Chapter 3

HCI in Computer Musical Instruments

"The quality of the interface between artistic intention and the materials of composition is, therefore, a major contributory factor to the outcome of an electroacoustic work. As a consequence it justifies a distinct area of study within the tradition"

[*The Human Machine Interface in Electroacoustic Music Composition*,

M. Vaughan, 1994]

Overview

This chapter focuses on the subject of Human Computer Interaction applied to musical computing situations. A historical perspective is given followed by a survey of literature regarding current interfaces. The chapter concludes with a discussion of several fundamental ways to improve the quality of interaction between musician and computer.

3.1 Metaphors of HCI for musical applications

Computers have been used for making music from the earliest days of digital computing. It is therefore no surprise that the history of the user interface for musical applications closely follows the general design trends for non-musical interfaces of the time.

The following five sections outline a series of metaphors which describe the possible modes of interaction that a musician can undergo when using a computer-based system for musical purposes.

3.1.1 The ‘orchestra-building programmer’

Chapter 1 described how users of the many systems derived from Mathews' *Music* series of programs were made to take on two distinct roles:

i) ‘Algorithmic Instrument Designer’ - in which the user selects pre-compiled code fragments (representing signal processing operations) and connects them together to form networks known as *orchestras*.

ii) ‘Parameter specifier’ - where the user types streams of numerical data into a file known as a *score*. This score is subsequently processed by the ‘player’ program and the numbers passed on at the appropriate time to the orchestral algorithms which produce the sound.

```
sr = 22050
kr =  2205
ksmps = 10
nchnls = 1

instr 19
;p4  : amplitude
;p5  : fundamental
;p6  : rise time
;p7  : decay
;p8  : number of partials

k1        linen     p4, p6, p3,
p7                  ; envelope
a1        buzz      k1, p5, p8, 1
                    ; oscillator
          out       a1
                    ; output
endin
```

```
f1 0 8192 10 1
;                   p3      p4      p5      p6      p7      p8
;instr    start     dur     amp     freq    rise    decay   No. of
;                                                           partials

i19       0         2       30000   40      .1      .1      5
i19       3         .       .       40      .1      .1      6
i19       6         .       .       40      .1      .1      7
i19       9         .       .       40      .1      .1      8
i19       12        .       .       40      .1      .1      10
i19       15        .       .       40      .1      .1      15
i19       18        .       .       40      .1      .1      20
i19       21        .       .       40      .1      .1      40
i19       24        .       .       40      .1      .1      80
e
```

Figure 3.1: An example Csound orchestra (left) and score

The joint effect of these roles is to make the composer into a programmer. People working in this manner are almost indistinguishable, to casual onlookers, from others in the same room using computers for programming in conventional languages such as C, Pascal, BASIC and FORTRAN. The computing tools are based around a computer screen and involve text editors and command-line operations. Only the syntax of the program code is different.

> *“The tactile relationship between the medium and the activity all but disappears. The material is reduced to a common representation as `data' which can only be interpreted via a program and the computer's operating system to the composer, thereby adding another layer to the composer's distance from it.”*
>
> [Vaughan, 1994]

3.1.2 The 'Conductor'

The direction of Max Mathews' research efforts, having spawned a whole new approach to making music, changed dramatically in the 1970s. His new emphasis was on enabling users to 'conduct' music, the score of which is stored on the computer. This change of direction was most probably taken as an antidote to the restrictions imposed on a composer by a textual, non-real-time interface. It should be noted that for composers involved in programming Music-N type orchestras this was not seen as a restriction. On the contrary, many actually expressed the freedom they now had from other people's interpretations and from the sounds available from acoustic instruments.

Mathews, however, felt that the human expression which is found in live performances of acoustic instruments and orchestras had to be harnessed for the new sound worlds of the digital computer. He made it quite clear that he thought it was impossible to specify all the necessary sound parameters in real time. His solution was to pre-store a score (consisting of an ordered set of pitches) in computer memory and allow the performer to conduct this score by adjusting the most important characteristics in real time. These characteristics were usually one or two of the following: pitch offset (or pitch-bend), note volume or playback tempo.

Whilst this approach offers a novel entry point to the art of conducting, the computer is still in charge of the notes being played. The human conductor simply perturbs some of the parameters that are stored in digital memory.

3.1.3 'Graphic Sound and Score designer'

The spread of microprocessor-based computing systems in the late 1970s contributed to the increasing popularity of the Visual Display Unit (VDU) as the primary source of feedback to the user.

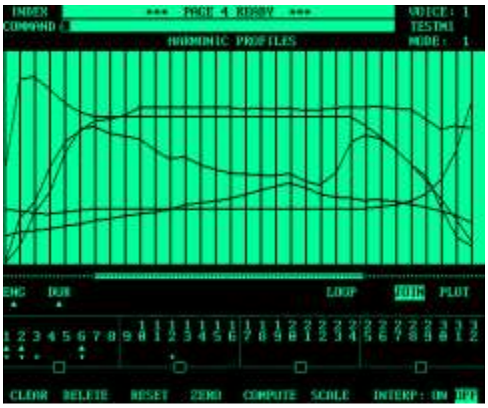

Figure 3.2: The Fairlight's 'harmonic profile' screen which was edited with a light-pen

Data was displayed in a variety of formats (graphs, shaded areas, sliders etc.) making good use of the highly developed human visual-spatial abilities. Even today it is assumed that the primary source of feedback should be visual and consequently a VDU is still a standard part of most computing systems. However, we should note at this point that proficient players of conventional instruments rarely look at the instrument, but rely more on acoustic, tactile and kinesthetic feedback (see 3.4.4)

The Fairlight and Synclavier systems (see section 1.9 and Figure 3.2) popularised the idea of allowing the editing of sound parameters via visual feedback to the user. A mouse, joystick or cursor buttons were used to move a cursor around a screen. Graphical displays of synthesis parameters (or score notation, or even sample data) could be manipulated by the action of the cursor.

Nearly all modern musical computing systems follow this paradigm. It is generally seen to be more 'friendly' and satisfying to use than the numerical manipulation of a set of score parameters. The very popular concept of the graphical MIDI-based sequencer relies upon the combination of visual displays (usually giving the user a choice of editor types) and direct manipulation of notes via a MIDI instrument (see Figure 3.3).

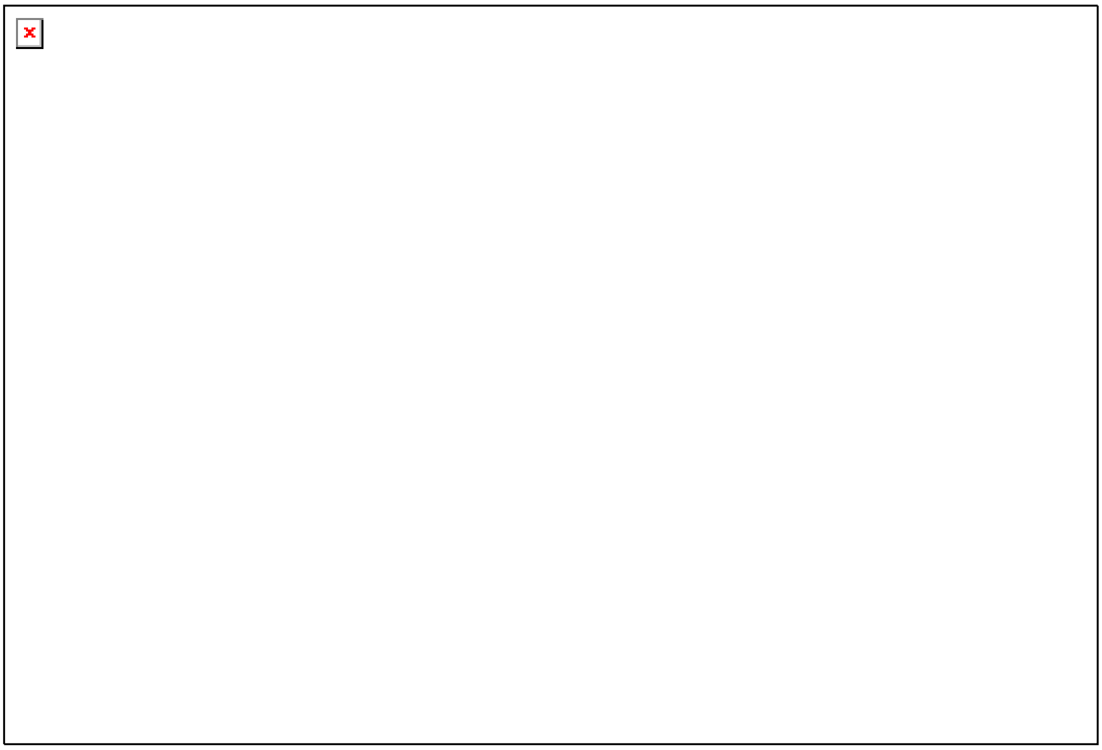

Figure 3.3: Typical MIDI software sequencer interface screen (Steinberg's *Cubase VST*)

It should be noted, however, that not all operations are best portrayed graphically. The more abstract concepts which computing allows (such as text string matching, algorithmic manipulation, recursion and scalability over different hierarchical levels) are notoriously difficult to represent graphically. This is why there is an increased emphasis on systems that involve both visual and textual display and editing.

3.1.4 'Push-button LCD controller'

Another type of user operation appeared with the growth of the digital keyboard industry in the early 1980s; that of adjusting parameters by displaying them on a small embedded Liquid Crystal Display (LCD).

Prior to this, keyboard synthesisers had presented the user with an array of knobs, buttons and sliders for sound control. Each of these controlled a single parameter of the synthesis process (for example filter cut-off frequency, output volume or pitch offset) and thus gave the user direct continuous control of any of the available parameters. The user still had to think in terms of 'parameters' but they were controllable by physical actions which could be learnt. Thus editing was made intuitive and immediately correctable, with the added bonus that the same interface could be used to alter the sound in the course of a live performance.

Since the advent of the digital age the knobs and sliders which gave this directness of control have been mostly replaced by a small parameter screen. The manufacturers put most of their microprocessing effort into the sound generation process and its real-time triggering from the keyboard. A small part of the software is devoted to the user interface. Keyboard players are actively discouraged from editing sounds because of the complexities of navigating a hierarchical menu structure of cryptic alphanumeric parameter values via a 2-line liquid crystal character display and a small pair of 'Up/Down' buttons. This causes the keyboard player to regard the control of sound as intellectually divorced from (and on a different plane to) the directness of triggering the sound by playing the keyboard.

Figure 3.4: The Roland D110 which is programmed by cursor buttons.

This resulted in the affirmation by the commercial music industry that the purchase of pre-made sounds is the way forward for musical creativity, and that sound production should be left to the experts. It is no wonder that a large gulf emerged between the academic community (whose focus was on the detailed creation of new sounds) and the more public music technology industry with its emphasis on 'quick and easy to use' sounds and rapidity of production.

In 1990 Roland brought out a new MIDI synthesiser, the JD-800, which allowed direct control over all its major digital parameters via an extensive array of sliders and buttons (see Figure 3.5).

Figure 3.5: The Roland JD-800 synthesiser

They stopped manufacture within the year due to a lack of sales because ironically users found it 'difficult to use and reminiscent of the old way of doing things'! Such was the entrenchment of ideas that users now expected a small parameter editing screen. This is just one instance of the market dictating what it thinks is best for itself. Manufacturers have no choice but to produce those things which will sell. However, this does not mean we have to accept the current way of doing things as being the best way.

Since Roland's subsequent rejection of the comprehensive slider bank, they and most other keyboard manufacturers have developed higher-resolution displays that allow a limited form of graphical interaction, coupled with a small set of sliders which allow real-time control of a few parameters. This is at least an improvement on the '2-line LCD with Up/Down buttons' interface.

3.1.5 'Instrumentalist'

As can be seen from the historical overview in Chapter 1, much of the design effort for the musician-machine interface is centred on producing better editing and recording options. This is primarily due to the fact that people wish to capitalise on those operations which can only be done easily using digital technology (e.g. *sampling* and *sequencing*). Another reason for this specialisation is that it can be easily achieved using the existing forms of computer interfaces since these devices have evolved for the editing and representation of data.

In contrast, live performance demands strict real-time operation and flexibility of response. Designers of new musical instruments need to work very hard in order to produce a device that gives the user a good sense of control.

Once again the piano-type keyboard dominates the real-time control market. The majority of MIDI instruments are keyboards (as the MIDI specification was effectively designed around the requirements of keyboard players). As mentioned earlier, the keyboard is a good way of triggering polyphonic ballistic events, but not of controlling them throughout time (a requirement of much electronic music).

> *"One of the limitations of real-time musical instruments is their methods for gestural capture. The normal Man-Machine Interface is constructed from a series of switches organised as an electronic keyboard, and can typically provide only two or three control dimensions (time struck, velocity of strike and pressure). Conversely, traditional musical instruments are capable of being controlled in a much more subtle and complex way"*
>
> [Bailey, 1993]

However, the keyboard is clearly not the only form of interface for live electronic music. When new systems (beyond MIDI) are devised, the question of the form of the performance interface becomes even more important. The communication protocols are likely to be more detailed than a series of note triggers, so a keyboard is not likely to be sufficient.

It is the category of alternative live performance instruments which we will explore more thoroughly in the following sections.

3.2 Review of Recent Literature

This section is concerned with the categorisation of contemporary ideas relating to computer-based instruments. A relatively small amount of the literature deals with the specific issues of the *mapping* of the instrument controller onto the internal synthesis parameters. This is a key issue in this thesis, and it is covered in more detail in Chapter 6.

3.2.1 Philosophy and Aesthetics

Several authors tackle the issues underlying the design of new instruments. Mike Vaughan [1994] proposes the increased use of tactile instrumental control and contrasts this with the current requirement to 'think through' the control of instruments via computer terminals.

Simon Emmerson [1994(1)] explains how the advance of technology has gradually removed (or *dislocated*) the audience from the performer, and the performer from the instrument. He later explains [1994(2)] the need to study more closely the area of performance control. Jeff Pressing [1994] takes up this theme, but focuses on the decreasing role of the performer in the light of computer composition systems.

Miller Puckette and Zack Settle [1993], from IRCAM, argue for less 'passive' instruments by the application of system feedback. Their proposed instruments are still played by human performers, but also have 'a life of their own'. Andrew Gerzso [1992] attempts to categorise the use of computers in music to three basic paradigms (Text-control, Graphics Programming and Sequencing) which, strangely, do not include live performance.

Mari Kimura [1995(1)] describes the problems personally encountered in the live performance of electroacoustic music.

> *"The degree of excitement caused by new inventions that can "do anything" functionally surprises me, especially in comparison to the lesser interest in mastering these new devices"*
>
> [Kimura, 1995(2)]

Todd Winkler [1995] examines the use of physical gestures to control different levels of musical structure in interactive systems. He cites the Theremin as an instrument from which we can learn much about design and movement. He urges designers to consider the energy of a performer and to build instruments which capture his or her movement. Choi et al [1995] also discuss the possibilities of controlling a multidimensional instrument in real-time. Their *manifold interface* is an attempt to allow musicians to explore complex control spaces with auditory and graphical feedback.

Lonny Chu [1996] urges designers to include 'haptic' (sensory and force) feedback into new computer musical instruments. He notes that little work has been done on the cognitive effect of *not* having any haptic feedback on a musician's performance. Plans are outlined for a force-feedback MIDI control device.

A similar theme is taken up by Anna Christiansen who stresses that the crucial part of interface design is in the mapping of human actions onto the domain of response of the computer. She notes that:

> *"The performer is no longer a cognitive system relying on mental representation, but a multi-sensory system whose mental and behavioural act is emerging in context."*
>
> [Christiansen,1996]

3.2.2 Acoustic Instrument Adaptations

Many performers wish to retain the performance control they have spent years building up on acoustic instruments and yet use this to control new sounds. Consequently, many instrument designers are producing customised 'add-ons' to acoustic instruments which act as an interface mechanism for synthesiser control, usually via MIDI. Examples of such adaptations are Perry Cook's brass instrument control system [1993], the 'Meta Trumpet' [Impett, 1994] and 'Trombone-Propelled Electronics' [Collins, 1991] (see Figure 3.6).

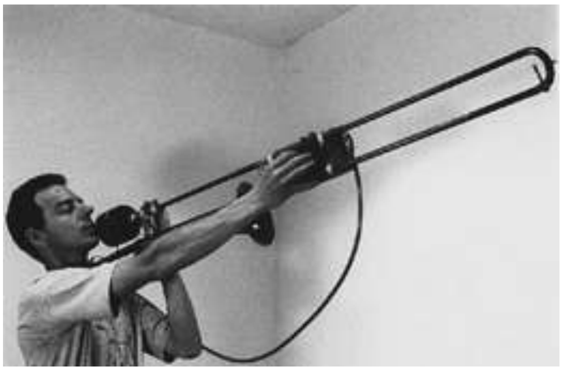

Figure 3.6: Nick Collins' *Trombone-Propelled Electronics*

Risset [1996] describes his use of an acoustic piano which has been adapted so that the computer can not only sense what keys are pressed by the player, but can also remotely control the notes. The computer acts as a musical partner, responding algorithmically to what the pianist plays. All sound is made by the acoustic piano.

Greg Schiemer [1996] describes how the sound of his 'Carnatic Violin' is processed by the A4 MIDI Tool Box to provide audio delay and the control of several MIDI sounds in live performance.

Bart Hopkin [1991] supports the development of new acoustic instruments because of the physicality of playing them and the richness of sound found by exploring physical objects. He hosts a web-site [Hopkin, www] dedicated to experimental (mostly acoustic) instruments. He is associated with Will Menter, who makes new instruments out of a variety of materials. Some of these can be seen in Figure 3.7.

Figure 3.7: Will Menter's Karimba (left) and Slate Marimba (right)

3.2.3 Performance Environments

The term 'Performance Environments' is used here to describe systems which are intended for live performance, but are larger than the concept of one instrument. Computers offer the opportunity of processing the movements and inputs of several performers and using these signals to jointly control a sound source. This type of configuration has been termed a *HyperInstrument* by Tod Machover [1989] and Tim Anderson [1994].

Examples of such environments are 'Virtual Performer' [Kanamori, 1993], and an 'expression tool' which uses a neural net to analyse human gestures to activate sound [Hartono, 1994].

Mark Bromwich [1995] describes the implementation of the multimedia theatre performance *The Navigator*. It centred around the *MidiCreator* system (see section 1.13.4) which was used to convert the outputs of several specially constructed interfaces into MIDI and then into sound.

The interfaces included *Velocity Wands*, a *Cosmic Instrument*, joystick controllers (see Figure 3.8) and light sensors embedded into a series of objects on stage. The performer activated music and sound effects by opening and closing 'books', dancing and manipulating the various objects on stage.

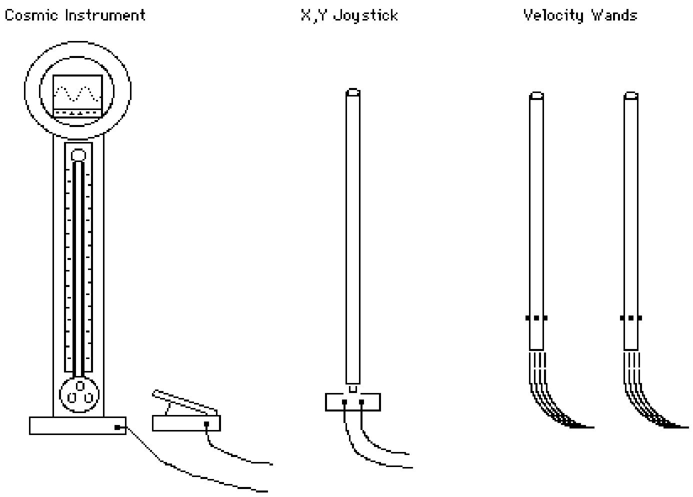

Figure 3.8: Mark Bromwich's instruments for *The Navigator*

Several authors describe systems for sensing the position and footsteps of dancers, notably Pinkston [1995] and the *LiteFoot* system by Griffith [1998].

Chen describes a method of controlling sound sources from video images of pre-recorded films and live performers wearing reflective costumes [Chen, 1994]. Other environments include stage-mounted detectors for triggering multimedia compositions [Povall, 1993].

3.2.4 Conducting / Real-time Control

Systems where a performer interacts with a computer-based note-selecting algorithm or pre-stored score can be considered to be in the same category as Max Mathews' 'Conductor' program (section 3.1.2). Systems of this type are considered by some to be 'instruments' (hence their inclusion in this review), but the presence of a pre-stored series of events or event-generating algorithm implies that they place the performer into a role closer to the traditional model of a conductor than that of a solo instrumentalist.

Examples of such systems are the 'Blues-o-matic' [Garton, 1993] where a data-glove is used to influence the outcome of a computer-generated blues guitar solo, the Computer-Extended Ensemble [Jaffe, 1994] and 'Ensemble Performer' [Grubb, 1994] which can interact with many musicians by 'following' each individual solo line.

Several systems are intended to track human conducting motion by gestural extraction. An example of this is given by Tobey [1996] where musical expression is extracted by following a conducting baton in three dimensions of movement. Teresa Marrin [1997] explains how the *Digital Baton* is used to replicate the traditional conductor's baton, then later [1998] presents the "Conductor's Jacket" which extracts gestural data from the body of a performer (see Figure 3.9).

Figure 3.9: Teresa Marrin's *Conductor's Jacket*

3.2.5 Bio-Feedback

Some researchers are working on ways of using the body's electrical signals and brain patterns as a means of generating live control signals.

Tanaka describes the *Biomuse* system (Figure 3.10) which senses electrical currents in the body and processes the control signals via the *Max* program, thus generating MIDI messages to control a synthesiser [Tanaka, 1993].

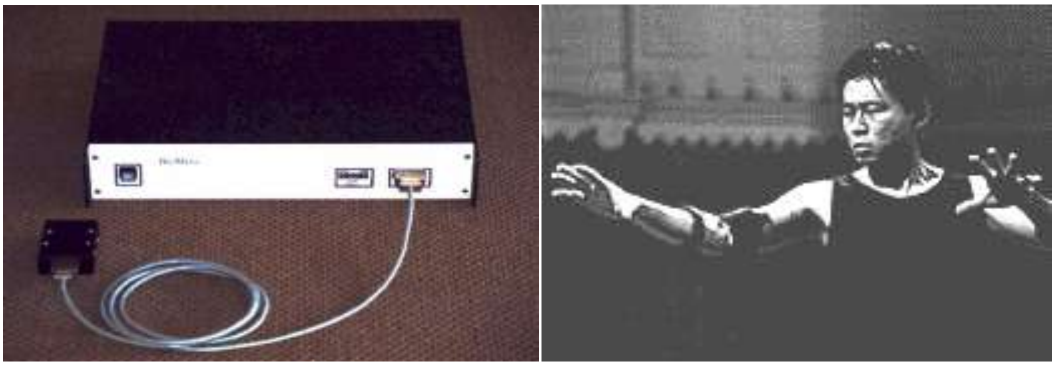

Figure 3.10: The *Biomuse* system (left) and in use by Tanaka (right)

3.2.6 Environmental Control

This category of devices is included here because they are presented in literature as if they are `musical instruments for human performance'. The common factor in these devices is that they exist as 'installations', often involving sculpture and light, and producing sound by reacting to changes of conditions in the environment (such as light-levels, temperature, wind-speed etc).

Examples include the ‘Airplayer’ series [Armstrong, 1991] and the ‘Skyharp Interactive Electroacoustic Instrument’ [Allik, 1993] (see Figure 3.11).

Figure 3.11: The *SkyHarp* installation of ‘wand’ sensors in an outdoor location

3.2.7 New Instruments

This category is concerned with new computer instruments that have been built for real-time performance, with the main emphasis on allowing the player to generate notes/sounds (as opposed to the ‘conducting’ category described in 3.2.4).

Jorg Spix [1994] describes various implementations of a digital *Trautonium* based on the original 1930 version.

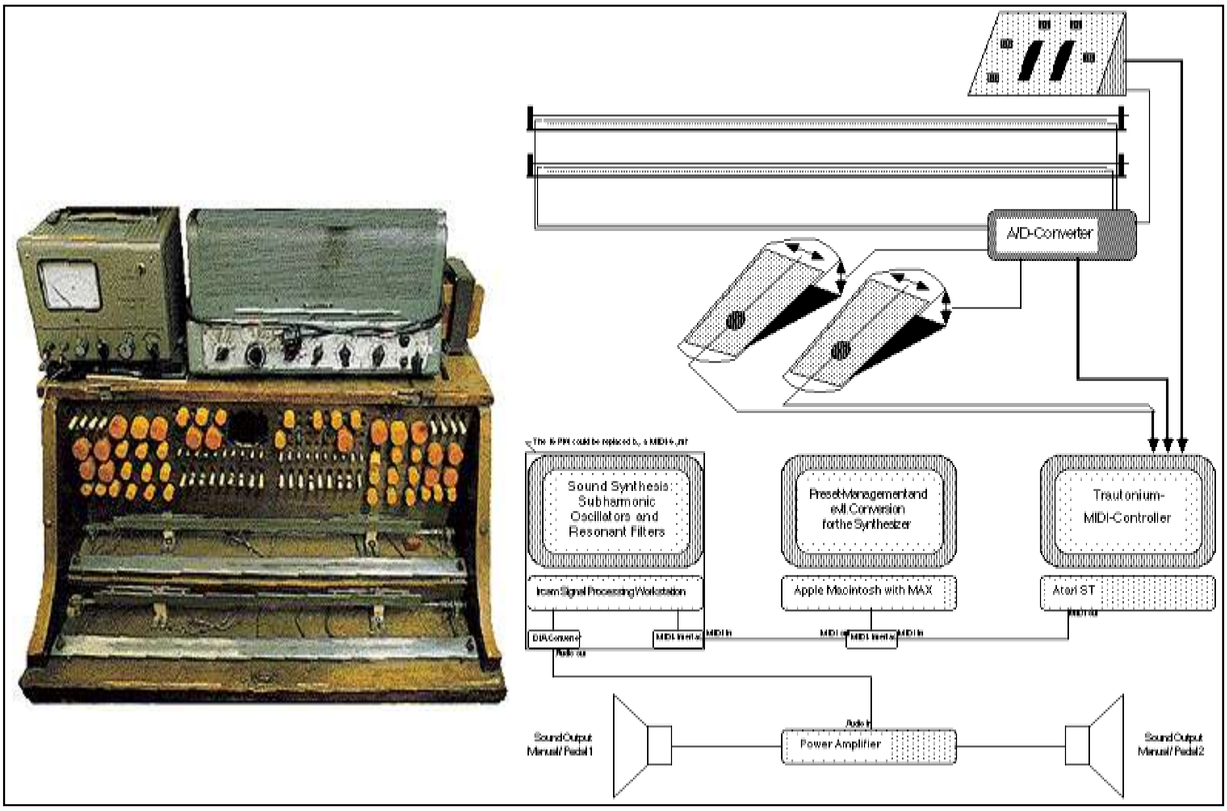

Figure 3.12: The original *Trautonium* (left) and the Digital version (right)

The playing interface consists of two resistive wires, the output voltages of which vary according to where they are pressed and with what pressure. The player can also control the harmonics of the tone generator with the two foot pedals.

The *LightHarp* (Figure 3.13) and *LaserLyre* are two instruments which sense the performer's proximity to light dependent resistors (LDR's) [Favilla, 1994].

Figure 3.13: The *LightHarp*

They provide a non-contact form of playing and plenty of visual feedback. Favilla [1996] explains how the control gestures often have to be mapped to the synthesis parameters in a non-linear manner in order to make them playable in live performance. We return to this discussion in Chapter 6.

The *aXi0* MIDI Controller uses a chord keyboard, coupled to a joystick and switch array to provide multi-parametric input to a synthesiser [Cariou, 1994]. The chord keyboard is a way of selecting a large range of pitches with one hand. The specially constructed joystick allows simultaneous control of many synthesis parameters such as sustain, portamento and pitch-bend.

Roel Vertegaal [1994] compares different physical input devices for a real-time representation of timbre-space. He compares the results of the timbre manipulation tests using a mouse, relative joystick, absolute joystick and data-glove. The results seemed to show that the mouse was the most consistent device (but we would have to question the familiarity of the operators with all four input devices). Vertegaal and Ungvary [1996] emphasise the importance of physical tension in the performance of a piece and its appreciation by the audience. They highlight the main sources of feedback for performing musicians as being *tactile* (sensed on the surface of the skin), *kinesthetic* (sensed internally by muscle and other receptors) *and visual*. They note a special category of controller as being *isotonic* where movement is necessary to operate it.

Xavier Chabot [1993] also introduced a data glove and the 'Sonar System' as methods of gestural parameter control. Chabot is another of the authors who does regard as important the translation (mapping) of human gesture to the final sound.

Haken describes *The Continuum* (Figure 3.14) which is a keyboard designed to allow continuous control of pitch (movement in the x-axis), loudness (pressure in the z-axis) and timbre (position in the y-axis).

Figure 3.14: The *Continuum*

He explains how it is difficult for a performer to control these three variables at the same time.

> *"Continuum performers will probably need to practise for many years to become proficient."*
>
> [Haken, 1998]

Bert Bongers [1994] describes several of the interfaces developed at STEIM, such as the 'Hands' and the 'Web' (mentioned in section 1.13.3), various glove-based controllers based on the Mattel PowerGlove, microtonal keyboards and the importance of tactile and force feedback in instruments.

Bongers [1998] has also interviewed the members of *Sensorband* - a group of musicians who improvise with gesturally-controlled electronic instruments. Their instruments consist of *Biomuse* (see section 3.2.5), *Soundnet* (a huge construction which is played by the performers climbing on it, see Figure 3.15) and a series of sensors that measure the position and speed of the performer's arms. The software 'heart' of the system is a set of patches on the *Max* software.

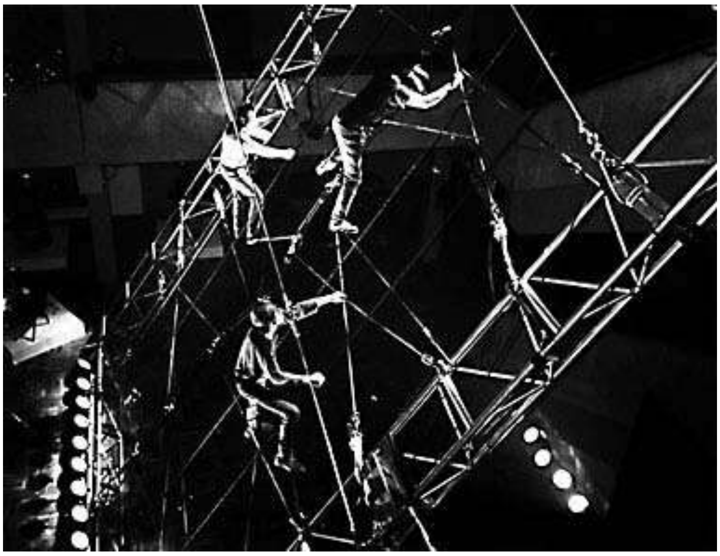

Figure 3.15: *Sensorband*'s climbing instrument; The *Soundnet*

Another device which uses *Max* as the processing tool is the *Meta-Instrument* [Laubier, 1998]. It is an ergonomically designed audio-visual instrument which can operate a large number of variables in real-time. The performer positions his hands and elbows on two pairs of pads, each of which is pressure sensitive. There are also two foot-pedals and a suite of touch sensitive keys for the fingers.

Don Buchla has invented a variety of instruments over two decades, with perhaps his most famous being the *Thunder* (see Figure 3.16). It was an ergonomically designed set of 36 sensors which could be used to control several simultaneous MIDI streams.

Figure 3.16: Don Buchla's *Thunder* MIDI controller

The author's own experiences of designing a new musical instrument interface (MidiGrid) are now described in more detail, as these were a motivating force behind the development of the ideas expressed in this thesis.

3.3 MidiGrid

The MidiGrid project [Hunt, Kirk 1988] was an experiment to investigate the design of a new computer-based interface to electronic tone generators. It was started at the University of York UK in 1987, and has been developed in stages since then [Hunt, Kirk 1990 and 1994]. MidiGrid has been used by a wide range of people (composers, schoolchildren, and special needs teachers and their clients) and its use has raised several important issues relating to the design of interactive musical systems. It was, in fact, the original motivating factor in the development of the MIDAS system (sections 1.14.2 and 6.5) since the authors wished to construct a test-bed for interactive musical instruments and their interfaces in order to explore those areas of interest raised by the use of MidiGrid.

Full details of the system and its operation and application areas are given in Appendix F, but a summary of its key features is now given, followed by a discussion of the issues that arise from its use.

3.3.1 Brief overview of the MidiGrid system

MidiGrid is a graphical computer-based performance instrument. It allows the user to access pre-stored musical material in a variety of modes of operation. The most used and original of these modes is the grid-based *performance screen*.

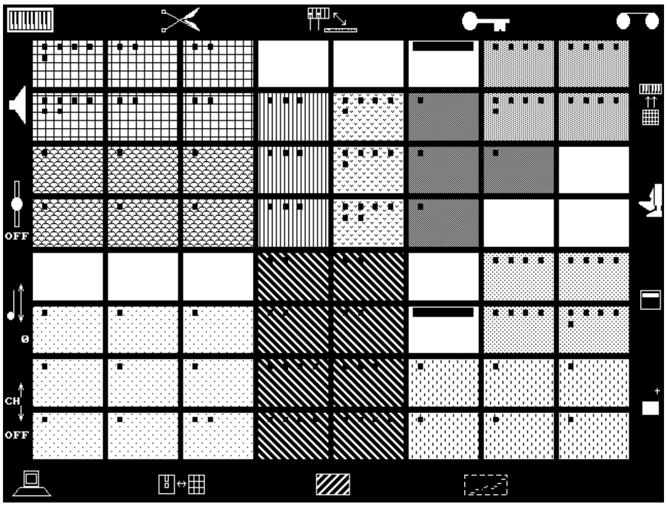

Figure 3.17: MidiGrid's main performance screen

The grid is of a user-definable size and shows the musical contents of each box by simple graphical representations of notes and sequences. The player sweeps a cursor (using the computer mouse) around the screen and notes are triggered when the mouse buttons are pressed. Consequently identical gestures produce identical musical results, but these gestures have to be learnt and rehearsed by the player.

The system uses the MIDI protocol for controlling notes on an external synthesiser or sampler. Versions of the software since 1990 have allowed a variety of MIDI continuous controllers to be sent in response to mouse movement. The effect of this is to permit much more subtle forms of musical control using gestures to bend the pitch, sweep the panning position and swell the volume, using the same gestural input to choose and control the notes.

3.4 Important issues raised by MidiGrid

Several issues have emerged from personal experimentation with MidiGrid and from watching others performing with it. The following sections highlight the five major topics which will be useful later in this report.

3.4.1 Learning an Instrument

Players of musical instruments have always required considerable dedication and commitment to hard work and rehearsal in order to learn how to play well. During the difficult times, particularly at the start of this process, it is the inspiration of watching an accomplished musician perform on the instrument that provides the motivation for continuing to practice for long periods of time.

> *"A recent study showed that the best students at a conservatory had accumulated over 10,000 hours of formal practice by the age of 21"*
>
> [O'Neil 1994]

This model of long-term human learning does not seem to enter into the requirements for learning a new computer interface. The goals of HCI research tend to be focused upon ease of use (section 2.6.2), resulting in the belief that interfaces which are not fully operable within a very short period of time are inappropriate.

This should not be interpreted as an argument for making interfaces difficult, but rather as a plea for a more careful examination of what humans can adapt to. Nobody would accept a 10-week training scheme in order to operate a high-street cash-point machine. Such machines are successful because they are relatively easy to use and allow only a restricted set of operations. On the other hand, nobody would expect a learner driver to be competent after 10 minutes of training! It is taken for granted in *non*-computer-based interactive control systems that the human operators have to learn to adapt to their new control environment - a task that will take much time, effort and individual tuition.

Therefore we should assume that if a computer interface demands more than a surface level of operation, users should be expected to spend long periods of time learning the dynamics of how to 'drive' it. Many computing interfaces are, however, based on the assumption that users do not need to learn it, since they navigate the menu system and interpret accordingly each time they want to access a certain function.

> *"A conventional acoustic musical instrument is an inanimate object which relies on the innate 'processing power' and creativity of the human player in order to bring it to life. However, it is the nature of this intimate interaction between performer and instrument which is the very essence of the performance. There must be a radical shift of thinking in the design of computer systems if computer instruments are to be accepted as long-term, worthwhile tools to which people will be willing to dedicate a large amount of time and effort."*
>
> [Hunt, Kirk 1994].

3.4.2 Configurable Instruments

MidiGrid can be customised by the user to produce individual grid patterns of different size and complexity, containing whatever layout of performance material is required. This flexibility of configuration has been responsible for MidiGrid's successful use in schools and for various Music Therapy situations. Teachers and therapists can devise, restrict or expand the musical material that is available to the end-user. In doing so they form customised musical environments where the tonality, instrumentation and physical layout of the notes (and thus the type of hand gestures used to play them) are defined for a particular music/client combination.

However, there is a danger that players will never learn to control the instrument beyond a surface level of exploration because the 'goalposts are constantly being moved'. Players of traditional acoustic instruments undergo a good deal of configuration themselves in the process of learning to control their instrument!

Generally, if we allow system interfaces to be continually reconfigured, we are perhaps in danger of removing any reason for human operators to work hard at learning to control the system interactively.

3.4.3 Scalable Control Intimacy

Some individual MidiGrid configurations (grid patterns) have been used, without re-customisation, in vastly different activities. One particular pattern (an array of harp sounds) was explored with intense interest by a 2-year old child with severe brain damage and later that same day was used by an electroacoustic composer for an interactive performance.

Each player was exploring the contents of the grid at a very different cognitive level. The child (who was also blind) was probably not even aware she was using a computer or holding a mouse. She simply moved a small plastic box, listened to the harp sounds, smiled and then tried to repeatedly play the sounds she liked best. Changes of her hand speed, which were difficult for her to achieve, were rewarded with more sounds. The electroacoustic composer was using the pre-arranged grid pattern to explore novel streams of notes. The mouse/grid arrangement allowed him to explore note patterns which were almost impossible to play on a piano-type keyboard. He followed a paper ‘score’ which he had pre-arranged to remind him of the best trajectories around the screen.

The above example shows that control intimacy can occur at different levels (and can involve different operations) depending on the user and the situation in which the system is being used. **A good real-time system interface should therefore allow *scalable control intimacy* for individual users with individual needs.**

For the purposes of this study the following definition will be formulated:

A system is exhibiting 'Scalable Control Intimacy' if a user's confidence, accuracy and flexible physical control of a dynamic system can increase over time, without the system having to undergo any step-change reconfigurations.

This definition contrasts with many computer programs which either:

i) cater only for one level of user (e.g. 'beginner' or 'expert'), or

ii) allow a complete change of 'mode' to move from one level of user to another. An example of this is an electronic 'mailer' program which permits a change of mode from 'verbose' (i.e. for beginners) to 'terse' (for experts).

Therefore the goal of being 'simple and easy-to-use' is only valid for those systems where operators will only ever be beginners (in terms of the amount of user practice time) for example the users of cash-point machines. However, the goal of complex real-time system design (such as musical instruments, vehicle or industrial control) should be to accommodate scalable control intimacy by its operators.

3.4.4 Necessity of Graphics in Live Performance

One aspect of a developing control intimacy shown by a traditional instrumentalist is a decreasing reliance on visual cues.

> *"Novice users of MidiGrid frequently request that material be annotated so that they may remember the location of material within the grid. This is analogous to the labelling of a piano keyboard with the letter names of the notes on the stave. Observation of competent pianists will quickly reveal that they do not even look at their fingers, let alone any annotation which may be associated with the keys".*
>
> [Hunt, Kirk 1994]

As users develop their musical performance ability on a particular instrument, they rely increasingly on tactile, audio and kinesthetic feedback, and less on graphical information. There is a general lesson here for the designers of human-computer interfaces in high-performance systems; graphics are a useful way of presenting information (especially to beginners) but are not necessarily the primary channel which humans use when they are fully accustomed to the system.

3.4.5 'Performance Mode'

In the example mentioned in section 3.4.3 (the two-year-old child and the electroacoustic composer), each of the performers explored the grid patterns at a level which was appropriate to them. There is no 'dialogue' between user and computer, instead the computer responds instantly to the user's hand movements. It is then up to the users to interpret the sound produced and to use this as feedback to whatever process they are currently involved in. The computer does not set the agenda or dictate the conversation or insist the users select from a set of predefined options, but instead provides an environment for creative exploration.

This mode of operation is very different to the conventional means of communication with a computer. Traditionally the software is there to gather data, and often does so by dominating the interaction. Even in those situations where the user is fully in charge of the interaction, it usually takes place at a certain level of language ability (for example, the need to read, interpret and take action on hierarchically arranged menus).

3.5 Summary

The advent of digital computer instruments has given birth to many new modes of user interface operation. Interfaces are often designed for the convenience of the engineer, and not necessarily for the user. This chapter has examined the literature pertaining to electronic musical instruments. There are many creative instruments being designed, but there is room for improvement in the ways in which the instruments dynamically respond to the user’s control. This is not necessarily any longer a question of the performance speed of the hardware and software, but more a question of the mapping between performance gesture and instrument response. Too often a dynamic system is hindered by the lack of an interface tailored to the real-time control capabilities of a human user.

This thesis therefore proposes an investigation into the design of computer-based interfaces which allow the user to enter ‘performance mode’. The aim of this mode is to allow users to explore an environment and discover relationships for themselves, getting a real-time 'feel' for the system. This will have benefits in the flexibility, confidence and accuracy of system control.

Chapter 4 considers this mode in more detail and explores how the human mind can learn about different interfaces in very different ways.