003*. http://hct.ece.ubc.ca/nime/2003 onlineproceedings/TOC.html [accessed May, 2005].

Hurley, S. (1998). *Consciousness in Action.* Cambridge: Harvard University Press.

Husserl, Edmund. (1970). *The Crisis in European Sciences and Transcendental Phenomenology.* Evanston: Northwestern University Press.

Hutchby, Ian. (2001). "Technologies, Texts and Affordances" in *Sociology*. vol. 35, p. 441.
-----. (2000). *Conversation and Technology*. Cambridge: Polity.

Hutchins, E. (1995). *Cognition in the Wild*. Cambridge, MA: MIT Press.
-----. (1995). "How a cockpit remembers its speeds". *Cognitive Science*, 19, 265-288.

Huyssen, Andreas. (1986). *After the Great Divide: Modernism, Mass Culture, Postmodernism*. Bloomington: Indiana University Press.

Iazzetta, Fernando. (1996). "Formalization of Computer Music Interaction Through a Semiotic Approach " in *Journal of New Music Research*. vol. 25. pp.212-230.

Ihde, Don. (1979). *Technics and Praxis.* Dortrech, Holland: D. Reidel Publishing Company.
-----. (1983). *Existential Technics.* Albany. State University of New York Press.
-----. (1990). *Technology and the Lifeworld: From Garden to Earth*. Bloomington: Indiana University Press.
-----. (2002). *Bodies in Technology*. University of Minnesota Press, Minneapolis.
-----. (2003). "Introduction: Part One", Ihde D. & Selinger E. (eds.), *Chasing Technoscience: Matrix for Materiality.* Bloomington: Indiana University Press.

International Standard Organisation. (1998). *ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11 : Guidance on usability*. Geneve: International Standard Organisation.

Ishii, Hiroshi & Ullmer, Brygg. (1997). “Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms” in *Proceedings of CHI 97 Conference on Human Factors in Computing systems*, Atlanta, Georgia USA.

Jacko, Julie A & Sears, Andrew. (eds). (2003). *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*. Lawrence Erlbaum Associates, Publishers, London.

Jacob, Bruce L. (1996). “Algorithmic composition as a model of creativity” in *Organised Sound* 1(3), pp. 157-65. Cambridge University Press.

Jacob, Robert J. K; Sibert, Linda E; McFarlane, Daniel C & Mullen, M. Preston Jr. (1994). "Integrality and Separability of Input Devices" in *ACM Transactions on Computer-Human Interaction*, vol. 1 (1).

Jakobson, Roman. (1960). “Closing statement: linguistics and poetics” in *Style in Language* (ed.) Sebeok. Cambridge: MIT Press.
----- & Halle, Morris. (2002). *Fundamentals of Language.* New York: Walter de Gruyter.

Jameson, Frederic. (1991). *Postmodernism, or, the Cultural Logic of Late Capitalism*. New York: Duke University Press.

Jaromil (2007) dynebolic.org [accessed November, 2007]

Jensenius, Alexander R. (2007). *Action-Sound: Developing Methods and Tools to Study Music-Related Body Movement.* PhD Thesis. Oslo: University of Oslo.

Johnson-Laird, Philip N (1983). *Mental Models: Toward a Cognitive Science of Language, Inference and Consciousness*. Cambridge: Harvard University Press.

Johnston, John. (2002). "A Future For Autonomous Agents: Machinic *Merkwelten* and Artificial Evolution" in *Configurations.* vol. 10.
-----. (2008). *The Allure of Machinic Life: Cybernetics, Artificial Life and the New AI.* Cambridge, MA: MIT Press.

Jordan, Patrick W. (2000). *Designing Pleasurable Products.* London: Taylor and Francis.

Jordà, Sergi. (1999). “Faust Music On Line (FMOL): An Approach to Real-time Collective Composition on the Internet” in *Leonardo Music Journal*. Vol. 9, pp. 5-12.
-----. (2001). *New Musical Interfaces and New Music-Making Paradigms*. Webarticle: http://www.iua.upf.es/~sergi/articles/NIME2001.pdf
-----. (2004). "Digital Instruments and Players: Part I - Efficiency and Apprenticeship" in *Proceedings of NIME 2004*. http://hct.ece.ubc.ca/nime/2004/NIME04/paper/index.html (May, 2005).
-----. (2005). "Multi-user Instruments: Models Examples Promises" in *Proceedings of NIME 2005*. (eds.) Fels, Sidney & Blaine, Tina. University of British Columbia. Vancouver.
-----. (2005). *Digital Lutherie: Crafting musical computers for new musics' performance and improvisation*. PhD Thesis, University of Pompeu Fabra, Barcelona.
-----. (2008). "Musical Interfaces" - presentation at the *International Music Computing Workshop 2008* in Open University, Milton Keynes, UK. <http://musiccomp2008.open.ac.uk/>

Juslin, Patrick N. & Sloboda, John A. (2001). *Music and Emotion: Theory and Research.* Oxford: Oxford University Press.

Jönsson, Bodil; Malmborg, Lone & Svensk, Arne. (2004). "Situated Research and Design for Everyday Life". Certec Report 2:2004, Lund University.

Kahn, Douglas. (1995) "Track Organology" in *Critical Issues in Electronic Media* (ed.) Penny, Simon. Albany: SUNY Press.
-----. (1999). *Noise, Water, Meat: A History of Sound in the Arts.* MIT Press, Cambridge.

Kaltenbrunner, Martin; Geiger, Günter & Jordá, Sergi. (2004). "Dynamic Patches for Live Musical Performance" in *Proceedings of NIME 2004*. http://hct.ece.ubc.ca/nime/2004/NIME04/paper/index.html (May, 2005).
-----. & Bencina, R. (2007). "reacTIVision: A Computer-Vision Framework for Table-Based Tangible Interaction", *Proceedings of the first international conference on "Tangible and Embedded Interaction" (TEI07)*. Louisiana: Baton Rouge.

Kaplan, David. M. (2004). (ed). *Readings in the Philosophy of Technology.* Lanham, Maryland: Roman & Littlefield Publishers.

Kapur, Ajay; Lazier, Ariel J, Davidson, Philip; Wilson, R. Scott & Cook, Perry. (2004). "The Electronic Sitar Controller" in *Proceedings of NIME 2004*. http://hct.ece.ubc.ca/nime/2004/NIME04/paper/index.html (May, 2005).

Kartomi, Margaret J. (1990). *On Concepts and Classifications of Musical Instruments.* Chicago: University of Chicago Press.

Kay, Alan & Goldberg, Adele. (1977). "Personal Dynamic Media" in *Computer* vol. 10(3)pp. 31–41.

Kensing, Finn. (2003). *Methods and Practices in Participatory Design*. Copenhagen: ITU Press.

Kiczales, Gregor. (1996). "Beyond the Black Box: Open Implementation" in *IEEE Software*, January, 8–11.

Kirakowski, J and Corbett, M, (1988). "Measuring User Satisfaction" in *People and Computers IV*. D M Jones and R Winder (Eds.). Cambridge: Cambridge University Press.

Kirsh, David. & Maglio, Paul. (1994). "On distinguishing epistemic from pragmatic action." *Cognitive Science*. vol 18. pp. 513-49.

Kittler, Friedrich, A. (1995). "There is No Software" in *CTheory.net.* <http://www.ctheory.net/articles.aspx?id=74> [accessed December 2008].
-----. (1999) *Gramophone, Film, Typewriter.* Stanford: Stanford University Press.

Kline, Ronald. (2003). "Resisting Consumer Technology in Rural America: The Telephone and Electrification" in *How Users Matter: The Co-construction of Users and Technology.* Oudshoorn, N & Pinch, T. (eds.) Cambridge: MIT Press.

Knappett, Carl. (2005). *Thinking Through Material Culture: An Interdisciplinary Perspective.* Philadelphia: University of Pennsylvania Press.

Kolb, David. (1983). *Experiential Learning: Experience as the Source of Learning and Development*. New Jersey: Prentice Hall.

Kramer, G. 1996. "Mapping a single data stream to multiple auditory variables: A subjective approach to creating a compelling design." In *Proceedings of International Conference on Auditory Display* - ICAD'96.

Kranzberg, Melvin. (1986). "Technology and History: Kranzberg's Laws" in *Technology and Culture.* vol. 27 (3) pp. 554-560. Baltimore: John Hopkins University Press.

Kroker, Arthur & Kroker, Marilouise. (1987). *Body Invaders: Panic Sex in America.* New York: St. Martin's Press.

Kuhn, Thomas .S. (1962). *The Structure of Scientific Revolutions*. Chicago: University of Chicago Press.

Kurzweil, Ray. (1999). *The Age of Spiritual Machines*. London: Phoenix.

Kvifte, Tellef. (1989). *Instruments and the Electronic Age: Towards a Terminology for a Unified Description of a Playing Technique*. online here: <http://www.hf.uio.no/imv/om-instituttet/nfs/iaea/instrume.html> [accessed June 2005].

LaBelle, Brandon & Steve Roden. (1999). *Site of Sound: of Architecture and the Ear*. Los Angeles: Errant Bodies Press.

Lakoff, George & Johnson, Mark. (1999). *Philosophy in the Flesh: The Embodied Mind and its Challenge to Western Thought.* New York: Basic Books.
-----. (1980). *Metaphors We Live By.* Chicago: The University of Chicago Press.

Langton, Christopher (ed.) (1989). *Artificial Life*. Redwood City: Addison-Wesley.
-----. (1997). *Artificial Life: An Overview*. Cambridge, MIT Press.

Latour, Bruno. (1987). *Science in Action: How to Follow Scientists and Engineers Through Society.* Cambridge: Harvard University Press.
-----. (1992). "Where are the Missing Masses? The Sociology of a Few Mundane Artifacts" in *Shaping Technology/Building Society: Studies in Sociotechnical Change.* (eds.) Bijker, W. & Law, J. Cambridge, MA: MIT Press.
-----. (1993). *We Have Never Been Modern.* Cambridge: Harvard University Press.
-----. (1994). "On Technical Mediation - Philosophy, Sociology, Genealogy" in *Common Knowledge,* 3:2. 29-64.
-----. (1999). *Pandora's Hope. Essays on the Reality of Science Studies*. Cambridge, MA: Harvard University Press.
-----. (2000). "The Berlin Key or How To Do Words with Things" in *Matter, Materiality and Modern Culture.* Graves-Brown, P. M. (ed.). London: Routledge. pp. 10-21.
-----. (2005). *Reassembling the Social: An Introduction to Actor-Network Theory.* Oxford: Oxford University Press.

Laurel, Brenda. (ed.) (1990). *The Art of Human-Computer Interface Design*, Reading, MA: Addison-Wesley.

Law, John. (1992). "Notes on the Theory of the Actor Network: Ordering, Strategy and Heterogeneity" published by the Centre for Science Studies, Lancaster University, Lancaster <http://www.comp.lancs.ac.uk/sociology/papers/Law-Notes-on-ANT.pdf> [accessed December 2007].
----- & Hassard, John. (1999). *Actor Network Theory and After*. Oxford: Blackwell and Sociological Review.
----- & Bijker, Wiebe. (1992). *Shaping Technology/Building Society*. Cambridge, MA: MIT Press.

Leder, Drew. (1990). *The Absent Body*. Chicago: University of Chicago Press.

Lefebvre, Henri. (1991). *The Production of Space*. Oxford: Basil Blackwell.

Leman, Marc. (2008). *Embodied Music Cognition and Mediated Technology.* Massachusetts: MIT Press.

Lerdal, Fred & Jackendoff, Ray. (1985). *A Generative Theory of Tonal Music*. Cambridge: The MIT Press.

Levin, Golan. (2000) "Painterly Interfaces for Audiovisual Performance". *MSc. Thesis*, MIT Media Laboratory.
-----. (2005a). "A Personal Chronology of Audiovisual Systems Research." *Proceedings of NIME '05*, Vancouver, BC, Canada. May 26-28.
----- & Lieberman, Zac. (2005b). "Sounds from Shapes: Audiovisual Performance with Hand Silhouette Contours in The Manual Input Sessions". *Proceedings of NIME '05*, Vancouver, BC, Canada.
-----. (2006). "The Table is The Score: An Augmented-Reality Interface for Real-Time,

Tangible, Spectrographic Performance." *Proceedings of the International Conference on Computer Music 2006 (ICMC'06)*. New Orleans.

Levinson, Paul. (1997). *Soft Edge: A Natural History and Future of the Information Revolution.* London: Routledge.

Levitin, Daniel J; McAdams, Stephen & Adams, Robert L. (2002). "Control Parameters for Musical Instruments: A Foundation for New Mappings of Gesture to Sound" in *Organised Sound*, 7(2), pp. 171-189.

Lewis, George. (2000). "Too Many Notes: Computers, Complexity and Culture in *Voyager*" in *Leonardo Music Journal.* vol 10. pp. 33-39.

Lewis, J. R. (1995) "IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use" in *International Journal of Human-Computer Interaction,* vol. 7 (1), pp. 57-78.

Libin, Laurence. (2000). "Progress, Adaption, and the Evolution of Musical Instruments" in *Journal of the American Musical Instrument Society.* vol. 26. pp. 187-213.

Licklider, J.C.R. (1960). "Man-Computer Symbiosis" in *IRE Transactions on Human Factors in Electronics.* vol HFE-1, March 1960, 4-11.

Liddle, David. (1996). "Design of the Conceptual Model" in *Bringing Design to Software.* Winograd, Terry (ed.) New York: ACM Press.

Liley, Thomas. (1998). "Invention and Development" in *The Cambridge Companion to the Saxophone*. (ed.) R. Ingham. Cambridge: Cambridge University Press.

Lin, Angela & Cornford, Tony. (2000). "Framing Implementation Management" in *Proceedings of the twenty first international conference on Information systems*, p.197-205, Brisbane, Australia.

Lin, Han X. Choong, Yee-Yin. & Salvendy, Gavriel. (1997). "A Proposed Index of Usability: A Method for Comparing the Relative Usability of Different Software Systems" in *Behaviour & Information Technology*. vol. 16(4/5), pp. 267-278.

Lindgaard, G. (1994). *Usability Testing and System Evaluation: A Guide for Designing Useful Computer Systems*, Chapman and Hall, London, U.K.

Link, Stan. (2001). “The Work of Reproduction in the Mechanical Aging of an Art: Listening to Noise” in *Computer Music Journal*, vol 25 (1), pp. 34-47. Massachusetts Institute of Technology.

Livingston, H. (2000). "Paradigms for the new string instrument: digital and materials technology" in *Organised Sound*, vol. 5 (3). pp. 135-147.

Lopez de Mantaras, Ramon & Llluis Arcos, Josep. (2002). "AI and Music: From Composition to Expressive Performance" in *AI Magazine*, Fall 2002. American Association for Artificial Intelligence.

Lopez-Lezcano, Fernando. (2005) "Surviving on Planet CCRMA, Two Years Later and Still Alive" in *Proceedings of the 2005 Linux Audio Conference*. Karlsruhe: ZKM.

Lovejoy, Margot. (1997). *Postmodern Currents: Art and Artists in the Age of Electronic Media*. (2nd. ed.) New Jersey: Prentice Hall.

Lövgren, Jonas. (1993). *Human-Computer Interaction: What Every System Developer Should Know.* Lund: Studentlitteratur.

Lyons, Michael J; Haehnel, Michael & Tetsutani, Nobuji. (2003). "Designing, Playing and Performing with a Vision-based Mouth Interface" in *Proceedings of NIME 2003*. http://hct.ece.ubc.ca/nime/2003/onlineproceedings/TOC.html (May, 2005).

Machover, Tod & Chung, J. (1989). "Hyperinstruments: Musically intelligent/interactive performance and creativity systems" in *Proceedings of ICMC 1989.* San Francisco: ICMA.

Mackay, H; Creacen, C; Beynon-Davies, P & Tudhope, D. (2000). "Reconfiguring the User: Using Rapid Application Development" in *Social Studies of Science.* vol. 30, no. 5; pp. 737-759.

Mackenzie, Adrian. (2002). *Transductions: Bodies and Machines at Speed.* London: Continuum.
-----. (2006). *Cutting Code: Software and Sociality.* New York: Peter Lang.

MacKenzie, Donald & Wajcman, Judy. (eds.) (1984). *The Social Shaping of Technology: How the Refridgerator got its Hum*. Milton Keynes: Open University Press.
-----. (1996). *Knowing Machines: Essays on Technical Change.* Cambridge: MIT Press.

Maeda, John. (1999). Design by Numbers. Cambridge: The MIT Press.
-----. (2000). *Maeda @ Maeda*. London: Thames and Hudson, 2000.

Magnusson, Thor. (2005). "ixi software: The Interface as Instrument" in *Proceedings of NIME 2005*. Vancouver: University of British Columbia.
-----. (2005). "ixi software: Open Controllers for Open Source Software" in *Proceedings of ICMC 2005*. Barcelona: University of Pompeu Fabra.
-----. (2006). "Screen-Based Musical Instruments as Semiotic Machines" in *Proceedings of NIME 2006*. Paris: IRCAM.
-----. "Affordances and Constraints in Screen-Based Musical Instruments" in proceedings of the NordiCHI conference, Oslo: Oslo University, 2006.
-----. (2006). "The ixi instruments as Semiotic Machines" in Proceedings of ICMC 2006, New Orleans: Tulane University.
-----. & Hurtado Mendieta, Enrike. (2007). "The Acoustic, the Digital and the Body: A Survey on Musical Instruments", Proceedings of the NIME Conference, New York, USA.
-----. (2007). "The ixiQuarks: Merging Code and GUI in One Creative Space" in Immersed Music: The ICMC 2007 Conference Proceedings. Copenhagen: Re:New.
----- & Magnusson, Runar. (2007). "SameSameButDifferent v.02 - Iceland" in *YLEM Journal.* <http://www.ixi-audio.net/thor/SSBD_v02-Iceland.pdf> [accessed March 2008].
-----. (2009). "On Epistemic Tools: Acoustic and Digital Instruments as Cognitive Scaffoldings" in *Organised Sound*. Vol 14 (2). Cambridge: Cambridge University Press.

Mallery, John C; Hurwitz, Roger & Duffy, Gavan. (1986). *Hermeneutics: From Textual Explication to Computer Understanding?* A.I. Memo No. 871. MIT.

Manovich, Lev. (2001). *The Language of New Media*. Cambridge: The MIT Press.
-----; (2008). *Software Takes Command.* Software Studies Initiative. <http://lab.softwarestudies.com/2008/11/softbook.html> [accessed March 2009]

Mansoux, Aymeric & de Valk, Marloes. (2008). *FLOSS + Art.* Poitiers: Goto10.

-----; Galanopoulos, Antonios & Lee, Chun. (2007). “pure:dyne” in *Proceedings of the 2007 Pd Convention*. Montreal.

Marcuse, Herbert. (1964). *One-Dimensional Man*. Boston: Beacon.

Markham, Arthur B. (1999). *Knowledge Representation*. Mahwah NJ: Lawrence Erlbaum Associates.

Marshall, Mark T. and Wanderley, Marcelo M. (2006). "Vibrotactile Feedback in Digital Musical Instruments." in *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression (NIME06)*, Paris: IRCAM, pp. 226-229.

Marx, Karl. (1887). *Capital: The Process of Production of Capital.* <http://www.marxists.org/archive/marx/works/1867-c1/index.htm> [accessed August 2008].
-----. (1964). *Selected Writings in Sociology and Social Political Philosophy,* (ed.) Bottomore, T & Rubel, M. Maidenhead, UK: McGraw-Hill.
-----. (1971). *The Poverty of Philosophy,* New York: International.

Mateas, Michael. (2001). “Expressive AI: A Hybrid Art and Science Practise” in *Leonardo*, vol. 34 (2).

Matthews, Robert J. (1994). "Psychological Reality of Grammars" in *Noam Chomsky: Critical Assessments.* Otero, Carlos P. (ed). New York: Routledge.

Maturana, Humberto & Varela, Francisco (1980). *Autopoiesis and Cognition: The Realization of the Living.* Dordecht: D. Reidel Publishing Co.
----- & Varela, Francisco. (1987). *The Tree of Knowledge: The Biological Roots of Human Understanding.* Boston: Shambhala Publications.

Maybury, Mark T & Wahlster, Wolfgang. (1998). *Readings in Intelligent User Interfaces.* Morgan Kaufmann Publishers, San Francisco.

Mayer, Leonard, B. (1994). *Music, the Arts, and Ideas: Patterns and Predictions in Twentieth-Century Culture.* Chicago: University of Chicago Press.

Mayer, Paul A. (1999). *Computer Media and Communication: A Reader*. Oxford: Oxford University Press.

McCarthy, John & Wright, Peter. (2004). *Technology as Experience.* Cambridge: MIT Press.

McCartney, James. (2002). "Rethinking the Computer Music Language: SuperCollider" in *Computer Music Journal,* 26:4, pp. 61–68.
-----. (2003). "A Few Quick Notes on Opportunities and Pitfalls of the Application of Computers in Art and Music" in *Ars Electronica, 2003*. Ars Electronica.

McClelland, J.L, Rumelhart, D.E. and the PDP Research Group (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Volume 1 and 2: Foundations, Cambridge, MA: MIT Press

McCullough, Malcolm. (1996). *Abstracting Craft: The Practiced Digital Hand.* Cambridge: MIT Press.
-----. (2004). *Digital Ground: Architecture, Pervasive Computing, and Environmental Knowing.* Cambridge: MIT Press.

McLean, Alex. (2004). "Hacking Perl in Nightclubs" in *perl.com* August 31, 2004.

<http://www.perl.com/pub/a/2004/08/31/livecode.html> [accessed July 2007]

McLuhan, Marshall. (1964). *Understanding Media: The Extensions of Man*. Cambridge, MA: The MIT Press.
----- & Watson, Wilfred. (1970). *From Cliche to Archetype*. New York: Viking Press.

Merleau-Pony, Maurice. (2002). *The Phenomenology of Perception.* London: Routledge.
-----. (1962). *The Phenomenology of Perception.* New York: The Humanities Press.

Metzinger, Thomas & Gallese, Vittorio. (2003). "The Emergence for a Shared Action Ontology: Building Blocks for a Theory" in *Consciousness and Cognition*. vol. 12. pp. 549-471.

Meyer, Leonard B. (1994). *Music, the Arts, and Ideas: Patterns and Predictions in Twentieth-century Culture*. Chicago: University of Chicago Press.

Michelfelder, Diane P. & Palmer, Richard E. (1989). *Dialogue and Deconstruction: The Gadamer-Derrida Encounter*. Albany: SUNY Press.

Microsoft Corporation. (1992). *The Windows Interface: An Application Design Guide*. Seattle: Microsoft Press.

Miranda, Eduardo R. & Wanderley, Marcelo M. (2006) *New Digital Musical Instruments: Control and Interaction beyond the Keyboard*, A-R Editions.

Mitcham, Carl. (1994). *Thinking Through Technology: The Path Between Engineering and Philosophy.* Chicago: University of Chicago Press.

Mithen, Steven. (1998). "A Creative Explosion? Theory of the Mind, Language and The Disembodied Mind of the Upper Palaeolithic" in *Creativity in Human Evolution and Prehistory*. (ed.) Steve Mithen. New York: Routledge.
-----. (2000). "Mind, Brain and Material Culture: An Archaeological Perspective" in *Evolution and the Human Mind: Modularity, Language and Metacognition*. (eds.) P.Carruthers and A. Chamberlain. Cambridge: Cambridge University Press.

Modler, Paul. (2000). "Neural Networks for Mapping Gestures to Sound Synthesis" in *Trends in Gestural Control of Music*. Paris: Ircam - Centre Pompidou.
-----, Myatt, Tony & Saup, Michael. (2003). "An Experimental Set of Hand Gestures for Expressive Control of Musical Parameters in Realtime" in *Proceedings of NIME 2003*. http://hct.ece.ubc.ca/nime/2003/onlineproceedings/TOC.html (May, 2005).

Molnar-Szakacs, Istvan & Overy, Katie (2006). "Music and Mirror Neurons: From Motion to 'E'motion" in *Social Cognitive and Affective Neuroscience.* vol. 3 (2). Oxford: Oxford Journals.

Momeni, Ali & Wessel, David. (2003). "Characterizing and Controlling Musical Material Intuitively with Geometric Models" in *Proceedings of NIME 2003*.
http://hct.ece.ubc.ca/nime/2003/onlineproceedings/TOC.html [accessed May 2005].
-----. (2005). "Composing instruments: Inventing and performing with generative computer-based instruments". PhD Dissertation. UC Berkeley.

Moog, Bob. (2002). "Foreword" in *Analog Days: The Invention and Impact of the Moog Synthesizer.* Pinch, T & Trocco, F. Cambridge, MA: Harvard University Press.

Moore, F. Richard. (1990). *Elements of Computer Music*. New Jersey : Prentice Hall.

Moradi, Iman. (2009). *Glitch: Designing Imperfection.* New York: Mark Batty Publisher.

Morales-Manzanares, Roberto; Eduardo F Morales; Roger Dannenberg; and Jonathan Berger. (2001). “SICIB: An Interactive Music Composition System Using Body Movements” in *Computer Music Journal*. vol. 25 (2), pp. 25-36. MIT Press.

Moorefield, Virgil. (2005). *The Producer as a Composer: Shaping the Sounds of Popular Music.* Massachusetts, MIT Press.

Mumford, Lewis. (1934). *Technics and Civilisation.* New York: Hartcourt Brace.
-----. (1952). *Art and Technics.* New York: Columbia University Press.

Nadin, Mihai. (1988). "Interface Design: A Semiotic Paradigm" in *Semiotica*. vol. 69 (3/4). pp. 369-302.

Nardi, Bonnie A. & Zarmer, Craig L. (1990). "Beyond Models and Metaphors: Visual Formalisms in User Interface Design". Hewlett Packard Tech Report: HPL-90-149.

Nagashima, Yoichi. (2004). "Measurement of Latency in Interactive Multimedia Art" in *Proceedings of NIME 2004*. <http://hct.ece.ubc.ca/nime/2004/NIME04/paper/index.html> [accessed May, 2005].

Nake, Frieder. (1994). "Human-computer interaction: signs and signals interfacing" in *Languages of Design 2* pp. 193-205.

Nettl, Bruno & Russell, Melinda. (1999). *In the Course of Performance: Studies in the World of Musical Improvisation.* Chicago: University of Chicago Press.

Newton-Dunn, Henry; Nakano, Hiroaki & Gibson, James. (2002). “Block Jam”, in Emergent Technologies of SIGGRAPH 2002, p.67.
-----; Nakano, Hiroaki & Gibson, James. (2003). "Block Jam: A Tangible Interface for Interactive Music" in *Proceedings of NIME 2003*. http://hct.ece.ubc.ca/nime/2003/onlineproceedings/TOC.html [accessed May 2005].

Nielsen, Jakob. (1993) *Usability Engineering*. Boston: Academic Press. (Chapter 5), p. 115.

Nietzsche, Friedrich. (2004). *Beyond Good and Evil: Prelude to the Philosophy of the Future.* Cambridge: Cambridge University Press.

Nilson, Click. (2007). "Live Coding Practice". in *Proceedings of NIME 2007*.

Noe, Alva. (2004). *Action in Perception*. Cambridge, MA: The MIT Press.

Norman, Donald A. (1999). "Affordances, Conventions, and Design". in Interactions, 6 (3). p. 38-41.
-----. (1998). *The Design of Everyday Things*. London: MIT Press.
-----. (1988). *The Psychology of Everyday Things.* New York: Basic Books.
-----. (1993). *Things that make us smart: defending human attributes in the age of the machine.* Reading: Addison-Wesley Publishing Company.
-----. (2000). "Making Technology Invisible: A Conversation with Donal Norman" in *Information Appliances*. (ed.) Bergman, E. San Francisco: Morgan Kaufmann.

Norman, Katherine. (2004). *Sounding Art: Eight Literary Excursions through Electronic Music*. Ashgate, Aldershot, UK.

Nye, David. E. (2006). *Technology Matters: Questions to Live With.* Cambridge: MIT Press.

Nyman, Michael. (1974). *Experimental Music: Cage and Beyond*. Cambridge: Cambridge University Press.

O'Doherty. E. F. (1961). “Report on Engineering Design” in *Journal of Engineering Education* 51. No 8.

Oldenziel, Ruth. (2006). "Signifying Semantics for a History of Technology" in *Technology and Culture.* Baltimor: John Hopkins University Press.

O'Modhrain, Sile (2000). *Playing by Feel: Incorporating Haptic Feedback into Computer-Based musical Instruments*. PhD Dissertation. Stanford University.
------ & Essl, Georg. (2004). "PebbleBox and CrumbleBag: Tactile Interfaces for Granular Synthesis" in *Proceedings of NIME 2004*.
----- & Essl, Georg. (2005). "Enaction in the Context of Musical Performance" in *Interdisciplines Conference Proceedings.* http://www.interdisciplines.org/enaction/
-----., (2004). "Touch and go - designing haptic feedback for a hand-held mobile device" in *BT Technology Journal*, Vol 22 (4). Kluwer Academic Publishers.
-----., (2006) "Movement and music: designing gestural interfaces for computer-based musical instruments". Keynote address: in *Proceedings of the 8th international conference on Multimodal interfaces.* CA:Banff.

O'Reilly, Tim. (2005). "What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software" <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html> [accessed March 2009].

Orio, Nicola; Schnell, Norbert & Wanderley, Marcelo M. (2001). "Input Devices for Musical Expression: Borrowing Tools from HCI" in *Proceedings of NIME 2001*. http://hct.ece.ubc.ca/nime/2001/program.html (May, 2005).

Oudshoorn, Nelly & Pinch, Trevor (eds). (2003). *How Users Matter: The Co-Construction of Users and Technologies.* Cambridge: MIT Press.

Pacey, Arnold. (1999). *Meaning in Technology.* Cambridge: MIT Press.

Paradiso, Joseph A. (1997). "New ways to play: Electronic music interfaces". *IEEE Spectrum* 34, 12. pp. 18-30.
-----. (1999). "The Brain Opera Technology: New Instruments and Gestural Sensors for Musical Interaction and Performance" in *Journal of New Music Research.* vol. 28 (2). pp. 130-149.
-----. (2003). "Dual-Use Technologies for Electronic Music Controllers: A Personal Perspective" in *Proceedings of NIME 2003*. http://hct.ece.ubc.ca/nime/2003/onlineproceedings/TOC.html (May, 2005).

Parnas, David L. & Clements, Paul C. (1986). "A Rational Design Process: How and Why to Fake It" in *IEEE Transactions in Software Engineering.* vol. 12 (2).

Paiva, Ana. (2000). *Affective Interactions: Towards a New Generation of Computer Interfaces*. London: Springer.

Pearce, Marcus & Wiggins, Geraint. (2002). "Aspects of a Cognitive Theory of Creativity in Musical Composition" *ECAI 2002 Workshop on Creative Systems,* Lyon.

Peirce, Charles S. (1955). *The Philosophical Writings of Peirce*. New York: Dover.
-----. (1958). *Collected Papers*. Cambridge: Harvard University Press.

Pelletier, Jean-Marc. (2005). "A Graphical Interface for Intuitive Signal Routing" in *Proceedings of NIME 2005*. (eds.) Fels, Sidney & Blaine, Tina. Vancouver: University of British Columbia.

Pecquet, Frank. (1999). “From the Model to its algorithmic application” in *Organised Sound* vol. 4 (2), pp. 73-78. Cambridge University Press.

Perlman, G. (1997) *Practical Usability Evaluation*. <http://sigchi.org/chi97/proceedings/tutorial/gp.htm> [accessed June 2005].

Perlman, Marc. (2004). "Golden Ears and Meter Readers: The Contest for Epistemic Authority in Audiophilia" in *Social Studies of Science.* vol. 34(5). p. 783.

Pinch, Trevor J. & Bijker. (1987). "The Social Construction of Facts and Artifacts: Or How the Sociology of Science and the Sociology of Technology Might Benefit Each Other" in *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology.* (eds.) Bijker; Hughes & Pinch. Cambridge: MIT Press.
----- & Trocco, Frank. (2002). *Analog Days: The Invention and Impact of the Moog Synthesizer* Cambridge, MA: Harvard University Press.
----- & Bijsterveld, Karin. (2003a). "Should One Applaud?" in *Technology and Culture.* July, 2003. vol. 44.
-----. (2003b). "Giving Birth to New Users: How the MiniMoog was Sold to Rock and Roll" in *How Users Matter: The Co-construction of Users and Technology.* Oudshoorn, N & Pinch, T. (eds.) Cambridge: MIT Press.
-----. (2006). "Voices in the Electronic Music Synthesizer: An Essay in Honor of Don Ihde" in *Postphenomenology: A Critical Companion to Don Ihde.* Selinger, E. (ed.) Albany, NY: SUNY Press.

Pinker, Steven. (1994). *The Language Instinct*. London: Penguin.

Polanyi, M. (1966) *The Tacit Dimension*. Garden City, NY: Doubleday & Company.

Pope, Stephen T. (1995). "Computer Music Workstations I have Known and Loved" in *Proceedings of the 1995 International Computer Music Conference*. p. 7.

Price, Derek de Solla. (1984). "Notes Towards and Philosophy of the Science/Technology Interaction" in *The Nature of Technological Knowledge: Are Models of Scientific Change Relevant?* (ed.) Lauden, Rachel. Dordrecht: D. Reidel Publishing Co.

Puckette, Miller S. (2002) "Max at Seventeen" in *Computer Music Journal* 26/4, pp. 31-43. MIT Press.
-----. (2002). "Using PD as a Score Language" in *Proceedings ICMC 2002*. pp. 184-187. and http://www-crca.ucsd.edu/~msp/publications.html (April, 2005)
-----. (2004). "A Divide Between 'Compositional' and 'Performative' Aspects of PD" reprinted from the PD convention in Graz 2004. http://www-crca.ucsd.edu/~msp/publications.html

Putnam, Hilary. (1963). "Brains and Behavior." *Analytical Philosophy, Second Series*, Oxford: Basil Blackwell, pp. 211-235.

Pye, David. (1968). *The Nature and Art of Workmanship.* London: Cambium Press.
----. (1969). *The Nature of Design.* New York: Reinhold Books Corporation.

Pylyshyn. Z. (1986). *Computation and Cognition.* Cambridge, MA: MIT Press.

Ramon, Pelinski. (2005). "Embodiment and Musical Experience" in *Trans: Transcultural Music*

Review. vol 9. <http://www.sibetrans.com/trans/> [accessed May 2009].

Ramos, Vitorino. (2002). “On the Implicit and on the Artificial: Morphogenesis and Emergent Aesthetics in Autonomous Collective Systems.” In *Architopia. Art, Architecture and Science*. (eds.: Maubant & Moura) Lisboa: Ministerio da Ciencia e Tecnologia.

Reas, Casey & Fry, Ben. (2007). *Processing: A Programming Handbook for Visual Designers and Artists*. Cambridge: MIT Press.

Regier, Terry. (1996). *The Human Semantic Potential: Spatial Language and Constrained Connectionism*. Cambridge: MIT Press.

Rendall, F.G. (1932). "The Saxophone before Sax" in *The Musical Times*, vol. 73, No. 1078 (Dec. 1, 1932), pp. 1077-107.

Reich, Steve. (1968). *Music as Gradual Process*. <http://www.oberlin.edu/~jaltieri/gradualprocess.html> [Accessed June 2002].
-----. (2002). Writings on Music 1965-2000. Oxford: Oxford University Press.

Rheingold, Howard. (2000). *Tools for Thought: The History and Future of Mind-Expanding Technology.* Massachusetts: MIT Press (first published 1985)

Richard, Dominique. (1996). “Code-notes-music: an epistemological investigation of algorithmic music” in *Organised Sound* vol. 1 (3), pp. 173-7. Cambridge University Press.

Ricoeur, Paul. (1981). *Hermeneutics and the Human Sciences.* Cambridge: Cambridge University Press.

Riddell, Alistair. (2001). “Data Culture Generation: After Content, Process as Aesthetic” in *Leonardo*, vol. 34 (4), pp. 337-343.

Risset, Jean-Claude. (1985). "Digital Techniques and Sound Structures in Music" in *Composers and the Computer.* (ed.) Roads, Curtis. San Francisco: Morgan Kaufmann.

Roads, Curtis. (1996). *The Computer Music Tutorial*. Cambridge, MA: The MIT Press.
-----. (2001). *Microsound*. Cambridge, MA: The MIT Press.

Rohrhuber, Julian; Campo, Alberto de & Wieser, R. (2005). "Algorithms today - notes on language design for just in time programming". in *Proceedings of International Computer Music Conference. ICMC*. Barcelona: Escola Superior de Música de Catalunya.

Rogers, Yvonne. (2004) "New Theoretical Approaches for Human-Computer Interaction" in *Annual Review of Information, Science and Technology* 38: 87-143.

Rooney, David. (1997). "A Contextualising, Socio-Technical Definition of Technology: Learning from Ancient Greece and Foucault". in *Prometheus*, 15: pp. 399-407.

Root-Bernstein, Robert S. (2001). “Music, Creativity and Scientific Thinking” in *Leonardo*. Vol. 34, No. 1, pp. 63-68.

Ross, Andrew. (1991). *Strange Weather: Culture, Science, and Technology in the Age of Limits.* New York: Verso.

Roszak, Theodore. (1969). *The Making of Counter Culture: Reflections on the Technocratic Society and Its Youthful Opposition.* Doubleday.

Rothenberg, David & Ulvaeus, Marta. (2001). *The Book of Music & Nature.* Wesleyan University Press, Middletown.

Rovan, J; Wanderley, M; Dubnov, S and Depalle, P. (1997). "Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance." Kansei, The Technology of Emotion. *Proceedings of the AIMI International Workshop*, A. Camurri, ed. Genoa: Associazione di Informatica Musicale Italiana, pp. 68-73.

Rowe, Robert. (2001). *Machine Musicanship.* Cambridge, MA: MIT Press.
-----. (1993). *Interactive Music Systems: Machine Listening and Composing*. Ohio: NetLibrary.

Rubin, Jeffrey. (1994). *Handbook of Usability Testing*. New York: John Wiley and Sons.

Rumelhart, David E; McClelland, James L & The PDP Research Group. (1986). *Parallell Distributed Processing: Explorations in the Microstructure of Cognition.* vol 1. Cambridge: MIT Press.

Russolo, Luigi. (1913). *The Art of Noise*, 1913. (webarticle) http://www.futurism.fsnet.co.uk/manifestos/manifesto09.htm, Jan, 2002.

Rutsky, R.L. (1999). *High Techné: Art and Technology From the Machine Aesthetic to the Posthuman*. Minneapolis: University of Minnesota Press.

Ryan, Joel. (no date). *Some Remarks on Instrument Design at STEIM.* <http://www.steim.org/steim/texts.php> [accessed May 2008].

Ryle, Gilbert. (1949). *The Concept of Mind*. Oxford: Oxford University Press.

Sachs, Curtis. (1940). *The History of Musical Instruments.* New York: Norton.

Sacks, Harvey. (1992). *Lectures on Conversation*. Cambridge: Blackwell.

Sangild, Torben. (2002). *The Aesthetics of Noise.* Copenhagen: Datanom. <http://www.ubu.com/papers/noise.html> [accessed July 2008]
-----. (2004). "Noise - Three Musical Gestures" in Journal of Music and Meaning. Spring 2004, section 4. <http://www.musicandmeaning.net/issues/showArticle.php?artID=2.4> [accessed December 2007]

Said, Edward W. (1978). *Orientalism*. New York: Pantheon Books

Sardar, Ziauddin. (2000) "Alt.civilizations.faq" in *Cybercultures Reader.* London: Routledge.

Saussure, Ferdinand de. (1983). *Course in General Linguistics*. London: Duckworth.

Schiemer, Greg. (1999). "Improvising Machines: Spectral Dance and Token Objects" in *Leonardo Music Journal*. vol. 9 (1), pp. 107-114.

Schmitt, Antoine. (2001). "Artiste - programmeur". <http://www.gratin.org/as/txts/index.html> [accessed March 2009]

Schneider-Hufschmidt, M; Kuhme, T & Malinowski, U. (1993). *Adaptive User Interfaces: Principles and Practice.* North-Holland, London.

Shneiderman, Ben. (1983). "Direct manipulation: a step beyond programming languages," *IEEE*

Computer 16(8) (August 1983), pp. 57-69.
-----. (1993). "Beyond intelligent machines: just do it! in *IEEE Software,* 10(1).
-----. (1989). "A nonanthropomorphic style guide: overcoming the Humpty Dumpty syndrome, *The Computing Teacher,* 16(7), 5.
-----. (1997a). "Direct manipulation vs. interface agents" in *Interactions*, vol 4 (6).
-----. (1997b). "Direct Manipulation for Comprehensible, Predictable, and Controllable User Interfaces" in *Proceedings of IUI97, 1997 International Conference on Intelligent User Interfaces.* Orlando, FL.
----- & Plaisant, Catherine. (2004). *Designing the User Interface: Strategies for Effective Human–Computer Interaction*. 4th ed. Boston: Addison Wesley.

Schnell, Norbert & Battier, Marc. (2002). "Introducing Composed Instruments, Technical and Musicological Implications" in *NIME 2002 Proceedings*. Limerick: University of Limerick, Department of Computer Science and Information Systems.

Schwarz, K. Robert. (1996). *Minimalists*. London: Phaidon.

Selfe, Cynthia L. (1985). "The Electronic Pen: Computers and the Composing Process" in *Writing On-Line*, J. Collins and E. Sommers (Eds.). Urbana, IL: NCTE. pp. 55-66.
-----; Holdstein, D. (eds.) (1990). *Computers and Writing: Theory, Research, and Practice*. New York, NY: Modern Language Association.
-----; Wysocki, Anne, Johnson-Eilola, Johndan & Sirc, Geofferey. (2004). *Writing New Media: Theory and Applications for Expanding the Teaching of Composition*. Logan, UT: Utah State University Press.

Selinger, Evan. (2006). *Postphenomenology: A Critical Companion to Don Ihde*. Albany, NY: SUNY Press.

Serafin, Stefania & Young, Diana. (2004). "Toward a Generalized Friction Controller: From the Bowed String to Unusual Musical Instruments" in *Proceedings of NIME 2004*. http://hct.ece.ubc.ca/nime/2004/NIME04/paper/index.html (May, 2005).

Serra, Xabier; Leman, Marc & Widmer, Gerhard. (2007). *Sound and Music Computing: A Roadmap*. Brussels: The S2SConsoConsortium. <http://smcnetwork.org/roadmap> [accessed February 2009]

Shanken, Edward A. (1998). *The House that Jack Built: Jack Burnham's Concept of "Software" as Metaphor for Art.* http://www.duke.edu/~giftwrap/House.html, February 2002.

Sharp, Helen; Rogers, Yvonne & Preece, Jennifer. (2002) *Interaction Design: Beyond Human-Computer Interaction.* New York: Wiley.
-----. (1994). *Human-computer Interaction*. Pearson Education Limited. The Open University. Harlow.

Shulgin, Alexei & Goriunova, Olga. (2004). *Readme Edition 2004. Software Art and Cultures* Aarhus: Digital Aesthetics Research Centre.

Sidner, C.L.; Kidd, C.D.; Lee, C.H.; Lesh, N.B. (2004). "Where to Look: A Study of Human-Robot Engagement", *ACM International Conference on Intelligent User Interfaces* (IUI), pp. 78-84, January 2004.

Simondon, Gilbert. (1980). *On the Mode of Existence of Technical Objects*. London: University of Western Ontario, [original version1958].
<http://accursedshare.blogspot.com/2007/11/gilbert-simondon-on-mode-of-existence.html> [accessed August, 2008.

Smith, Merrit Roe & Marx, Leo. (eds.) (1994). *Does Technology Drive History?: The Dilemma of Technological Determinism*. Cambridge: MIT Press.

Smyth, Tamara & Smith, Julius. (2003). "A Musical Controller Inspired by the Cicada's Efficient Buckling Mechanism" in *Journal of New Music Research*. vol. 32.

Sorensen, Andrew. (2005). "Impromptu: An interactive programming environment for composition and performance" in *Proceedings of Australasian Computer Music Conference*. Brisbane: ACMA.
-----. & Brown, Andrew. (2007). "aa-cell in Practice: An Approach to Musical Live Coding" in *Proceedings of the International Computer Music Conference.* Copenhagen: ICMA.

Spiegel, Lauri. (1993). "Music Mouse™ - An Intelligent Instrument" Program and Manual: http://tamw.atari-users.net/Atari_Music_Mouse_Manual.pdf. [accessed June 2007].

Stiegler, Bernard. (1998). *Technics and Time, 1: The Fault of Epimetheus.* Stanford: Meridian. Stanford University Press.
-----. (2003a). "Technics of Decision: An Interview" in *Angelaki: Journal of Theoretical Humanities.* 8:2, 151-168.
-----. (2003b). "Our Ailing Educational Institutions" in Culture Machine. vol 5. <http://culturemachine.tees.ac.uk/Cmach/Backissues/> [accessed March 2009].
-----. (2009). *Technics and Time, 2: Disorientation.* Stanford: Meridian. Stanford University Press.

Sterilny, Kim (2004). "Externalism, Epistemic Artefacts and the Extended Mind" in *The Externalist Challenge*. (ed.) Richard Schantz. New York: Walter de Gruyter.

Stokes, Patricia. (2006). *Creativity from Constraints: The Psychology of Breakthrough.* New York: Springer Press.

Suchman, Lucy. A. (1987). *Plans and Situated Actions: The Problem of Human-Machine Communication.* Cambridge: Cambridge University Press.
-----. (1994). "Do Categories Have Politics? The language/action perspective reconsidered" in *Computer Supported Cooperative Work* (CSCW) 2. pp. 177-190.
-----. (2002). Practice-Based Design of Information Systems: Notes from the Hyperdeveloped World" in *The Information Society.* vol 18.
-----. (2002) (with R. Trigg and J. Blomberg). "Working Artefacts: Ethnomethods of the prototype" in *British Journal of Sociology* Vol. 53, No. 2:163-179.

Sudnow, David. (2001). *Ways of the Hands.* Cambridge, MA: MIT Press.

Supper, Martin. (2001). "A Few Remarks on Algorithmic Composition" in *Computer Music Journal*, 25:1, pp. 48-53, Spring 2001. Massachusetts Institute of Technology.

Swift, Gordon N. (1990). "South Indian Gamaka and the Violin" in *Asian Music: The journal of the Society for Asian Music*, Volume XXI-2.

Tanaka, Atau. (2004). "Mobile Music Making" in *Proceedings of NIME 2004*. http://hct.ece.ubc.ca/nime/2004/NIME04/paper/index.html (May, 2005).

Tanzi, Dante. (2001). "Observations about Music and Decentralised Environments" in *Leonardo*. vol. 34 (5), pp. 431-436.

Taylor, Mark C. (1997). *Hiding*. Chicago: The University of Chicago Press.

Taylor, Timothy D. (2001). *Strange Sounds: Music, Technology, and Culture.* New York: Routledge.

Tedre, Matti. (2002). "Ethnocomputing: A Multicultural View on Computer Science". University of Joensuu. MSc Thesis.
-----; Sutinen, Erkki; Kähkönen, Esko; Kommers, Piet. (2006). "Ethnocomputing: ICT in Social and Cultural Context" in *Communications of the ACM* vol. 49 (1). pp.126-130.
----- & Eglash, Ron. (2008). "Ethnocomputing" in *Software Studies: A Lexicon.* (ed.) Matthew Fuller. Cambridge: MIT Press.

Tehrani, J., and M. Collard. (2002). Investigating cultural evolution through biological phylogenetic analyses of Turkmen textiles. Journal of Anthropological Archaeology 21:443–63.

Temkin, Ilya. (2004). "The evolution of the Baltic psaltery: A case for phyloorganology" in *The Galpin Society Journal.* vol. 57. pp. 219–30.

Terrenghi, Lucia; Kirk, David; Sellen, Abigail & Izadi, Shahram. (2007). "Affordances for Manipulation of Physical versus Digital Media on Interactive Surfaces" in *CHI 2007 Proceedings*, San Jose, CA, USA. pp. 1157-1166.

Théberge, Paul. (1997). *Any Sound you can Imagine: Making Music/Consuming Technology.* Hanover, NH: Wesleyan University Press.

Thompson, Evan (2005). "Sensorimotor subjectivity and the enactive approach to experience" in *Phenomenology and the Cognitive Sciences*.
----- & Stapleton, Mog (forthcoming) "Making Sense of Sense-making: Reflections on Enactive and Extended Mind Theories" in *Topoi*. vol 27. (2).

Toop, David. (1995). Oc*ean of Sound: Aether Talk, Ambient Sound and Imaginary Worlds*. London: Serpent's Tail.

Traube, Caroline, Depalle, Philippe & Wanderley, Marcelo. (2003). "Indirect Acquisition of Instrumental Gesture Based on Signal, Pheysical and Perceptual Information" in *Proceedings of NIME 2003*. http://hct.ece.ubc.ca/nime/2003/onlineproceedings/TOC.html (May, 2005).

Trescott, Martha M. (1979) (ed.) *Dynamos and Virgins Revisited: Women and Technological Change in HIstory.* Lanham, MD: Scarecrow Press.

Trosby, Finn. (2004). "SMS, the Strange Duckling of SMS" in *Perspectives in Telecommunications. Teletronikk.* Oslo: Telenor.

Truax, Barry. (1989). "Computer Language Design and the Composing Process" in *The Language of Electroacoustic Music*. Emmerson, S (ed.) London: MacMilan Press.

Tullis, Thomas S. & Stetson, Jacqueline N. (2004). "A Comparison of Questionnaires for Assessing Website UsabilityAssessing Website Usability", in Proceedings of Usability Professionals Association 2004 International Conference. Minneapolis: Minnesoda.
----- & Albert, Bill. (2008). *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. San Francisco: Morgan Kaufmann.

Tuomi, Illka. (2001). "Internet, Innovation, and Open Source: Actors in the Network" in *First Monday*. Issue 6 (1). <http://outreach.lib.uic.edu/www/issues/issue6_1/tuomi/index.html> [accessed February 2009].

Turk, Matthew. "Perceptual User Interfaces" in *Frontiers of Human Centered Computing, Online Communities and Virtual Environments*. Earnshaw, Rae A; Guedj, Richard; Dam, Andries van & Vince, John. (eds.) London: Springer, 2001.

Turkle, Sherry. (1984). *The Second Self: Computers and the Human Spirit.* New York: Simon and Schulster.
-----. (2007). *Evocative Objects: Things we Think With.* Cambridge MA: MIT Press.
-----. (2008). *Falling for Science: Objects in Mind*. Cambridge MA: MIT Press.

Upton, Catherine, Eaglestone, Barry & Ford, Nigel. (2003). "The Compositional Processes of Electroacoustic Composers: Contrasting Perspectives" in *Proceedings of ICMC 2005.* Barcelona, University of Pompeu Fabra, ICMA, 2005.

Varela, Francisco; Thompson, Evan & Rosch, Eleanor. (1991). *The Embodied Mind*. Cambridge, MA: MIT Press.
----- & Bourgine, Paul (eds). (1992). *Toward a Practice of Autonomous Systems: proceedings of the first European conference on Artificial Life.* Cambridge, MA: MIT Press.

Varèse, Edgard. (1936). "The Liberation of Sound" from the lecture series of *Perspectives on New Music, New Instruments and New Music* given in 1936.
-----. (1939). "Music as an art-science" in *Source Readings in Music History*. (ed) Oliver Strunk. London: W.W. Norton and Company, 1998: 1339-1345.

Vera, A. H. & Simon, H. A. (1993). “Situated Action: A Symbolic Interpretation” in *Cognitive Science*. vol 17. pp 7 – 48..

Verfaille, Vincent; Zoelzer, Udo & Arfib, Daniel. (2006). "Adaptive Digital Audio Effects (A-DAFx): a New Class of Sound Transformations", IEEE Transactions on Audio and Speech Signal Processing. vol 14 (5).

Verplank, Bill; Sapp, Craig & Mathews, Max. (2001). "A Course on Controllers", in *Proceedings of NIME 2001*. http://hct.ece.ubc.ca/nime/2001/program.html (May, 2005).

Virilio, Paul. (1997). *Open Sky.* London: Verso.
-----. (1995). *The Art of the Motor*. Minneapolis: University of Minnesota Press.

Ward, Adrian; Rohrhuber, Julian; Olofsson, Fredrik; McLean, Alex; Griffiths, Dave; Collins, Nick & Alexander, Amy. (2004). "Live Algorithm Programming and a Temporary Organisation for its Promotion" in read_me Software Art & Cultures Edition.

Waters, Simon (2000). "The musical process in the age of digital intervention" in . ARiADA Texts 1. <http://www.ariada.uea.ac.uk> [accessed July 2005].
-----. (2007). "Performance Ecosystems: Ecological approaches to musical interaction." in EMS : Electroacoustic Music Studies Network – De Montfort/Leicester 2007

Waisvisz, Michel. (1999). "Gestural Round Table" in *STEIM Writings.* <http://www.steim.org/steim/texts.php?id=4> [accessed July 2008]
-----. (2003). "Composing the Now", 2003 lecture: http://crackle.org/composingthenow.htm, Accessed: Sept 2005.
-----. (no date). "Waisvisz on Gestural Controllers". http://crackle.org/MW's%20gestural%20round%20table.htm, Accessed: Sept 2005.
-----. (no date). "Touchstone" http://crackle.org/touch.htm, Accessed: Sept 2005.

Walker, Gabrielle. (2003). "The Collector" in *New Scientist*. vol. 179. issue 2405. July 2003. p. 38.

Wall, Larry. (1999). "Perl, the first postmodern computer language". <http://www.wall.org/larry/pm.html> [accessed June 1999].

Wanderley, Marcelo M. (2000). "Gestural Control of Music". Online paper at IRCAM: <http://recherche.ircam.fr/equipes/analyse-synthese/wanderle/Gestes/Externe/kassel.pdf> [accessed November 2008].
-----. (2001). "Performer-Instrument Interaction: Applications to Gestural Control of Sound Synthesis". PhD thesis. Paris. http://www.ircam.fr/wanderle/Thesis/Thesis_comp.pdf
-----., and Battier, M. (2000). (eds). *Trends in Gestural Control of Music*. Paris: Ircam - Centre Pompidou.
-----. (2002). (guest editor). "Mapping Strategies in Real-time Computer Music." *Organised Sound*, vol. 7(2).
-----. & Orio, N. (2002). "Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI". *Computer Music Journal* 26:3, 62-76. MIT Press.

Wang, Ge. (2008). *The ChucK Audio Programming Language: "A Strongly Timed and On-the-fly Environ/mentality".* PhD thesis. New Jersey: Princeton University.

Weber, Eugene. (1992). *Movements, Currents, Trends: Aspects of European Thought in the Nineteenth and Twentieth Centuries.* Lexington: D.C. Heath and Co.

Weber, Max. (1958). *The Rational and Social Foundations of Music*. Illinios: Southern Illinios University Press. First published 1921.

Weibel, Peter. (1996). "The World as Interface" in *Electronic Culture.* Druckery, T. (ed.) New York: Aperture.

Weinberg, Gil; Driscoll, Scott & Thatcher, Travis. (2006). "Jam'aa – A Middle Eastern Percussion Ensemble for Human and Robotic Players" in *Proceedings of ICMC 2006*, New Orleans.

Weiser, Mark. (1993). "Hot Topics: Ubiquitous Computing" in *IEEE Computer*, October 1993.

Wessel, David & Wright, Matthew. (2001). "Problems and Prospects for Intimate Musical Control of Computers", in *Proceedings of NIME 2001*.
http://hct.ece.ubc.ca/nime/2001/program.html (May, 2005).
-----., M. Wright, and J. Schott. (2002). "Intimate Musical Control of Computers with a Variety of Controllers and Gesture Mapping Metaphors." In *Proceedings of the 2002 International Conference on New Interfaces for Musical Expression* - NIME02, pp. 171-173. 2002.

Wheeler, Michael. (2007). *Reconstructing the Cognitive World.* Cambridge, MA: MIT Press.
-----. (2008). "Minds, Things, and Materiality" in *The Cognitive Life of Things: Recasting the Boundaries of the Mind*. Renfrew C. and Malafouris L. (eds.). Cambridge: McDonald Institute for Archaeological Research Publications.

Whitby, Blay. (2008). "Sometimes it's hard to be a robot: A call for action on the ethics of abusing artificial agents" in *Interacting with Computers*, vol. 20 (3) (May 2008).

Whitelaw, Mitchell. (2001). "The Abstract Organism: Towards a Prehistory for A-Life Art" in *Leonardo*. vol. 34 (4), pp. 345-348.
-----. (2004). *Metacreation: Art and Artificial Life.* Cambridge: MIT Press

Whorf, B. (1940). "Linguistics as an Exact Science. Language" in *Thought & Reality*. (ed.) J. B. Carroll. Cambridge: MIT Press.

Wiggins, Geraint. (2002). "Understanding (Musical) Creativity using AI methods". AI Symposium at the ESCOM 10th Anniversary Conference on Musical Creativity, Liège, Belgium.
-----. (2003). "Characterising Creative Systems". Proceedings of the IJCAI'03 Workshop on Creative Systems, ed. Bento, Cardoso & Gero, IJCAI.
-----. (2005). "Searching for Computational Creativity". New Generation Computing 24(3), pp. 209-222.

Williams, Rosalind. (2000). "All That Is Solid Melts into Air" in *Technology and Culture.* vol. 41. Baltimore: John Hopkins University Press.

Wilson, Stephen. (2002). *Information Arts: Intersections of Art, Science and Technology*. Cambridge: The MIT Press.

Winkler, Todd. (1995). "Making Motion Musical: Gestural Mapping Strategies for Interactive Computer Music". In *Proc. of the 1995 International Computer Music Conference*. pp. 261–264.

Winner, Langton. (1977). *Autonomous Technology: Technics-out-of-Control as a Theme in Political Thought.* Cambridge: MIT Press.
-----. (1980) "Do Artifacts Have Politics?" in *Dædalus: Journal of the American Academy of Arts and Sciences.* Vol. 109 (1). Winter.
-----. (1983). "Technologies as Forms of Life" in *Epistemology, Methodology and the Social Sciences.* (eds.) Cohen and Wartofsky. The Hague, Netherlands: Kluwer Academic Publishers. Republished in: Kaplan, David. M. (2004). (ed). *Readings in the Philosophy of Technology.* Lanham, Maryland: Roman & Littlefield Publishers.
-----. (1993). "Upon Opening the Black Box and Finding It Empty: Social Constructivism and the Philosophy of Technology" in *Science, Technology, & Human Values*, Vol. 18, No. 3 (Summer, 1993), pp. 362-378.

Winograd, Terry & Fernando Flores. (1986). *Understanding Computers and Cognition*. Norwood NJ: Ablex.
-----. (1996). (ed.) *Bringing Design to Software.* New York: ACM Press.

Wishart, Trevor. (1996). *On Sonic Art*. Amsterdam: Hardwood Academic Publishers.

Wittgenstein, Ludwig. (1969). *The Blue And Brown Books*. Oxford: Blackwell Publishers.
-----. (1968). *Philosohpical Investigations*. Oxford: Blackwell Publishers.

Woolgar, S. (1983). "Irony in the Social Study of Science" in *Science Observed.* (eds.) Knorr-Cetina, K & Mulkay, M. Milton Keynes: Open University Press.
-----. (1991)."The Turn to Technology in Social Studies of Science" in *Science, Technology & Human Values.* Vol. 16. pp. 20-50.

Wright, M; Freed, A; Lee, A; Madden, A. and Momeni, A. (2001). "Managing Complexity with Explicit Mapping of Gestures to Sound Control in OSC." in *Proceedings of the 2001 International Computer Music Conference*. San Francisco: International Computer Music Association.
-----. (2003). "Open Sound Control: State of the Art 2003" In *Proceedings of NIME 2003*. Montreal, Canada. 2003.

Wyatt, Sally. (2003). "Non-Users Also Matter: The Construction of Users and Non-Users of the Internet" in *How Users Matter: The Co-construction of Users and Technology.* Oudshoorn, N & Pinch, T. (eds.) Cambridge: MIT Press.

Xenakis, Iannis. (1996). “Tutorial Article. Determinacy and indeterminacy” in *Organised Sound.* vol. 1 (3), pp. 143-55. Cambridge University Press.

Yanagi, Soetsu. (1972). *The Unknown Craftsman: A Japanese Insight into Beauty.* New York: Kodansa International.

Young, Diana & Essl, Georg. (2003). "HyperPuja: A Tibetan Singing Bowl Controller" in *Proceedings of NIME 2003*. http://hct.ece.ubc.ca/nime/2003/onlineproceedings/TOC.html (May, 2005).

Z, Pamela. (1998). *A Tool is a Tool*. http://www.pamelaz.com/tool.htm, [accessed July 2006].

Zee, Emile van der & Nikanne, Urpo. (2000). *Cognitive Interfaces: Constraints on Linking Cognitive Information.* Oxford University Press, Oxford.

Zmoelnig, IOhannes & Eckel, Gerhard. (2007). "Live-Coding: An Overview" in *Proceedings of ICMC.* Copenhagen: ICMA.

Zölzer, Udo. (ed.) (2002). *DAFX: Digital Audio Effects.* Chicester: Wiley.

Østerlie, Thomas (2003). "Open Source: Innovation or Simply Chasing Tail-lights", paper for doctoral course in Science and Technology Studies. <http://www.idi.ntnu.no/~thomasos/paper/kult8850.pdf> [accessed February 2009].

Appendix I

Survey of Audio Programming Language Designers

Participants were: James McCartney (SuperCollider), Miller S. Puckette (Max and Pure Data), Andrew Sorensen (Impromptu), Ge Wang (ChucK), Paul Lansky (CMix), Phil Burk (JSyn) Torsten Anders (Strasheela), Peter Stone (Symbolic Composer), Matt Wright (Open Sound Control) and Graham Wakefield (LuaAV).

AI.1 Rationale

- **Why did you embark on the project of designing a new musical language?**

McCartney: I wanted to make a tool for myself. I wanted to be able to write expressions that represented "parameterizable classes of sound processes". That is a phrase I wrote down early on as my goal. By "classes" I wasn't thinking of the object oriented term class, but class in the sense of a realm of sound. One thing I was interested in was live performance of algorithmic music. The tools that existed at the time (Csound, CMusic, CLM) weren't really made for real time. Max/MSP did not yet exist and even once it did it was too static for what I wanted to do. Csound was too low level. When I learned Csound I knew that I never wanted to actually have to use it.

Puckette: I couldn't do what I wanted with any of the existing ones..

Wang: I deeply enjoy programming and music-making, and even more generally, the act of building things. In both computer science and music, there are engaging aesthetic considerations engrained into their fabrics, and I was intensely interested in exploring this intersection. I also enjoy creating tools that others might find useful, and moreover, that provide different way of thinking. Aesthetically, from a music programming language perspective, I was both greatly inspired by existing systems, but also felt a longing for a certain expressiveness that, at the time, I could neither find nor even articulate yet. The ill-defined nature of the matter, combined by my love for programming and music, motivated me to create the ChucK programming language. It was and remains an exploration.

Sorensen: Impromptu really was somewhat fortuitous. In March 2005 I stumbled across Alex's Hacking Pearl article. At the time I was working on (yes yet another) real-time audio signal processing and scheduling architecture (C++) for real-time use in the computer games industry. I was also playing around with Genetic Programming (Koza etc.) for music composition and had naturally gravitated towards Common Lisp for this purpose. During this process I had fallen in love with Lisp. So after reading Alex's article I thought to myself, wow if I hack Common Lisp ontop of my C++ code I would have a pretty nifty live coding tool. I pretty much prototyped the idea over a weekend and realized that I had a pretty powerful tool. Of course since then the whole things been rewritten three times etc. but that's the original genesis of the project.

Lansky: this was in the 1970's. I wanted a language that would allow me to emulate a rehearsal rather than a performance (i.e. work on sections), and a mixing metaphor (add to disk rather than write from scratch)

Burk: HMSL: I wanted to create precisely controlled sounds and could only do so by using a computer. Since there were no commercial packages available we had to write our own.

JSyn: I wanted to create interactive computer synthesis pieces that could run in a browser and thus be shared with a wide audience.

Anders: "Strasheela is a highly expressive constraint-based music composition system. Users declaratively state a music theory and the computer generates music which complies with this theory. A theory is formulated as a constraint satisfaction problem (CSP) by a set of rules (constraints) applied to a music representation in which some aspects are expressed by variables (unknowns). Music constraint programming is style-independent and is well-suited for highly complex theories (e.g. a fully-fledged theory of harmony). Results can be output into various formats including MIDI, Lilypond, and Csound."

Constraint programming has been used for doing music composition before. On the one hand, musical constraint satisfaction problems were implemented from scratch. Such an approach is generic, but requires much work and expertise. On the other hand, there existed a few frameworks which aimed at making music constraint programming accessible for a wider audience (e.g., composers). However, these existing systems are severely restricted in what kind of music theory model users can define. With Strasheela I aimed at creating a software which combines the benefits of these two existing approaches. A wide range of music theories can be computationally modelled in Strasheela (e.g., rhythm, harmony, counterpoint, musical form and a combination of all these), and implementing these theories is far more simple than implementing them from scratch.

I am developing a programming system which does things which existing systems did not do (see above).

BTW: I would say that I developed a composition system or an application framework, but not a new language. For example, I did not introduce any new programming semantics nor syntax.

Stone: I started writing Symbolic Composer at 1988 for my own use: trying out how fractals would sound, and also to convert AIDS RNA into music in a way that corresponds to the structure of genes and their expressions on protein folding. This work was done for the Finnish Science Centre Heureka on a project that aimed for a CD, which was then buried in a time-capsule of the concrete layer of the building.

3 years later British composer and music educator Nigel Morgan contacted me, and said he had found that the concepts of Symbolic Composer are musically solid - as wide as the traditional music notation, but fully programmable on the algorithmic level.

This started a long collaboration with Nigel Morgan, and also with Janusz Podrazik, Jesper Elen, and with other composers, during we further extended the language for contemporary and experimental music purposes.

Matt Wright: To meet our own practical research needs.

Wakefield: Mainly, from using other languages I could see the vast potential of the computational domain for musical (and audiovisual) exploration and expression. The greatest contribution Max Matthews made (according to JC Risset and J Chowning, at a recent talk here at SB), was to open up the domain of computer sound synthesis for others to explore, to enable research and musical experimentation. But at a certain point, conceptual allegiances and design decisions in each language effectively prevent certain kinds of explorations. These decisions may have been driven by a need for computational efficiency or limitations in the available languages and hardware (which may be less problematic today), some may have come from historical traditions which may be of less value to contemporary artists. I wanted to break down these barriers, both for my own compositions and for the community.

Typical examples: quantizing control updates to block-rate divisions makes it cumbersome or impossible to develop new forms of microsound. Separation of score and orchestra makes it difficult to respond to the active low-level sonic process with high-level behavior, valuable for generative systems. Non real-time compilation can interrupt the flow of composition activity, exacerbated in live coding. Black box units limit open-ended experimentation. And so on.

- **What was the design philosophy you had?**

McCartney: I wanted a language with a high level of expressiveness and a very dynamic synthesis environment. By dynamic I mean that signal chain scan come and go over time at high rates. And for expressiveness I wanted to be able to express, as concisely as possible, parameterizable classes of sound processes.

Puckette: I was thinking about the problems of real-time control, and trying to extend the notion of "unit generator" into a real-time context. The philosophy (I think) was to figure out the real-time semantics, timing, and interconnection format, then fill in the specific objects needed.

Wang: In many ways, I think ChucK embodied a "holistic" design principle. That is, we followed no strict design principles or philosophies. That being said, here were our general design goals:

• **Flexibility**: allow the programmer to naturally express ideas in code, and to flexibly create, edit, and maintain audio programs.
• **Time**: allow the programmer to program the passage of time, and to control and reason about time with precision and across a wide range of temporal granularities;
• **Concurrency**: allow the programmer to write parallel modules that share both data and time, and that can be precisely synchronized; provide a deterministic concurrent programming model for audio, minimizing the hassle and complexity of (preemptive) concurrent programming by taking advantage of time and events in the language.
• **Readability**: provide/maintain a strong correspondence between code structure and timing.
• **Rapid prototyping**: allow programs to be created and edited as they run, for rapid experimentation, pedagogy, and live performance.
• **A do-it-yourself language**: combine the expressiveness of lower-level languages and the ease of high-level computer music languages. Support high-level musical concepts, precise low-level timing, and the creation of “white-box” unit generators, all directly in the language.
• **Pedagogy**: make audio programming more accessible; an observation is that many people are willing to (learn to) program in order to make music, presenting an opportunity to teach programming more eﬀectively (possibly to people who would otherwise never learn to program). Conversely, the clarity and logic of a programming language can help teach computer music concepts.

Overall, it's exploring the idea of the "Strongly-timed" and "On-the-fly" programming language and paradigm.

Sorensen: To build a tool that allowed me to rapidly construct high quality work. To give an example, I made a difficult decision fairly early on in Impromptu's development to replace my own DSP engine (the original C++ engine) with Apple's AudioUnit plugin architecture. This was really flying in the face of what a "serious" computer music environment should be. But the reality was that I spent my time complaining that people using Logic and Abelton Live had it "easy" because they had all these fantastic plugins to use. In short, I was complaining because they got to use better quality tools than I did! And I was supposed to be the "serious" one. So, I built AudioUnit's directly into impromptu. Now I have the best of both worlds, a vast array of high quality sound processing engines at my disposal and the ability to control and manipulate them with complete programatic control. For me it really came down to a question of do I want to get real work done ... and the answer was a resounding yes. This philosophy has carried over onto the graphics side as well for me. The trick is to bring programatic control to all of these fantastic tools we now have at our disposal.

Lansky: take advantage of high level programming languages, in this case Fortran X

Burk: HMSL: Create a rich toolbox that would support a wide variety of compositional styles and experimentation and would run on inexpensive home computers.

JSyn: Create a simple toolbox of essential synthesis modules and leave the rest up to Java.

Anders: Sorry, I don't understand your question. Are you asking for the research goal? Or the methodology?

Stone: Before starting writing Symbolic Composer I had a long background on playing piano (starting at age 3), guitar (at age 14), building analog synthesizers, drum machines and sequencers (at age 16), and playing in several guitar and early techno-music bands.

My father was a professor of music, and made his themes on studying Bach fugue themes, chordal movements and alternative tuning systems. He was a good teacher that answered to my questions about music theory back them.

Two elder brothers of mine (from his earlier marriage) were masters of contrapunctus, the other had also written a formal book, which I had to study on early age.

However, between ages 25-35 I got involved on computers and programming. I learned a dozen of programming languages, studied logical philosophy, phonetics and linguistics.

Then came the MIDI, and I set up my computer for music. Unfortunately my fingers had lost all the knowledge how to play. So, I decided to do the music in the way it was in my mind. I picked up the best language I had learned, which was LISP, and started writing.

I had no special philosophy in mind - I just let it all flow "by ear", based on the theory and performance of music I had earlier learned, taken account the best ways to realize them on computer programming.

Matt Wright: Meet our own practical research needs within the context of current technology. Generality. Trying to avoid replicating the problems of MIDI or ZIPI. Stateless protocols.

Wakefield: I should talk on behalf of Wes also, since one of the goals was to interweave audio and visual processes more flexibly. Simply, we've been constantly pushing for increased flexibility and dynamism. It was important to distinguish the problem space as creativity support rather than productivity support, for a particular orientation of composition that is experimental and engaging with the nature of computation, rather than simply using computers for a priori tasks. We have been calling this computational composition.

It is clear that computational composition demands some kind of formalism or codified representation, yet no representation can ever be completely generic. One option is the have a plurality of languages and representations, each capitalizing on a particular seam, a good domain fit, which is the current state of affairs. But if the task is to actively explore within the realm of computation the plurality itself can become a limiting factor. The key issue then becomes finding means to interconnect the plurality, to gain modularity... that's why things like common audio formats, JACK, OSC etc. are so important. Our focus on the other hand is to work on a level closer to computation itself, using a more modular and dynamic language, embedding runtime code generation and so on, and effectively allowing the composer to define the structure of the space to be explored and expressed. An example of meta-design (designing an environment for designers.)

- **How much do you feel was "given" to you from older environments, operating systems, hardware, code libraries? To phrase it crudely, did you feel that you were a node in a network implementing a your vision or an inventor that had to invent everything?**

McCartney: What already existed was the idea of unit generators, and of languages like Scheme and Smalltalk. I also had this paper on a simple way to write a real time garbage collector. So really it only required synthesizing these ideas into one program. If invention is a novel assembly of existing pieces then SuperCollider can be called an invention. There were certain innovations in optimization, new unit generators, and the idea for Patterns. There were antecedents for Patterns in HMSL, DMix and CLM, but I think that Patterns improved on them in certain ways.
Puckette: I used the unit Generator concept from Music N, and also Max Mathews' and Barry Vercoe's ideas about real-time scheduling. Also, the designs of the early real-time digital synthesizers influenced my thinking. I didn't use any linguistic constructs as such.

Wang: For ChucK, we really fundamentally started from the "ground up", to provide the greatest degrees of freedom in exploring and evolving the programming language. Of course, we used many existing libraries and borrowed syntax and semantics from other systems. For example:

ChucK syntax (C + Java + ChucK)
Libraries (Synthesis Toolkit, libsndfile, SDL)

Sorensen: Definitely a node. Impromptu is - an AudioUnit plugin architecture (thanks Apple) , an OpenGL client (thanks SGI+World), a video processing engine (thanks shading languages), a Scheme interpreter (thanks TinyScheme) etc.. Having said this, I think it's also important to point out that anyone who thinks that you build these environments by just plugging API's together really doesn't understand what is involved. Almost everything you add has to be massaged into context - real-time, concurrent, dynamic, distributed etc.. As a result integrating these API's often requires substantial reworking - TinyScheme for example has been substantially modified for use in Impromptu.

Lansky: more to the latter, I wanted to redesign everything from the ground up

Burk: HMSL: We started from scratch in the early 80's writing our own compilers in assembly. We added cross platform GUI tools and an object oriented dialect. HMSL started before MIDI was defined. When MIDI came out we had to solder our own MIDI interfaces and write serial drivers with timestamps.

JSyn: The Java environment is very rich with networking, GUI tools, etc. But it was lacking synthesis tools and high resolution audio I/O. This was before JavaSound. So we have to write the synthesis and I/O in native 'C'. We built on existing traditional modules like envelopes and added a few new modules.

Anders: I consciously tried not reinventing the wheel again :) My work makes use of a large number of existing software and, possibly even more importantly, existing concepts. For example, I am using a particular programming language, because it provides many features which are beneficial for my software (e.g., very good constraint programming system, also I am using many programming paradigms and was happy to have a multi-paradigm language). Also for outputting results into music notation and sound I rely on various other software, and I created bridges for other systems so Strasheela could be used together with other software (e.g., Strasheela can run as a server in the background and be used by Lisp-based composition systems, and SuperCollider and Max can communicate with Strasheela via OSC).

Stone: I was mashing up things, which were coming from different branches. For example, the zone architecture I did model after RNA-DNA-Protein compiling process of the cell ... now, if I can make it analog to musical terms, then as the ribosome is capable of producing any life form, then the analog is capable of creating any music - this was what I aimed at the core compiler ... once this was ready and working as pure symbolic system I got the idea to try out how my semantic network algorithms that I had written 5 years earlier would sound as music - which blew up my mind ... next I started extending all kinds of algorithms and processing functions to the system ... and the first tonality systems I implemented came from my brother's book on classical scales and progressions .. that's how Symbolic Composer 1.0 came up.

Here I was focusing on making each new concept to match each other, so that any expression is transformable to any other expression, which became part of the basic architecture of Symbolic Composer, allowing later to snap in other modular extensions.

Matt Wright: Very much built on top of existing systems. We would have happily used an existing system completely as-is if there had been one that met our needs.

Wakefield: Definitely a node in a network! I am extremely grateful for the attitude of open source that has characterized the academic computer music community through most of its history.

AI.2 Aesthetics

- **Were your considerations (criteria for design) mainly personal or public user needs? Or both?**

McCartney:
Primarily personal. I think to design anything really good you have todesign something you want for yourself.
Puckette: I was thinking mostly of production needs at IRCAM. I was one of the production staff at the time, so it was partly personal but partly not.

Wang: Both

Sorensen: Almost entirely personal. I would say in general that I develop things that I need but I document for others (not that I document a lot ;)

Lansky: ease of use was not one. I wanted as much control as possible, and thus decided that this was more of a private approach.

Burk: I primarily wrote both these languages for composers that I admire, in particular Larry Polansky, David Rosenboom, Nick Didkovsky and Robert Marsanyi. They were also collaborators and co-developers.

Hundreds of other people also found these languages useful.

Anders: The needs were defined by myself -- coupled with a careful analysis of existing research in this field.

Stone: I did not think about that. I was working on the facts of music and math, which I needed to fuse together. On the interface level I considered graphical interface for making it easier to use, but since this reduced the capabilities, I decided to keep it as a written music language.

Matt Wright: The original design ("SynthControl") was only for our personal needs, but once we had it we realized that with just a little tweaking it would be suitable for public needs, so we did.

Wakefield: Both. Originally personal, but corroborated in community discussion, literature review, etc.

- **Did you have in mind a certain musical style or styles when designing the language?**

McCartney: No. I wanted to design a language that could represent structure as generically as possible. However with things like Patterns it is easier to use them to make rigid sounding music vs. more fluid timing, which I think is unfortunate, but that is the nature of computers.

Puckette: No, although IRCAM certainly was more open to some styles than others at the time. I aimed for stylistic neutrality.

Wang: Not really. We tried to keep things general, while being mindful that the language can't help but suggest/impose certain ways of doing things and indirectly the music made. For ChucK, we tried to take advantage of this latter point, rather fighting it.

Sorensen: No, almost explicitly not. In fact it is only recently that I have been shipping "music library" code with Impromptu. Prior to this I almost went out of my way to keep as much explicit musical knowledge out of the system as possible. As always happens though, you end up finding users asking for the same libraries that you use and then you have documentation mismatches etc. until it all just gets to hard and you include the music libraries in the baseline release. Another example is that I've only just recently started shipping AudioUnits with Impromptu for doing low level synthesis stuff - previously my attitude was that people should make their own. I guess this is one area where I have been bending to public need by shipping more of my personal libraries with Impromptu.

Lansky: no. I wanted it to be as flexible as possible. by the way I am talking now about MIX, which was written for an IBM mainframe. This is the predecessor to Cmix.

Burk: We wanted to support musical styles that were not supported by existing tools. I refer to this as "other music".

Anders: Strasheela consists in a core which is (aims to be) independent of any musical style, and various extensions (e.g., a model for harmony or motif models) which still aim to be highly generic but are more style specific.

Nevertheless, constraint programming is particularly well suited for dealing with musical aspects which are described from different viewpoints. This is the case for the pitch parameter, which is often described from different aspects such as harmony, melody or counterpoint. Consequently, Strasheela is particularly well suited to model music theories on pitches -- in contrast to most algorithmic composition approaches. This may lead to certain musical styles..

Stone: No, I was implementing the theory of music, and adding general-purpose algorithms on it. Each composer has his own style and progress, and hence the language should remain a tool that allows to realize that.

Matt Wright: We explicitly wanted to avoid proscribing musical style with OSC, in contrast to MIDI, which is founded on impoverished musical models such as representing everything in terms of "notes" with discrete pitches, single instants "on" and "off", etc.

Wakefield: No - I wanted it to be as open as possible.

- **Do you feel that your musical ideas are inscribed in the software structure? (can you (sometimes) hear your "signature" in the music produced by the software?)**

McCartney: Well sometimes I hear my own examples being used. That is an unintended signature because I would have hoped people would write their own sounds. Sometimes a particular unit generator can be distinguished.

Puckette: There's frequently a feeling of electronics-following-the-instrument in pieces using Max. That's not so much my personal style as one that emerged from what Max made easiest to do.

Wang: The hope, for ChucK, is that we can't do this - aside from characteristics of certain Unit Generators. But given the principle of encouraging a do-it-yourself way of working, people have definitely found hugely varied ways of creating music with ChucK.

Sorensen: I think this is difficult to answer, particularly for a programming environment. There are obvious applications of the software that are heavily influenced by the way I do things, but I'm not sure that this is really the environment talking - it's more that people watch my code examples and copy them. There are also examples of people doing things in completely different ways to me. It's interesting when people talk about the Max/MSP aesthetic or the Processing aesthetic - I'm not saying I don't agree with them, but you have to wonder if this is really a limitation, of affordance, of the environments, or just the sheep following the path of least resistance - which usually involves copying someone else's code.

Lansky: I can hear it in my own music, but not in the music of others using this software

Burk: We tried very hard not to do that. Judging from the incredible variety of pieces created, I think we succeeded.

Anders: Probably not. However, as mentioned before, Strasheela is good at creating music based on a theory of harmony (conventional or not). That could be noticeable.

Stone: I cannot directly hear them, but sometimes I get a feeling "this was certainly not played by hand, it must be a genius". Still, I do not get a credit on that, since it always requires the human mind no matter what tools he is using to express the vision. No tools can replace human intelligence, but Symbolic Composer has many of those that can help to jump on a new level.

I also enjoy fellow composers music that are using Symbolic Composer. Even on the most aleatoric levels beyond Shoenberg, Stockhousen, Messian, I can admire the logic of the piece with pleasure. But on the other hand, I also dig Techno, Europop, Hardtrance, Jazz, Top10 Hits, Industrial, Grunge, Garage - and J.S. Bach - anything that jumps ahead on their genres gives me a kick.

Wright: Definitely not.

Wakefield: To early to be evaluative.

AI.3 Response

- **How did you feel people responded to your software?**

McCartney: Originally, based on ICMC attendance, I figured that the world market for a new synthesis language was probably 40 people. I was surprised that as many people bought it as did when it was for sale. When it went open source the traffic at my website went up by a factor of eight. Once it was open source it took on a life of its own.
Puckette: People at IRCAM liked it right away - the older tools the were using were so hard to use, there really wasn't any difficulty getting people to use Max.

Wang: People have been highly generous, forgiving, and extremely encouraging! ChucK has a long way to go before it might be considered "mature".

Sorensen: In general fairly positively. Overall people seem to think it's pretty elegant.

Having said that there are an awful lot of people who won't even look at Impromptu because it's Lisp. Of course they've never tried to learn it. I'm seriously not kidding, the whole parenthesis thing is a huge issue. The usual complaint of course is that people can't read it with all those parenthesis. Hardly surprising when they've never spent more than a day trying. It's truly amazing how perocial people can be and there's no doubt that Algol won the syntax battle. Maybe I should swap to Ruby and see exponential growth ;)

Lansky: a select few who were willing to put in the time to learn programming languages (at first Fortran, and later C) became quite devoted, and some are still at it (Brad Garton, John Gibson, etc)

Burk: They have/had a small following of people who really enjoyed it. The pieces that were produced using these tools were often stunning and I am proud to have helped make them possible. I feel the creativity and significance of the pieces created outweigh their relatively small number.

Anders: I am surprised how many people are interested in this work -- given the facts that users are expected to use a niche-programming language and that they are expected to formalise music theories. For example, I am surprised about the number of downloads, or that important figures in the computer music community meanwhile know my name (just back from the ICMC).

Stone: It made a good feeling, and does so every time nowadays, too, when you get queries about it. At the first release it was really exciting to find out that there are indeed some others, who are interested in the same topics!

You spend years in a lab and only your girlfriend and closest friends hear what comes out ... first compilations also took a night to make, so it was really zooming into the unknown ... then you release it .. it hits some like-minded, and you get also kicks that it perplexes many others :-)

Wright: It took quite an effort at first to get anybody to use it, but by now (10+ years later!) it's impossible to avoid.

Wakefield: To early to be evaluative.

- **What is the best/worst criticism you've heard about your software?**

McCartney: Well there are lots of people who don't deal well with text based programming. People say that the learning curve is steep. Of course once you get past it you have more power. In some cases there were things that were more complex than they needed to be. The client server split added in SC3 has allowed more power and flexibility, but I feel the cost in added complexity was perhaps too much and I think that is probably a reason for some of the criticism about it being too hard to learn. One of the best critical experiences I had was when I tried to use SC1 in Berlin for a live performance to control audio for several performers and handle events from a data glove. I was able to see the shortcomings of SC1 for a larger project and that experience led to the ideas in SC2.

Puckette: Hoenig and Dessain et al. once complained it wasn't truly object oriented (reprinted in CMJ - a good read).

Wang: Some of the best criticism: "ChucK is like finding that word you were always looking for", "It's a power trip with personality", "great for teaching and learning"

Complaints: "the performance is slow compared to X, Y, Z" (this is definitely true)

Many more of both!

Sorensen: You have to laugh but I think one of the most severe and ongoing criticisms of my software, is why it's not open source :) Oh and of course the fact that it's Lisp ;) Oh yeah and also being OSX only seems to be offensive to some people ;)

Lansky: it's too complicated

Burk: Best: They make it possible to create music that would otherwise be impossible.

Worst: They are programming tools and require special skills to use. Thus they are somewhat elitist.

Stone: Best, these warm my heart:

"I love SCOM because it is so pure."
"It's really the most wonderful tool I have ever seen for music composition."
"SCOM steps beyond notation and playing skills, like no other software can do."
"SCOM is highly suggested tool to teach and study music theory."
"I don't want alternatives."

Worst, these hurt:
"I've used it for years, but I still cannot understand the concept."
"It is not what I was looking for, and I cancel my credit card payment."

I understand that many might get interested on the concept of programming music, but on the other hand the majority of music makers rely on keyboard interaction and sounds. Music language poses the music theory and additional concepts to the musician, which often requires formal music education to open up.

It also hurts to see the software on warez sites. Symbolic Composer, or other niche music programs are not like Cubase or Logic, that apply to all, and thus generate sufficient income even when it is illegally copied.

Niche music program developers sales are rare, and warez make it hardly possible to sustain constant porting and maintenance costs, that the Operating System and computer language platforms pose to the programmer -- I think you should focus also on this on your study.

Matt Wright: Best: OSC lacks enough standardization to allow "plug and play", so its use is limited to programmers.

Wakefield: To early to be evaluative.

- **Which events do you feel have increased the public awareness of your creation? (media coverage, "famous" users, awards, user communities, etc.)**

McCartney: I'm not sure. I think it is mostly word of mouth and some mentions by artists using it.
Puckette: I think it's almost all by word of mouth.

Wang: * ChucK User community (http://chuck.cs.princeton.edu/community/)
* Princeton Laptop Orchestra, 2005-present (ChucK is used from the beginning as the primary software platform for teaching, instrument design, sound design, etc.)
* Stanford Laptop Orchestra, 2008-present (ChucK is used as the primary platform here as well)
* Media Coverage: http://slork.stanford.edu/media/
* Used for teaching at Princeton, Stanford, UCSB, CalArts, Virginia, Georgia Tech, and many other institutions
* Winner: 2004 ICMA Best Presentation Award
For: "The Audicle: A Context-sensitive, On-the-fly Audio Programming Environ/mentality"
* Co-winner: 2004 ACM Multimedia Best Open-Source Software Competition
For: "ChucK : Programming Language for Real-time Audio and Multimedia"
* Winner: 2003 ICMA Best Presentation Award
For: "ChucK: A Concurrent, On-the-fly Audio Programming Language"

Sorensen: Hmm, I'm not really sure about this. I would guess that my own use of the software is probably the main point of contact for most people, either through the gallery page on the impromptu website or via my live performances using the software. Overall though the main point of dissemination is probably via download/news sites such as Apple's downloads pages, macupdate, kvraudio macintouch etc..

Interestingly wikipedia also seems to be becoming quite a significant knowledge distribution centre - there are impromptu links on a few wikipedia computer music related pages that see quite a lot of activity.

Impromptu did make it into the Computer Music magazine (the UK industry mag) I guess this gave it some exposure but I'm not sure how much (impromptu is pretty fringe for the Computer Music mag crowd ;)

Overall though, Impromptu isn't really a high volume proposition. There aren't any "famous" Impromptu users that I know about ;)

Lansky: I never imagined this would catch on in a big way. Cmix is basically a toolkit more than a programming language

Burk: HMSL: Concerts and papers at ICMC.

JSyn: Pieces on the web.

Both: Use as a teaching tool in many colleges.

Anders: Likely by web pages like freshmeat.net -- their content is mirrored by so many other web pages.. Also, by (every now and then) crossposting release notes to important computer music mailing lists.

Stone: Magazine CD ROM inclusions, and user references on sites were earlier very important on increasing the awareness. On the other hand Google brings up now increasingly many queries.

The reason for Google dominance is that information on new releases is now very hard to send, because of spam filtering on the sender site, and also because the receiver may even block a technical press release.

Awareness can be increased on sites that list libraries of music software for downloading. Their response to new additions has worsened, and they also keep changing their site structure that makes it time-consuming to submit the release.

Social networks and other user communities also take considerable efforts to set up and maintain, and easily focus the programmer to waste time on wrong direction.

Search Engine ranking algorithms also boost mostly referenced topics, which might make it difficult for a newcomer to get expose.

Famous users and awards were important earlier, but I'm not sure if anybody anymore pays attention to these cliches on the scattered social media/blog spheres?

Wright: ICMC 1997 was the unveiling of OSC to the academic computer music community. After that it's been a lot of behind-the-scenes work building the community.

Wakefield: Too early to evaluate; public awareness so far has been mostly limited to conference presentations and publications.

- **Has the usage of the software surprised you in any ways that you didn't foresee? How?**

McCartney: I've been happy to see many of the things that have been done, but I wouldn't say I'm surprised. When one creates a language, the whole point of that is to allow for things that you didn't foresee. Now what has surprised me is when people do things that are clearly wrong or dangerous, yet they get away with it. For example, Frederic Olofsson's using 700% CPU in performance and letting the audio break up.
Puckette: I first imagined Max as a tool for making classical computer music that would sound like real-time Music N. But users immediately started making beat loops and generative music. I wasn't expecting that at all!

Sorensen: Yes, I guess I have been surprised by the ways in which people want to connect impromptu to things. I guess this shouldn't be surprising but it's the variety of ways people want to do this that's interesting. Some people for example use just impromptu's audio side and do all their graphics in processing. Other people do only graphics in impromptu and pass all signal processing off to supercollider. I have one crazy impromptu user who loves to use csound and has been controlling csound from within Impromptu - certainly not my cup of tea ;)

Lansky: I was surprised that they could make it realtime, since the architecture I designed was specifically non-realtime

Burk: Yes. I did not foresee Larry Polansky using goldfish and video cameras to control vocal processors. I did not foresee Nick Didkovsky constructing a network based lottery where people won the rights to control the spectrum of a sound. Nor did I foresee Matthew Yee-King constructing a web based farm where users bred complex FM instruments using genetic algorithms. The list is very long.

Anders: The first paper which uses Strasheela and was not written by me is situated in music education for instrument performance. I did not expect that :)

Stone: It was fun to hear that it was used in making music for raves, which made me also interested in these genres.

Matt Wright: In many ways. The biggest, perhaps, is the use of Open "Sound" Control for video and other applications with no relationship to "Sound". I also did not predict that someday there would be what I've called the "second generation" of OSC implementations, idiosyncratic tools like many of the ones from IXI that only produce OSC messages based on interesting and quirky little worlds.

Wakefield: To early to be evaluative.

AI.4 The Technosocial Context

- **Have you had to deal with larger institutions (copyright issues, patents, media, software/hardware companies, academia) in your work with your creation? Has it been positive or negative? How?**

McCartney: Well as an employee of Apple now, it is virtually impossible for me to do anything with SuperCollider or any open source project. Apple owns everything I produce. Ultimately I will want to go back to composing, which means I'll be wanting to work on tools again. This will require me leaving Apple at some point.
Puckette: positive at first, then entirely negative. Without copyright protection it would have been impossible to get Max commercialized (which was important to get people to use it). But then as control of Max became the subject of a tug-of-war between IRCAM and Cycling, it became totally negative. That's what obliged me to start work on Pd :)

Wang: Not so much. Princeton University has been very supportive without adding any overhead to the process. Stanford University/CCRMA has also been great. ChucK is released under GPL, and being free and open-source, it removes much of the difficulties of the equation.

Sorensen: Yes, although by and large this hasn't been a huge problem. I basically insist on a clause in any work related contracts specifying that Impromptu is and will remain my sole property, and that any work done on the software during my period of employment remains my property.

Lansky: I decided to distribute it as source code, and didn't want the responsibility of maintaining it

Burk: Many of my customers have been colleges or commercial content producers. I have never had any conflicts. Most of this stuff is so wierd that no one is fighting over it. Also we established prior art in most cases so if we conflicted with later inventions no one said anything.

Anders: There were no such issues.

Stone: No.

Wright: We explicitly wanted something "open", so no to 1 and 2. We've had limited success over the years trying to get mainstream music hardware and software companies to support it; I'd say that the companies that have indeed embraced OSC have had a tremendously positive effect. Our only support at first and our main support for a long while was academia.

Wakefield: I hope not!

- **Eventually, did the user-base constrain or ease the development of the language?**

McCartney: It has made it easier. People are able to take care of the issues that concern them rather than waiting for me to do it.
Puckette: It eased it. People answered each others' questions on forums and suggested improvements, even contributing objects.

Wang: The user base has been fantastic in helping to provide feedback and direction for the language development/research. The process has constantly been a symbiotic integration of push forward new ideas and integrating community feedback and suggestions.

Sorensen: At this stage I don't consider the user base to be sizable enough to impact upon the language in any measurable way.

Lansky: In the mid 90's a group of users took it over and made Cmix realtime --> RTCmix

Burk: The biggest contributions from the users has been in terms of suggestions and feedback. The financial return has been minimal.

Anders: So far, user input was very helpful and specific (e.g., bug fixed, documentation writing, specific feature requests).

Stone: Feedback is essential to extend the language, and eases the development.

Matt Wright: OSC hasn't really developed much over the years. One huge improvement early on was the addition of type tags, which were put in single-handedly by James McCartney when he added it to SuperCollider.

Wakefield: To early to be evaluative.

- **If the software is open source, has that been beneficial? Any negatives? If not OS, why?**

McCartney: The benefit is that many things have been done that I would never have had the time to do. The negative is that I am not able to derive any income that would sustain my being able to devote full time to developing it. Also as the project grows larger like a snowball sometimes I feel it loses the core simplicity, or the ability for someone to quickly grasp what it is.

Puckette: Max wasn't, because it came too early (it was very hard to distribute OS software in 1989!) Pd was, and that was by far the more appropriate thing to do in its time (97). The main benefit of OS has been that it's not necessary to make design decisions based on marketing considerations.

Wang: ChucK is released under the General Public License. It's been awesome, because it's free, free, and open. It also reinforces the "hide nothing" principle.

Sorensen: Nope, it's not OS. I have had OS projects before, including jMusic which was a Java based music library, but more recently I've been wondering if OSing is worth the hassle. Maybe in the future, but at the moment I'm happy to keep Impromptu closed source. I have no intention to "commercialize" impromptu and it will remain "free as in beer" - so money is not a factor in my decision. My primary interest at this stage is to keep improving the tool in areas that are of particular relevance and importance to me, if others find this useful then great, otherwise ...

If for some reason I decided not to continue the project I would certainly open source so others could continue to use and develop the program if they wished.

Lansky: Open source. It has been great

Burk: HMSL: Yes, bug reports were often accompanied by the patch that fixed the bug.

JSyn: Not open source. I wrote this during the 90's and was feeling overly protective.

Anders: I feel that releasing Strasheela under the GPL was beneficial because it encourages the just mentioned user input. Also, the development of this software was and is publicly funded (research grants), so releasing it as open source makes sense for me.

Stone: Open Sourcing has many advantages and has been thought about, but the user-base has not yet made queries on that, so it is too early to push on that direction - maybe later.

Wright: We released the "OSC Kit", an open-source implementation of the features of OSC, with the intent that it would make it easier for people to add OSC to their applications, as well as a utility library for constructing OSC packets and command-line utilities dumpOSC and sendOSC to print received messages and generate messages from a command-line interface. I'm sure OSC wouldn't have succeeded without these. However, many people have ignored these implementations and coded their own from scratch, often omitting aspects of the standard, so that we now have very few true implementations of the complete spec.

Wakefield: It is open source (BSD-style licence). It is too early to evaluate whether this has been beneficial or not, but we consider it important to remain open source for the benefit of research in the community.

- **What are the biggest constraining factors? (operating systems, processor speed, users, money, time, etc)**

McCartney: The biggest constraint right now is the intellectual property agreement I'm under.

Puckette: The human learning curve.

Wang: Always wish there are more time to work!

Sorensen: The sound DSP side really doesn't have to many processor related issues - audio performance just isn't really an issue these days. However, this is certainly not the case for video processing and OpenGL related stuff where the more cycles the better. Impromptu is already pretty well set up for distributed and concurrent use so more cores etc. (even grid computing to a certain extent) are all good news for Impromptu.

Time and money are kind of reverse sides to the same coin. Money buys you time - with more time you need less money. So I guess time and money would both be nice but at the moment I'm lucky to have a research position that affords me "some" time for doing this work and enough money to buy beer!

Actually the biggest constraint for me is probably being stuck on the other side of the world from everyone. Australia is really pretty isolated, so it's not easy to catch up with peers (and inspiration) throughout the rest of the world. I'm not much of an offline communicator so face to face is really important for me.

Lansky: it's hard to improvise

Burk: Variable operating system latency or "jitter" was always a big problem. We had to use timestamps to get smooth timing. But that requires an increase in average latency.

Processor speed was not a huge issue. Our expectations grew comfortably along with the increase in speed.

Lack of money meant lack of time which was the biggest limiting factor in the development of these languages.

Anders: There are some technical problems which are hard to overcome. For example, constraint programming problems are hard to debug (in all existing systems). For finding logical errors, one has to understand the problem definition, which is time consuming. Also, the decision for a non-mainstream programming language has certain cons..

Nevertheless, after working so many years with this technique now, I still feel I made a very good decision in the beginning when deciding for me programming platform :)

Stone: Operating system changes pose a need to upgrade the system to run on a new platform. Developer licenses cost a lot, and it requires months, or even years to make the changes.

Wright: Development energy, and the lack of coordination among would-be developers.

Wakefield: Currently, free time of the developers!

AI.5 Evaluation

- **What are the problems with your software?**

McCartney: I think in some ways the move from SC2 to SC3 lost some good featureslike spawn, tspawn, pause, xfadetexture, etc. I would be interested inthe future to recover those.Defining a UI is still too complex. That was also better with SC2.

Puckette: The most serious problem is the lack of an underlying language with a grammar. This makes it very hard to do operations like searching and sorting, for instance.

Wang: Still many "disaster zones" in the implementation
Lacking garbage collection
Lacking better string support
Performance (CPU usage)
Many others!

Sorensen: I think the most significant problems for impromptu at the moment are performance and debugging. Impromptu's scheme interpreter is to slow to do some of the things I want it to at the moment. If you want to do any heavy numeric processing in impromptu at the moment you need to drop down to the Objective-C layer. In practice this really isn't to much of a problem as it's trivial to call into ObjC in Impromptu but I'd like to work on a JIT compiler for Impromptu at some stage. Also impromptu currently only has fairly rudimentary debugging support - reports back on an error but doesn't provide anything more advanced than a symbol name and an error message. Better debugging support is probably my highest priority at the moment.

Lansky: not fast enough, too complicated. In the late 90's I switched to SuperCollider

Burk: HMSL: Forth is too powerful. It is a scary read-only language that lets you erase memory with a couple keystrokes.

JSyn: Getting native code plugins to install reliably in most browsers has been the most difficult problem and is still not entirely solved.

Anders: The learning curve is steep. Also, it is not as generic as I would like it to be (I could be very specific on the latter if you ask for it).

Stone: Changes on OS and LISP platforms take a lot of work to keep updated.

Wright: Lack of standardized namespaces for common tasks, and incomplete implementations.

Wakefield: It is very young and encompasses only a fraction of the vocabulary of similar systems. Our hope is that the more flexible grammar can help cover this ground more quickly.

Even though the language used is remarkably easy to learn, the tool is definitely going to have a steep learning curve for anyone not already familiar with some level of programming or scripting. We are trying to improve documentation and runtime errors and plan to incorporate a debugger.

- **How do you see the future of your software?**

McCartney: I think when I have a chance to work on this again that I will write something new, smaller and more focused and perhaps more personal rather than continue building on the current version of SC. Actually my idea would be to write something that allows more for what is asked in the next question: writing code as a musical practice (though not necessarily a live practice). It would be a language that supports growing a library of functions for transforming representations of sound.
Puckette: Stasis. Max is too big to move. Pd will develop slightly as I clarify my vision for it, but even there I feel I can't make incompatible changes anymore.

Wang: Hopefully it can continue to provide different ways of thinking about audio and music programming. We are strongly committed to working to develop the language. One area is to combine audio synthesis and analysis in the same strongly-timed programming framework. For example, Rebecca Fiebrink (Princeton Graduate students) and I are working on this.

The Laptop Orchestra has been a potent experimentation ground for ChucK - as more laptop ensembles arise, I believe ChucK provides a powerful, rapid, and expressive way to enable laptop orchestras. At the same time, I also hope ChucK will be one of many different tools used in these contexts - I strongly believe in finding the right tool for the task at hand.

Sorensen: Who knows, I don't really want to guess. I like the idea that it continues to evolve to support my ongoing practice - whatever that may be.

Lansky: dunno.

Burk: HMSL: I have ported it to pForth and may re-release it for modern processors. The current release only runs on 680x0 machines like Mac Plus and Amiga.

JSyn: I would like to add spectral synthesis tools, FFT/IFFT, etc.

Anders: After so many years of development and scientific research, in the future I plan to primarily focus on actually using it for compositions. That may lead to some changes in the software. Also, I found someone who is interested to help be with the documentation – Graham Percival – which hopefully makes Strasheela more accessible for new users.

Stone: Symbolic Composer function library is written on standard ANSI Common Lisp, and its documentation is on ASCII standard. This means that it has a life-span of hundreds of years, or even more. It is the base for ANSI Common Lisp Music Standard for the future.

Matt Wright: The development of "best practices" and quasi-standardized naming conventions that will enable more plug and play.

Obviously there will continue to be more and more implementations, projects, etc.

Audio and video over OSC?

Wakefield: There may be a parallel problem with the data explosion we have already seen (the demand for richer detail can exceed human capacity of specification, necessitating delegation either to collaborators or automation.) Increasing algorithmic complexity beyond human capacity to specify would make our language representation somewhat inappropriate (just as assembly language is an inappropriate representation for the extreme complexity of modern software.) Code-generation may be a partial solution, but expanding upon collaborative capacity is also very important.

In the long term, I imagine there will be an abstract and portable representation of systems and data upon a small set of computational primitives. The benefit of this for portability (and against obsolescence) is clear. The capacity for middleware to bridge between such a representation and diverse platforms or hardware is increasing rapidly. The open questions we will need to tackle I think will be oriented toward expressivity: focussing on the interaction between the human and this abstract representation.

- **Finally, as a programmer and musician, do you have any comments on writing code as musical practice? Is embodiment important in musical performance?**

McCartney: I am very interested in composing music by composing functions. Whether it needs to be done in performance I am more ambivalent about. Watching people typing is

actually more boring for me than tape music. With tape music there is nothing to look at, so the audience is more free to meditate, introspect, look around or do whatever. As a performer I think there can be a problem with being tied to watching a screen when one is performing sound. When I used to do sensor based music I would not even look at a screen, preferring to navigate the space by ear.
Puckette: I think software is more a musical instrument than a work of art... as such, the software designer has to be sensitive to how the software will be "played" but from a very different point of view than that of the person who will do the actual playing. I don't know of anyone who is simultaneously a professional-quality instrument maker and instrument player. Similarly, I know only one master software maker who is also a master composer (Keith Hamel). There may be a sense in which the software embodies the musical practice, but if so, it's extremely indirect.

Wang: ChucK is definitely meant to provide ways to write code on-the-fly, for rapid prototyping, experimentation, and performance. I believe this can fundamentally transform how we think about the process of programming to create music. As a performing live coder, I am highly interested in exploring code as means to create musical gestures. I am also interested in investigating ways of communicating coding as expressive gesture to the audience (via projection). At the same time, ChucK can also be used for composition. Overall, I think writing code as musical practice is a powerful idea, and is a different beast from say, more traditional forms of musical practices. Both are important because they are fundamentally different.

Sorensen: Way to big a question to ask at the end here. I'll get back to you on this if I find some time :)

Lansky: I loved writing code, and seemed to have a knack for it. This is not the case with a lot of my peers. I always thought of it as the modern day analog of having an instrument shop where I could design, build, and play personal and idiosyncratic instruments. I don't quite know what you mean about embodiment in performance. I always assumed that the finished product would seem unrelated to the code used to generate it. Although I could often hear a relation, I never noticed that anyone else did.

Burk: Writing computer music code can result in some really beautiful code.

Unfortunately writing software does not provide a shortcut to creating beautiful music. One must devote time to mastering a musical instrument whether it is made of wood or silicon.

Anders: Sorry, I feel don't quite understand this question.

With a grain of salt, writing code instead of composition by hand could be compared with playing an instrument instead of singing. Playing an instrument (and writing code) is more abstract than singing (or composing by hand), and singing (composition by hand) is supposedly more charged emotionally. However, playing an instrument (and writing code) potentially opens up new sound worlds.

Stone: You have a vision - just make it! The audience is changing, they want solutions that work now on their studios and computers. If you make a cool GUI you can be very rich, but if you don't then pay attention on the standards.

Wright: Writing code is not part of my musical practice. Writing code for me is engineering that produces tools that I can then use for my musical practice. It's like changing strings on an instrument: necessary music-related work that allows the musical practice to occur.

Wakefield: Is this embodiment in the sense of a physical device, or in the sense of an embodied process (rather than a surface representation)?

I think we're in a very exciting time. Many of the technical hurdles have been or are being overcome, and I believe we are entering a period of explosive creativity with respect to computational composition. In terms of musical performance this is demonstrated well by the live coding phenomenon, but I think that we can also expect to see some quite remarkable works embodying complex, autonomously generative systems that break through the barrier of ad-hoc demonstrations into fascinating and engaging worlds.

Appendix II

Further responses to the question in Section 6.7.2.3.1

- I have performed live with ixiQuarks once and use it quite often in composing. I use ixiQuarks alongside a software package that I wrote in SuperCollider that sends MIDI signals, so ixiQuarks provides the live audio signal processing elements of most of my work.
- I have used it to noodle around on somewhat randomly over the past year. I would love to use it with Ableton or Reason but I don't know how to do that. It has great potential for soundscape creation but the controls are somewhat limited.
- When i want to move away from the common sound of Lloopp i do a gig with ixiQuarks to throw everybody off with something different.
- See above, but I primarily use it for live improvisation. It is really useful for making ideas work on the fly.
- Tried once but gave up.
- I've used it about 15 times in a period of two months, essentially for preparing and designing sounds that i mount in a traditional daw. always in a studio context. always with no precise idea, just experimenting. I like to create some short loops with small variations in iterative process but also long evolving textures.
- I use it mostly for capturing improvisation as sketches or creating samples
- I use it for creating grainy sounds and also use it for processing cello sounds for lush granular soundscapes
- I don't use it at all in my personal music work (but, I design my own instruments for that). But, I use it for fun, inspiration, and recommend it often to people who need a basic but very free environment to mess around with sound
- I often use ixi for live performances, mostly eletronic improvisation with other musicians. On the studio I use it a lot for doing sound-scapes, glitches and other sounds that are later used on more "traditional" compositions, mostly done in Ableton Live.
- I don't compose without it.
- Tried once but gave up.
- ixi is my favorite piece of software. Use it in the studio all the time. Use it with Forester and Logic.

Further responses to the question in Section 6.7.2.3.3

- It was a little hard to get a first, but after a few tries it made sense.
- Excellent, very fun environment to play with, very original instruments and effects, for the most part.
- Extremely interesting program
- Odd at first... but once I figured out how to get things rolling it was fun
- Exciting, visually stimulating, especially the scalesynth
- I got a little confused with this inputs - outputs logic and I hade some problems with my soundcard. It seems that ixi does not work when it is on 48.000hz - 24bits, wich is the configuration that I mostly use.
- I was always frustrated by the lack of an all-in-one tool for improvisational electronic music composition. The ixi tools were a revelation. The fact that you've put all of the best together in one package has been a godsend.
- I've been using your various tools since, probably, shortly after you started putting them out.... I enjoy the non-traditional/non 'linear' interface... I like that they are not

sequencers as such....

- About as close as possible to a dream come true. I was trying to write similar software in SuperCollider, but lack the time and expertise. It allows me to focus on writing music instead of spending days on a software component.
- I thought it was very intriguing. I had high hopes for it to be able to create shifting ambient layers and the such.
- Very user friendly and inventive.
- The very first time i download the program i got quite confuse abut the inteface. i guess this happened becaouse i normally dont read the read me or help of this kind of interfaced software, then i understood this was a very serious software and i went back to read the tutorials. Then evrything went fine.
- Has a few bugs which made it shut down, which is not a big problem as SuperCollider is so fast to start up again. The bugs were easy to work around and I got my soundcard to work fast.
- Stuff to learn but not very difficult
- A lot seemed very random when experimenting, as you gain experience you learn the purpose of the tools
- Intrigued, not sure if it was music software or a video game. Predators hooked me in, and I started exploring the others.
- I get high with the software.. I feel that I have found exactly what I was looking for.
- A little confusing at first, but worked it out.
- It seems immediately a cool software. Intuitive, easy to learn and inspire creativity.
- One of frustration, because it wasn't necessarily intuitive, since it is essentially an overlay for SC (I didn't know SC at the time). The opening and closing, order of events for preparations is not FOOL-PROOF. I always get things wrong, despite being a regular user.
- Cool. Fun to play with. Not very stable. Creative.
- ...it was quite complicated when started to use it, but take it as a plus as I do not like easy and simple tools.
- Wow, I can generate a lot of material in small amount of time. You can get a lot going really quickly. I think that the filters, noise generators, and granular instruments are the best parts.
- Hard to say, since ixiQuarks wasn't the first program of yours that I tried, and when I did try it, I was already excited about many of your other, more esoteric programs.
- It was quiet exciting although i didnt understand the operation of all your programs so i haven't gained all the possible enjoyment i could have
- A little confused at first but that quickly vanished as i started playing around.
- first a little obscure, then rather clear with the help of documentation.
- confusion
- excited but disapointed because hard to understand.
- It was exiting, but after some time i realized that it is built for somebody`s specific needs, not mine, it is not enough flexilbe...
- I thought it was the most amazing cool software ever...
- This is fun but useless to my style of music
- "Cool, I finally have something that really illustrates how granular synthesis works."
- SoundScratcher is somethin I've been wanting to do for some time. While I have been using commercial granular synthesis processors/plugs (like Riverrun, GRM Freeze), SoundScratcher gives me gestural control.
- I knew ixi some years ago with some isolated instruments. The enviroment is something you would want to find when you're new to 'computer music'...
- Total frustration
- Seemed pretty familiar, having used other modular style software. I quickly felt as if I were playing somebody else's suite of programs rather than creating my own.

- What the hell is this??? How do you use it???
- At first I was a little overwhelmed, but now I like it
- It's very different than anything else I've played with.

Further responses to the question in Section 6.7.2.3.4

- Yes it is. It cuts out all the bullshit and focuses on creation.
- Much less idiomatic, more "play" oriented instrument design - allows for more experimentation.
- Different. I'm thinking of Max and Reaktor being the software which one could obtain the similar results (with a lot of programming of course).
- There is no software like ixi
- Ixi is not based on “recording” and “composing” in traditional way. It seems to me that it is more a performance software (of course you can record what you do, but it is different), which is a good thing also.
- None. Waveform editors aside, ixiQuarks is the only software that I would ever consider using.
- I think it is generally different enough. Not many commercial programs that I have encountered have as much emphasis on randomness
- I feel that it really fills a gap in the commercial product range with its flexibility and original set of tools specific to computer music generation.
- It’s a pleasant change working in an environment that's different from most commercial products.
- Yes. ixiQuarks -- as well as the rest of your software -- feels as though it’s a piece of art in itself, and provides the user with a more visceral and "hands-on" way to do music.
- ixiQuarks seems like a more friendly MSP.
- I think it feels a bit like AudioMulch, but with a Toshio Iawai-ish edge. So you could say it’s roughly a cross between AudioMulch and Electroplankton, but really, it's quite unique.
- It has got a great potential for experimental music
- Yes, it is different. I can't really put my finger on it, but there is some intangible quality not found in other products.
- It is different because it is not a VST instrument, or a sequencer, and it's not like MAX/MSP or PD. It's something very modular.
- Best free instruments for experimental music.
- Yep. very different (at least from what i know). Reinforcing chaotic environments, rather than controlled actions.
- It’s very different from commercial products more centered imo on 'physical metaphors' (-> virtual analog and so on)
- I like the simple use of the software, not like others, where you can make a lot of things but it is hard to find how!
- Yes, it is kind of weird, something you would not find in a commercial software.

Further responses to the question in Section 6.7.2.3.5

- Well both in some ways, it fuses a lot of common and experimental stuff together.
- I'd say its general as it can be used for a whole host of things.
- I would say it is general, it covers many areas of sound creation.
- Specific with regards to the processes and actions available, a slight understanding of basic synthesis terms and methods, also with using audio samples

- I think its general in that it has a lot of uses. Perhaps it is specific in terms of the type of music or sounds you'd have to be interested in creating. I suppose it is also specific in terms of the experience of using it – the user would have to be interested in experimental compositional and sound generation tools.
- Specific, but that's no problem at all.
- I think it is specific for a more experimental way of thinking music (as it deals with the matter of sound itself). Maybe "abstract" is a better word (maybe that's the purpose of this software). But inside this way of thinking there is a huge range of possibilities. I have other friends that also use ixi software and they use it in a very different way of what I use and the results are also very specific.
- The software grows with the composer, so I would deem it both general enough to serve the casual composer and specific enough for the seasoned, more demanding artist.
- A bit of both... I've managed some almost straight ahead type of stuff with the tools and some very abstract...
- It's excellent. The designers seem to be highly skilled software engineers who also have a thorough understanding of composition and performance methodologies.
- Both I think. The textures can be applied to a number of different styles, but some of the instruments seem a bit rigid. I need to spend more time with it.
- Specific. It's oriented toward experimental musicians (which is great).
- I think it is a very specialised tool, one which only a quasi-programmer (a prosumer) can use. In terms of the sounds it can produce, it needs a direct compliment, possibly only one in order to work in a live or studio setup.
- I think is very specific for interaction and experimental electronic experiences.
- The software is specific, I would say academic with educational qualities.
- General enough to be useful and specific enough to be a viable virtual instrument. I generally cringe at the sound of a modeled soft synth, but find that one can make sounds that can not be duplicated through other software or hardware.
- It's somewhere in the middle, I think you can use it for very precise (specific) tasks, or you can get stoned and mess with it for hours just for fun (general)
- General at times, and specific at others. Some instruments need more help in the interface to learn how to use them.
- Depends on user. Can be both.
- I would say, for me, it's specific. It seems to be custom engineered for synaesthetes, people who think about sound in visual terms. I am glad it's this way. There's too much generalist software out there. That type of "me-too", "swiss-knife" bullshit is a dime-a-dozen and oh so very boring.
- I think it is general for a very specific crowd. Not everyone can or will want to use it but those that do will have a whole new world opened to them.
- Working within constraints is refreshing. Music is so big - of course ixi is specific.
- I think both terms apply. General - there are many tools that have many functions and the sounds generated can be relatively 'normal' or can get pretty far out. Specific - each tool has a character of it's own and can manipulate the sounds in a unique way

Further responses to the question in Section 6.7.2.3.6

- Yes it does. I perform live with it and the UI is clean and simple and allows me to focus on the music.
- It makes me think more about microtones for sure.
- I'm always working on modular ways to connect my instruments and compose. ixiQuarks is a constant reminder that simplicity, good design, and good sound can give even a somewhat incomplete system MORE than enough power for performing exciting music.

- It encourages a totally abstract way of thinking which can only heighten your compositional process.
- no.no.no.
- As I said, it reaches an abstract view of sound itself (what is kind of normal in erudite music). And then you start to think backwards: you don't have a idea and then try to do it the best as you can. In my experience with ixi I have started do develop compositions based on the new possibilities that this software can offer me.
- There are times when I allow the software to make "decisions" in a composition. I relish the fact that the software allows for randomness and "happy accidents".
- I'm an improvising/spontaneous composition type of performer... I find the tools open up a world of possibilities for sound manipulation
- Any software, in my opinion, restricts the composer, though this is not always a negative. I've also found that software cannot make up for lack of artistic vision and also, for audio synthesis and processing, good source material, live or pre-existing.
- It encourages creativity by exploring sound manipulation that is not always commonly done in the same way.
- Never thought of it in that way.
- No, ixiQuarks is only a tool for use
- Yes, yes and yes. More non-linearity, more serendipity.
- No. I want to be in charge of things.
- It perfectly suits my compositional pattern, so I'd rather say ixiQuarks helps me to move towards my musical aims I set up for myself
- Ixi Quarks enable you to produce non-periodical rhythms and implement new tonal material. That's great, but it wasn't ixi-quarks that inspired me to do so, but opposite: I was looking for a software that supports such things and was introduced to ixi-quarks.
- Differnt instruments always lead me in differnt directions, I still like to think of them as a tool to help reach what I'm trying to say.
- I think more visually about my music. I also think more free form.
- In my case was the other way around, I had very clear in my mind what I wanted to hear but needed the tools to manipulate sound with such freedom
- Encourages new directions in composition. Alternative GUI's release preconceived methods.
- I do think differently about my music now - and it does absolutely drive me in a whole realm of thinking - so it does both effect my compositional and live performance.
- SoundScratcher gave me a renewed interest in granular synthesis.
- I do think differently, I think more about sound, than pitch, and notes.
- Every tool you use affects the way you work. IQ tends to be improv oriented which perfectly suits my workflow.
- It doesn't so much change the way I think about sound as it allows me to connect with the sounds I'm making to organize them in new ways in real time. I find it very good for free-noise improv.
- Yes! It is NOT a sequencer! It is more focused on my inabilities than on my skills.
- Working within constraints is refreshing.
- It encouraged me to compose experimental pieces that can constantly evolve. Yes I easily find new directions. To be honest I hope that it will evolve in future and dig deeper to bring up new exciting features. In my opinion future belongs to this kind of instruments.
- Yes. it makes it difficult to just add...drums...bassline...it makes you LISTEN more....
- I'm still using traditional DAWs for composing because I like to automate effects a lot. I'm an occasional live performer. So I'm focusing on musicality and experimentation with ixi.
- Yes, less linear.
- Sure, it opens up for new ideas. Different ways to conceptually see problems.
- ixiQuarks requires almost 100% realtime control, which means if I want to execute

anything again, I've got to, for the most part, remember what I did. There are presets, but ultimately the user has to be the driving force for controlling most instruments. This forces a fairly intimate relationship with the sounds you're producing, how you produced them, etc. Rather than just getting a good thing and recording it, you've got to try it over and over again, which always allows for new discoveries.

- Right now, ixiQuarks is the heart and soul of my improvisations and I would say it wouldn't be quite the same without it. I wouldn't say it's quite as defining as my guitar or keys but it is important all the same.
- it creates a framework that guides the creation like any other tool

Further responses to the question in Section 6.7.2.3.7

- I use my hands for just the software, but when I record in it I usually use my guitar with it.
- Though it’s only just starting to use Wacom [pen tablet] support, this is exciting. I've experimented with a range of physical controllers, from accelerometer-based things, to light and video style sensors.... putting aside the sense that one is doing something "new" by using these sorts of sensing systems, the Wacom is easily the best physical controller for digital music I’ve used.
- Ixi controlling is very limited to that mouse and keyboard action, which can be a problem. With midi controlling on other softwares you can control different parameters of different instruments or effects at the same time. This is very important to make a use of it as a instrument, as you can develop a instant control of what you want at the time. I'm sad that it is not possible with ixi Quarks.
- It’s not the same as beating on drums but I find I can get to a space where I feel the instruments/tools are expressive... I groove along quite physically while I’m playing/improvising
- I only use my fingers to control my track-pad on my laptop. Acoustic instruments allow for a much more direct relationship to the melody and rhythm.
- Sorry, I don’t understand.
- As with any software, bodily control remains rather academic, but youngsters seem to think different. For me my keyboard gives more bodily control than fumbling with mouse on scale synth. In general, I think bodily control needs a hardware controller, indeed something like an acoustic instrument.
- Acoustic instruments will always be the best choice for musicians who desire god-like control over every sound, but this software comes as close as software can to giving the musician a similar feeling of god-like control.
- It is still a computer instrument, does not apply comparing it to a acoustic instrument.
- I use pretty much just my hands (mouse, MIDI controller), so not a lot of body movement involved. Sure it's completely different from acoustic instruments, because playing an instrument is always a predominantly mechanical process, whereas ixiQuarks is more about artistic decisions than skill. But I guess that goes for pretty much all music software.
- I feel like this is controlled more by creativity and the mind. It seems a bit tougher, but more pure, more true.
- I use my body to sit on a chair. It is a software!
- Software intended as an extension of the nervous system is still at its beginning stages. I have no interest in throwing myself around for the sole purpose of producing sound.
- I don't understand the question.

Further responses to the question in Section 6.7.2.3.8

- I cannot find any limitation.

- Not enough opportunities for control and personalization. This is the reason that, as it stands, I would never use ixi for music I would claim as my own - the instruments, to varying degrees, always sound like ixi instruments. Being able to use custom synthdefs and samples is a start, but more control is also needed too - predators is fun to play with, but ideally, I'd like to be able to broadly explore some options for generating sound, and then begin to narrow down my focus on something I like and be able to control it - many of the instruments are good at the first part, but not so good at the second.
- Sure you can't do everything on that software. But it is only a software, not a miracle. And it does very well (and even better) what his purpose is.
- In analyzing the software system I wrote in SC (nowhere near as complex as ixiQuarks), I found, after lots of introspection and listening, that it lacked a very important musical element- silence. I've added controls to allow extended rests between patterns and phrases. I think ixiQuarks would benefit from something like this, though there are creative ways of achieving it.
- ixiQuarks steers away from straightforward composition and playing- i.e. no keyboard, chromatic scales etc.
- I use ixi in its own philosophy and it's good for me.
- Limitations in positive way can be inspiration
- I think that ixiQuarks is for a non-usual music, is a way to find some interesting elements for musician. If you want play the melody - you can choose from thousand programs.
- hum... basic things? - that's not what I use ixi-quarks for.
- Naturally, because the software's strength (to me, anyway) is to create novel ways of approaching sound and music creation, It's a bit more difficult to use more traditional compositional and playing techniques. However, if I want something that is more comfortable to me, I'll use something I'm already very familiar with. The strength of this program is its novel approach, and yes, it is inspiring. I probably wouldn't have bothered downloading it if it didn't provide me with a different way of creating sound.
- I could say sequencing but that exactly why I find it so interesting, because it’s not based on a time grid or anything like it... Continuous results that emulate the performance of an acoustical instruments
- It is lacking a visual signal pathway.
- I like ixiQuarks for real time soundscaping. It makes me shudder to think of it turning into some kind of crappy sequencer. If you have to change anything, make it even weirder and more wonderful than it already is. Give it some sort of neural networking properties or self-regulating feedback systems. You're the software wizards--Dazzle me!
- Most of the limits are mine and not the software.
- Anything I can't do with ixiquarks is probably a result of my own ignorance with the software. Sometimes it is frustrating but most times it takes my improvisations into new territory and I enjoy the learning process.
- Every software has its own philosophy, and so it’s specific lacks which makes it sometimes special and sometimes inspiring. That's why I always look for different software, which gives me new ideas of creating sounds.
- I have other software for more conventional music. It's not an issue.

Further responses to the question in Section 6.7.2.3.9

- Easy after some trying.
- Very easy to understand. Its design is very close to my mode of thinking and working.
- No. All is quite comprehensive.
- On the beginning it was a little bit complicated. But when you start to use it, it is a very

simple structure with very complex instruments and effects. That's satisfying.
- It's very easy and intuitive.
- It is easy. If you come from a background of beautiful interfaces and you are used to hitting a button and making things happen, you will be disappointed.
- It is not like my way of thinking completely, but has elements to it that I find useful.
- Quite difficult, essentially because I'm French
- ixi is very straight forward. Those concepts that are not easy to grasp by looking at it and reading about it are intuitive once you start experimenting with it.
- It is pretty straight-forward. I like devices such as this. So I feel that you and i think alike in the sense that we want a simple means to an end, or if we want, we can get under the hood.
- It's easy to understand, /except/ for routing and node ordering
- I only have a good grasp of these because I know Supercollider, and thus know that model well.
- I thinks ixiQuarks is very easy to understand and very intuitive.
- For me, it is easy to understand and difficult to master, and there are always improvements to be made.
- Ixi is complicated because many things are preset, some things are not. Ixi is a strange mix of theoretical academicness and intuitivity which may indeed work for some.
- Easy after some playing. You were reading my mind doing the tool
- It has a learning curve, for sure, but I think that it is infinitely easier to grasp once one has used some of your simpler software.
- it's easy when I JUST use its interfaces, but if I use them as SC classes then it's difficult
- kind difficult to get in and to understand.
- to some aspects yes. I would need a step-by step guide to get started. Too many things are assumed.
- I think it's amazingly straightforward, considering how "different" it is. It appears that what I do is very much in tune with what the ixiQuarks designers do, since the software encourages experimentation, and I have a very experimental way of making music. I like screwing around with things instead of reading manuals and find out about the functionalities myself, or eventually discover all new ways of use. I also enjoy the fact that the people at ixi seem to advocate free and open source software.
- I feel that the designers are much more technical than I am.
- I like sitting down without too many pre-conceived ideas of what music I'll be making, and this software helps me to do that. I can sit down and fiddle with the instruments and come up with things.
- Easy to understand. i feel that the designers are tapped in on many levels, my favorite is that they are offering their labor of love for free, and are really into feedback.
- I love the fact that there are tutorials and video representation. I would agree to some degree that my way of thinking may match the developers though one may never know. I know to some degree . . . Absolutely.
- I feel like this software was made for me.

Appendix III

Then on-line surveys conducted as part of this research can be found here:

1) On the phenomenology of musical instruments. [section 5.2]
http://www.ixi-audio.net/survey

2) A user survey of ixiQuarks [section 6.7.2]

http://www.ixi-audio.net/content/download/ixiquarks/survey

Surveys for studens that have used ixiQuarks in their education:

http://www.ixi-audio.net/content/download/ixiquarks/survey/angelaruskin.html

Survey 1)

A Questionnaire on Musical Instruments

This survey is aimed at people that are at least one of these: composers, acoustic instrumentalists, digital instrumentalists and/or ixi software users. It is part of our research and will be published in conferences and here online in due time.

The survey is divided into 6 short sections: (and should not take more than 20-30 minutes to fill out) - Personal details - Musical knowledge - Acoustic instruments - Digital instruments - Comparing acoustic and digital instruments - ixi software (if you haven't used ixi software, just skip this section)

You **don't need** to answer all the sections. For example, an instrumentalist that has never worked with a computer doesn't need to answer the computer related questions (although we would be interested in your views). Or a digital instrumentalist that doesn't play a traditional instrument will not have to answer the acoustic instrument part. However, we would like to have as much details as possible.

Feel free to answer in any of the following languages: - English - Spanish - French - Italian - Swedish - Norwegian - Danish - Icelandic - Basque

Personal details

Name (optional)
Email (optional)
Nationality
Gender
Age
Profession/vocation
Academic institution (if any)
Location
Project website (optional)
Where did you hear about this survey?
I would like to be notified of the results of this survey (Email above necessary)

Musical knowledge

How would you define yourself? (select as many as you want)

Musician

Composer
Designer
Engineer
Artist

How long have you been making music?
Do you have a musical education? (if so what kind?)
How long have you used computers in your music?
How would you (or others) describe your music? (as in genre)

Acoustic Instruments

What instrument do you play and how long have you played it? (list several if appropriate).

How would you describe your relationship with the instrument?

Do you sometimes find that it lacks functionality? If so, which?

How much "unstable" or "non-deterministic" behaviour is there in your instrument? Do you feel that you always have full control over it? Can you describe this feature?

How much do you know about the history of your instrument? Do you think it could have been built differently? Are you aware of the ergonomical (or other) issues in the way it is designed?

Do you find that it would be of benefit if your body was different? If so, how?

Have you used a computer for making/playing music? If so, what did you like or dislike about that experience?

Digital Instruments

Which operating system do you use? (tick as many as needed)
GNU/Linux
Mac OSX
Windows

What are the reasons you chose that system?

Do you have a programming experience? If so, what languages have you used and what is your language of choice? Which properties of programming languages do you value?

What machine do you use and which soundcard (if any)?

What music software do you use and why?

Have you tried or do you use any of the following audio programming environments? If so, how often do you use them?

- Pure Data	Tried it	Seldom	Often	All the time
- SuperCollider	Tried it	Seldom	Often	All the time
- ChucK	Tried it	Seldom	Often	All the time
- CSound	Tried it	Seldom	Often	All the time
- Max/MSP	Tried it	Seldom	Often	All the time
- Plogue Bidule	Tried it	Seldom	Often	All the time
- Aura	Tried it	Seldom	Often	All the time
- Open Snd World	Tried it	Seldom	Often	All the time
- AudioMulch	Tried it	Seldom	Often	All the time
- Reaktor	Tried it	Seldom	Often	All the time

What are the positive and negative aspects of the environments you have tried? If one of them is your main tool, why have you chosen it?

Do you use Open Sound Control (OSC)? If so, what do you use it for?

Do you use any graphics software/externals (such as Gem, Jitter, Fluxus, PDP, vvvv, Mirra)? What do you use it for? In what kind of social settings?

Comparing Acoustic and Digital Instruments

What are the biggest differences for you in playing acoustic and digital instruments? (if applicable)

Does computer software lack something that acoustic instruments have? If so, what?

Do acoustic instruments lack something that digital instruments have? If so, what?

Do you have a need for musical software that doesn't exist? What would your dream musical software be like?

What virtual or physical interfaces would you like to see in computer software?

Do you feel that the limitations of acoustic or digital instruments are a source of frustration or creative inspiration? Does it depend on the type of instrument? If so, how?

---

ixi Software

Questions for those that have used ixi software.

Note: We are only asking about our old applications that can be found under the "software" link on our website. Some people have already started using the new OSC applications, but they are not the focus of this survey.

Where did you hear about ixi?
When was that (approximately)?
In what context do you use ixi software?
(home) Studio
Live
Both

Of the following, which of the ixi applications have you used?

SpinDrum	Connector
Lauki	Virus
StockSynth	MicSpace
GrainBox	PolyMachine
Picker	Shell
Slicer	VideoBat

Can you describe the process in which you use ixi software? Do you use ixi exclusively or together with other software?

What are your joys and frustrations when using ixi software? What are the positive and negative aspects of the software?

Do you feel that there is a general design strategy in the design of ixi software? I.e. is it easy to learn a new ixi application when you have mastered one already?

Are there characteristics in the ixi software interface design that define elements in your music? If so, do you find this limiting or creatively engaging?

Are there elements in a specific ixi application that has inspired you? Do you get "emotionally attached" to some applications whilst not connecting to other applications?

Other...

Did we forget to ask a question? Have you got any other comments on the subjects we've been asking about?

Survey 2)
User Feedback on ixiQuarks

The development of the ixiQuarks are part of a research that involves human-machine interaction, philosophy of technology and the culture of software use in music. In return for free software we would like to ask you a few questions regarding your experience with the ixiQuarks. This survey can be answered no matter how often or how much you use the ixiQuarks.

The survey is divided into 6 short sections: (and should not take more than 10-15 minutes to fill out)
- Personal Details
- Musical Knowledge
- General Questions on ixiQuarks
- Experience Using ixiQuarks
- Usability/interaction questions
- Other Environments

You don't need to answer all the sections. However, we would like to have as much details as possible.

Personal details

Name (optional)
Email (optional)
Nationality
Gender
Age
Profession/vocation
Academic institution (if any)
Location
Project website (optional)
Where did you hear about this survey?
I would like to be notified of the results of this survey (Email above necessary)

Musical knowledge

How would you define yourself? (select as many as you want)
Musician
Composer
Designer
Engineer
Artist

How long have you been making music?
Do you have a musical background? (if so what kind? Years?)
What instrument(s) do you play? Years playing?
How much time every week do you spend playing/composing?
How long have you used computers in your music?
How would you (or others) describe your music? (as in genre)

General Questions on the ixiQuarks

Here we ask about your rating of the ixiQuarks as a system. Rate it from 1 (bad) to 7 (good) or click the NA (no answer) radio box.

- System Speed	bad -	- good	NA
- System Reliability	bad -	- good	NA
- Consistency	bad -	- good	NA
- Help and documentation	bad -	- good	NA
- Originality in Design	bad -	- good	NA
- Musical Output	bad -	- good	NA
- Sound Quality	bad -	- good	NA

Experience when using the ixiQuarks

We are interested in the experential (emotional) reaction when using ixiQuarks. Rate it from 1 (on the left) to 7 (right) or click the NA (no answer) radio box.

terrible	wonderful	NA
difficult	easy	NA
frustrating	satisfying	NA
inadequate power	adequate power	NA
dull	stimulating	NA
rigid	flexible	NA
difficult to learn	easy to learn	NA
difficult to operate	easy to operate	NA
difficult to understand	easy to understand	NA

Questions regarding the specific instruments and tools

Here we ask about your rating of the ixiQuarks instruments, utilities and effects/filters. These categories reflect the way they are categorised on the ixiQuarks tool-browser window. Rate them from 1 (bad) to 7 (good) or click the NA (no answer) radio box if you haven't tried it.

- Utilities	1 -	- 7	NA
- Instruments	1 -	- 7	NA
- Effects	1 -	- 7	NA
- Filters	1 -	- 7	NA
- Spectral filters (new in v4)	1 -	- 7	NA

Of the instruments (in the instruments category), how would you rate each one of them in terms of usability, originality of design and fun to use?

- SoundScratcher	1 -	- 7	NA
- StratoSampler	1 -	- 7	NA
- Mushrooms	1 -	- 7	NA
- Predators	1 -	- 7	NA
- Gridder	1 -	- 7	NA
- PolyMachine	1 -	- 7	NA
- GrainBox	1 -	- 7	NA
- BufferPlayer	1 -	- 7	NA
- ScaleSynth	1 -	- 7	NA

Do you have any specific comments on the utilities, instruments, effects or filters? Something you thought about when you were rating them here above?

Questions regarding the general usability/interaction design of ixiQuarks

The software provides enough feedback when used.

strongly disagree strongly agree NA

The words, concepts and visual metaphors in the instruments are familiar.
strongly disagree strongly agree NA

The use of non-musical metaphors in the interface is understandable.
strongly disagree strongly agree NA

The look and feel of the interfaces are consistent within the whole system. (interface design)
strongly disagree strongly agree NA

The experience of using the software is consistent throughout. (interaction design)

strongly disagree strongly agree NA

The software is useable in studio use.
strongly disagree strongly agree NA

The software useable in live performances.
strongly disagree strongly agree NA

The relationship between the controls and their actions is obvious.
strongly disagree strongly agree NA

The ixiQuarks allow for a range of user experiences.

strongly disagree strongly agree NA

The software guides the user sufficiently about its potential.

strongly disagree strongly agree NA

Repeated use of the software results in the feeling of getting better at the software.
strongly disagree strongly agree NA

The design is simple, intuitive, easy to learn and pleasing.
strongly disagree strongly agree NA

The tool is free from irrelevant, unnecessary and distracting information.

strongly disagree strongly agree NA

It is easy to remember how to use the tools.
strongly disagree strongly agree NA

The ixiQuarks inspire creativity.
strongly disagree strongly agree NA

The software is sufficiently documented (with for example help files).
strongly disagree strongly agree NA

It is easy to navigate the tool and start up new instruments and effects.
strongly disagree strongly agree NA

The environment fun to use.
strongly disagree strongly agree NA

Discursive Questions

Roughly how often have you used ixiQuarks? Do you use it regularly? Where does it fit in your workflow? In what context? (live, studio) Do you use it with other software? If so, how?

The hardware setup. What physical interfaces do you use to control the ixiQuarks? (Soundcard, Mouse, Keyboard, Wacom tablet, MIDI, etc)

Describe your overall reaction/experience when first encountering the software.

Do you feel the ixiQuarks environment is different from the commercial products? If so how? Can you think of any software (commercial or not) that focuses on the same areas of interactivity?

Would you say that the software is general or specific? Discuss.

Does the ixiQuarks environment affect your compositional or performative patterns? Do you think differently about your music? Does the environment encourage new directions in your music? If so, how?

Regarding embodiment and bodily control over music: to what extent do you think you use your body to control ixiQuarks? How is it different from acoustic instruments? Or other software?

When using the ixiQuarks, do you often find that the software does not allow you to perform certain basic musical things? What is lacking? If not, are these limitations perhaps inspiring?

Do you find ixiQuarks easy or difficult to understand? To what degree do you feel that your way of thinking matches that of the designers of ixiQuarks?

Are there characteristics in the ixiQuarks interface design that define elements in your music? If so, do you find this limiting or creatively engaging?

Can you describe in few words the most negative and positive aspects of the ixiQuarks environment?

Other environments

What other music software do you use and why?

Have you tried or do you use any of the following audio programming environments? If so, how often do you use them?

- Pure Data	Tried it	Seldom	Often	All the time
- SuperCollider	Tried it	Seldom	Often	All the time
- ChucK	Tried it	Seldom	Often	All the time
- CSound	Tried it	Seldom	Often	All the time
- Max/MSP	Tried it	Seldom	Often	All the time
- AudioMulch	Tried it	Seldom	Often	All the time
- Reaktor	Tried it	Seldom	Often	All the time

What are the positive and negative aspects of the environments you have tried? If one of them is your main tool, why have you chosen it?

Do you have a programming experience? If so, what languages have you used and what is your language of choice? Which properties of programming languages do you value?

Other...

Is there something lacking in the ixiQuarks? What instrument would you like to see as part of the environment? Or some protocols or hardware support?

Did we forget to ask a question? Have you got any other comments on the subjects we've been asking about?

Appendix IV

UNIVERSITY OF SUSSEX

THOR MAGNUSSON
COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE

EPISTEMIC TOOLS:
THE PHENOMENOLOGY OF DIGITAL MUSICAL INSTRUMENTS

SUMMARY

This interdisciplinary research investigates the nature of making creative tools in the digital realm, through an active, philosophically framed and ethnographically inspired study, of both practical and theoretical engagement. The study questions the nature of digital musical instruments, particularly in comparison with acoustic instruments. Through an enquiry of material epistemologies, the dichotomy between the acoustic and the digital is employed to illustrate the epistemic nature of digital artefacts, necessitating a theory of *epistemic tools*. Consequently *virtual embodiment* is presented as a definition of the specific interaction mode constituting human relations with digital technologies. Although of a different nature, it is argued that such interactions are indeed embodied, contrasting common claims that interaction with software is a disembodied activity. The role of cultural context in such design is emphasised, through an analysis of the digital system design's intricate process of analyses, categorisations, normalisations, abstractions, and constructions, where the design paths taken are often defined by highly personal, culturally conditioned, and even arbitrary reasons.

The thesis contributes to the field of music technology with a dissection of digital musical systems, both production and expressive tools, from the perspectives of ontology, phenomenology and epistemology. The nature of embodied interaction with such systems represents an ideal study in music, computer science and the philosophy of technology, due to the strong requirements for efficiency and ergonomics on the one hand, and the systems' highly inscribed theory on the other. From ethnographic work, a philosophical exegesis, and extensive surveys and interviews, a theory of digital musical instruments is derived. The thesis describes how its *practical* outcome, the ixiQuarks musical system intended for live improvisation, embodies many of the theoretical points, exemplifying the nature of affordances and constraints in expressive tools. The research thus investigates the cultural element of systems design, pointing at its political, ethical and aesthetic roles, emphatically concluding that technology can never be neutral.