Chapter 2

All Computers are Interactive: But Some Are More Interactive Than Others.

> "One must take computers into account, and take them to task, because it is a response to the technology of our time, to the situation of our time. To make music with the technology of our time, and specifically the computer, poses a tremendous challenge for the artist. To address this challenge, in itself, will help keep music alive and significant. To address this challenge in a way that acknowledges, directly and deeply, the human production of that music brings together into a new art form the diverse elements of performance, with its millennia of history, and the age of the computer, with its bare decades of history. . . . interactive computer music takes the fullest advantage of the ideas and technologies of today and unites them with a vision of what they could be." - Garnett (2001), p.31

As outlined in the introductory chapter, the ultimate practical aim of this body of work is to develop upon existing digital music systems which bring us the sense that we are "playing with", rather than "playing on" our computers: to be able to invite the laptop onto the stage and pursue a man-machine collaborative improvisation. Acoustic instrumentalists might talk about their relationship with their instrument as a form of collaboration, but digital instruments have the potential to play a more literally *active* role. As a starting point then, this chapter provides an overview of current approaches to interactive computer music and questions how digital instruments of various kinds have affected our understanding of interactivity itself.

Interactivity has been a buzz word in New Media Arts for decades. Nearly 30 years ago Kay and Goldberg (1977) recognised the significance of the active nature of the computer as an artistic medium. Since then researchers in the field of Human Computer Interaction (HCI) have been exploring the ways in which we interface with digital technology, artists have been developing interactive artworks and installations and more recently critics of New Media Art have begun to construct a framework for appreciating the aesthetic implications of interactive art. Indeed there has been so much activity under the rubric 'interactivity' that Manovich (2002), suggested that the concept has become "too broad to be truly useful" (p.55).

In some respects, the impact of a new dynamic, active medium was less alien to musicians than to other artists working in static visual or plastic arts: music is quintessentially temporal, and instruments inherently interactive. In fact some critics have offered the musical instrument as a model of interaction for the rest of New Media Arts. Although unqualified in this case the term 'instrument' is not a simple concept. Understanding the impact of technological advances on the nature and role of the instrument is central to appreciating the evolution of interaction.

Long before real-time interaction became a possibility, the musical potential of digital technology had been exploited, creating sound worlds which would have made the futurists weep. (e.g. Chowning (1973), Xenakis (1971b), Stockhausen (1964)). For these pioneers, their instrument was the studio. Meanwhile the use of live electronics in performance by Cage and other experimentalists, notably Gordon Mumma began to alter the traditional roles of instrument, performer and composer in performance. Even before digital technology entered the musician's world the term 'instrument' no longer referred solely to a passive device requiring a manual action to create each sound. Instruments now encompassed explicit temporal structure which had previously been the sole reserve of composition.

Composers such as David Tudor and Gordon Mumma have used the term 'composed instrument'. The term has also discussed by Schnell and Battier (2002), to highlight the fact that computer systems used in musical performance "carry as much the notion of an instrument as that of a score" (p.1). Computers can be used to predetermine aspects of a musical work as much as they can be used to realise it in performance. The concept is equally applicable to electronic instruments by virtue of the fact thay they are dematerialised. There is no longer a fixed or direct correspondence between the interface and sound production mechanism. Electronic and digital technologies make it possible to conceive of a sound producing device which is independant of its gestural control.

To give a simple example, a piano responds with a single note when a single key is pressed: there is a one-to-one correspondence between the player's action and the sonic output. We could say that there is an isomporphism between the gesture and the rhythmic and pitch content of the emitted sound. With wind or string instruments, a similar isomorphism exists, and we are in addition aware of a more obvious match in timbral terms. For example the speed and depth of hand or diaphragm movement corresponds directly to the rate and depth of vibrato in a string or wind instrument. On an analogue or digital synth, we are most likely to be presented with a physical or graphical knob, slider, or number box which we adjust in a single movement to create an ongoing change in depth and/or speed of modulation. The differences become more obvious when we consider a simple harmoniser or arpeggiator, or a Max/MSP patch loaded with a rhythmic sample - one simple gesture (a button press) unleashes a potentially endless stream of structured musical material.

The decoupling of interface and sound producing mechanism has great impact from the performer's perspective and, as will be presented in Section 2.1.2, the development of controllers that facilitate expressive control of the sonic output has become a major topic of research. The independence of gesture and sound production also means that physical gestures are not tied to particular sonic gestures but can be re-mapped at will. This is achieved in modular analogue synthesisers for example, by re-patching modules with physical patch cords. This was a trick explored by early electronic composers. Morton Subotnick, for example, applied an envelope follower to his vocal utterances, producing voltages changes which were then used to control a Buchla analog synthesizer (Winkler (2001)). This process was developed into one of the early multimedia operas *Ascent into Air* (1983).

If electronic instruments began to blur the boundaries between instrument and composition, and patch-able electronics demonstrated the effects of de-coupling the interface and sound engine, then the unique aspect of the *programmable* digital instrument goes further and facilitates not only 'composed' instruments with structure, but active, responsive properties that we would traditionally associate with the performer. The aim of this chapter is to examine how these properties have affected the model of interaction that underlies the design of digital instruments, and how this could be developed and realised.

The first section of the chapter outlines and illustrates the principle approaches to interactive computer music systems, bringing attention to the underlying metaphors on which models are based. The model of one-way interaction with an acoustic instrument is contrasted with an alternative two-way, conversation model that has been proposed by those who feel that one-way reactivity does not exploit the full potential of the digital medium. In search of some examples of this conversational form, a brief survey of some examples from interactive Alife-inspired installation art is given in Section 2.2. Section 2.3 begins by contextualising the one-way and conversation models of interaction in terms of the design protocols that subserve them. This is developed further by introducing the philosophical concepts of autonomy and heteronomy which illuminate the implications of adopting either model. In turn, these perspectives are allied with two contrasting computer science paradigms which have quite different methodologies. Consideration of these philosophical perspectives and methodological issues helps in developing a more solid understanding of these metaphors of interaction, but more importantly offers suggesting a set of conceptual and practical tools to aid in the implementation of interactive performance software that begins to realise this model. These are considered in Section 2.4.

2.1 Interaction in Live Computer Music

Almost all newly observed phenomenon are initially described using combinations of concepts drawn from phenomenon that we already understand. This is as true of new technologies as it is of natural world phenomenon: shortly after the invention of the telephone, people discussed the possibilities of broadcasting concerts directly into homes; the first motion pictures were shown in theatres as a backdrop to the actors on stage. Rapidly of course, the telephone and film came to be understood on their own terms and even provided metaphors for the next generation of technology or scientific understanding. Arguably the modern personal computer has not yet reached a similar level of maturity. This is well illustrated within the interactive computer music community, where digital systems are commonly conceptualised in terms of the elements of classical musical culture: virtual instruments, virtual performers, virtual composers, virtual improvisers, virtual listeners, virtual critics (Winkler (2001)).

Variations on these positions can be seen implicitly in various researcher's definitions of interactive music. Robert Rowe (1992) seems to have some kind of dynamic instrument in mind when he writes that interactive music systems are "those whose behaviour changes in response to musical input" (p.1). Other discussions reflect the changing musical roles associated with interactive music: "Interaction has two aspects: either the performer's actions affect the computer's output, or the computer's actions affect the performer's output." (Garnett (2001), p.23). Increasingly, however, there is a sense that this 'one-way reactivity' is not enough, that interactive systems should strive to amalgamate the characteristics of all of music's tools and personnel – the sound of an instrument, ears of an instrumentalist and mind of a composer – such that the musical flow is mutually influential. Winkler hints at this in the opening of his book with a conversational analogy:

> "*Interaction is a two way street.* Nothing is more interactive than a good conversation: two people sharing words and thoughts, both parties engaged. Ideas seem to fly. One thought spontaneously affects the next." - Winkler (2001), p.4

More recently Bert Bongers (2006) has extended this metaphor, stressing the point that interaction should involve a mutual influence which causes both partners in the discourse (whether machine or human) to have changed state, frame of mind or views after the interaction. The reciprocal nature of conversation captures the sense of interaction which we observe in fluid musical situations between two or more human performers, or indeed between any entities in the living world.

2.1.1 Metaphors from Traditional Performance Practice

In one of the earliest texts on interactive music systems, Robert Rowe (1992) proposed a classification of interactive computer music systems along three principle axes. Taking the central concepts of classical western performance practice, he made the distinction between composed and improvised (score-driven versus performance-driven), methods of composition (*transformative*, *generative* or *sequenced*) and between what he calls *instrument* or *player* paradigms.

Todd Winkler (2001) later expressed the need to establish new models of interaction in order to inspire the development of digital performance systems. Attempting to answer key questions such as *What role does the computer play?* and *What is the relationship between computers and humans?*, he suggested that consideration of the interactive relationships which occur in traditional performance ensembles may be a useful starting point to evolve "new modes of thought based on the computer's unique capabilities" (p.21). Focusing on the issue of control (who is in charge, who follows, who leads?) Winkler offers four models based on different types of ensemble and their associated musical idioms.

The Conductor Model (a la Symphony orchestra) describes the situation where the performer acts as conductor, influencing the computer's delivery of a pre-scored part. The earliest interactive music system GROOVE[1], developed at the Bell Labs, operates within this framework. The performer adopts the role of conductor, controlling the tempo, dynamic level and balance of a computer programmed with a pre-written score. This model epitomises a common approach to interactive computer music research known as *score-following* which is described in more detail in Section 2.1.3.

The Chamber Music Model (a la String quartet) proffers a richer model of interaction, alluding to the mutual influences between several players. "In a string quartet . . . the interplay between musicians demonstrates shared control. Intonation, phrasing and tempo are constantly in flux" (p.25). As an example of this model in interactive composition, Winkler offers the first movement of his *Snake Charmer* for clarinet and computer. The interplay here is achieved by explicitly switching control. The piece opens with a computer introduction set at a fixed dynamic level. When the clarinet enters, it is able to influence the dynamic level of the computer line for a short time, after which, the computer stops listening to the performer and continues on its own. Control switches several times during the performance, with the computer exerting most obvious influence over the clarinettist in its occasional outputs, which demand the player increases his dynamic level to match the computer.

The Improvisation model (a la Jazz Combo) outlines a more complex model of interaction, reflecting the fact that in traditional Jazz combos, not only do musicians mutually influence the interpretation of the head (i.e. the scored motifs), but commonly whip each other up into frenzied improvised solos. Winkler comments that "what makes these relationships function to produce music that does not sound like a random babbling is that there are a huge number of shared assumptions and implied rules based on years of collective experience" (p.26). This musical intelligence, as he calls it, is typically simulated with software, which contains both sets of analysis tools to recognise patterns in rhythm, melody and harmony, and generative components consisting of coded sets of rules and assumptions which respond according to the outputs of the analysis module.

---

[1] Generating Realtime Operations On Voltage-controlled Equipment

Free improvisation (a la AACM, Down town NYC, European free improvisers) is deemed to be the most challenging and complex model for interactive music. The spontaneous, expressive and unpredictable nature of free improvisation, evokes a much richer notion of interaction and demands software which allows more freedom. Such systems typically combine analysis components with a processing engine which has considerable generative power. These engines draw from a variety of AI methods such as GAs, markov modelling, directed graphs, rule based systems as well as 'extra musical' models, all of which will be discussed in Chapter 3. The aim is not only to turn the computer into an instrument you can play, but also to achieve a sense that the instrument it playing with you. This model begins to capture the sense of mutual influence central to the conversation model.

By drawing on our understanding of interactions between players in traditional ensembles, these models help to focus attention on the direction of influence and control. This is an issue that has become an intense area of discussion in other electronic arts with the introduction of dynamic and interactive media (Bongers (2006)). However, over-emphasis on traditional models and musical frameworks brings with it the danger of getting stuck in past paradigms, using new technology to parody old practices, rather than exploring new possibilities. Winkler himself recognises the limitations of using metaphors from traditional performance practice as models for interactive computer music, suggesting that "simulation of real-world models is only a stepping stone for original designs idiomatic to the digital medium" (Winkler (2001) p.23).

Since this time, there has been considerable research effort focusing on real-time analysis tools such as pitch detection, beat induction, phrase segmentation etc. which allow us to design software capable of tracking the sonic gestures of a separate performer. There is also a huge community of researchers exploring novel possibilities for gestural control, both in terms of hardware interfaces and video analysis of bodily movements, in order to explore the potentials of the computer itself as an instrument. Both of these research efforts concentrate on creating a front-end, an means of interfacing with a digital system rather than the model of interaction itself. Whilst the tradition of interaction in music may seem to provide it with a head start in comparison with other art forms, the apparent similarities between the interactive nature of traditional instruments and performance networks, and those afforded by the computer, mean that comparatively less time has been spent considering how the interactive possibilities of the computer differ from traditional performance practices.

This section presents a survey of some of the current approaches and issues in interactive computer music practice. Rowe's original conception of instrument verus player paradigm may be too polar in the current climate, but approaches can be usefully considered along a continuum between these two extremes. This section explores the nature of interaction in a variety of projects from the design of *New Musical Instruments*, through *responsive accompanists* and *virtual improvisers* to *audible ecosystems*. Comprehensive reviews and histories are available in Roads (1996), Chapters 14 and 15, Dean (2003), Impett (2001), Jorda (2002), Rowe (1992) and Rowe (2001)

2.1.2 New Musical Instruments

One of the major research efforts in interactive computer music focuses explicitly on the computer as an instrument in its own right, or as an extension to existing acoustic instruments. The term *Hyperinstrument* refers to a software augmentation of an existing acoustic instrument, whereas *Virtual Instrument*, in the broadest sense, refers to any software-based sound-making system that takes user input. Developing powerful interfaces has become a major field of investigation and research in itself, with institutes like

STEIM[2] and conferences such as New Interfaces for Musical Expression (NIME [3]) dedicated to the development and deployment of new musical instruments and interfaces. **Hyperinstruments**, as the name suggests, aim to augment the musical possibilities of acoustic instruments whilst preserving their expressive potential. In some respects the approach represents a digital exploration akin to the development of extended playing techniques, or the practice of 'preparing' acoustic instruments: one aim is to broaden the possible sound world. Beside the obvious extension of sonic possibilities, digitally extended instruments leave open the possibility to retain the original acoustic output of the instrument, creating space to explore the interplay between the acoustic and processed sounds. The interface between the acoustic and digital instruments can be purely sonic, either employing real-time analysis tools to capture key aspects of performance gesture, or directly treating the acoustic signal. Alternatively, physical interfaces can be implemented by adding sensors and switches to the physical instrument as in the *Hyperbow* developed by Young (2001), Jonathon Impett's *Hypertrumpet* or Ernest Rombout's *Electronic Piccolo Heckelphone*. The best known work in this area has been carried out by the Music and Cognition group of the MIT Media Lab.

> "Our approach emphasises the concept of 'instrument', and pays close attention to the learnability, perfectibility, and repeatability of refined playing technique, as well as the conceptual simplicity of performing models in an attempt to optimise the learning curve for professional musicians." - Machover and Chung (1994) p.186

Many of the early Hyperinstruments developed at MIT were typically used in notated compositions where composers could pay close attention to developing the interaction between acoustic instrumental lines and live electronics. Increasingly however, players are commissioning their own controllers. Performers such as Jonathon Impett and Ernest Rombout demonstrate the new possibilities for improvisation that emerge when a virtuosic player has control over an acoustic instrument *and* its manipulations. Where acoustic instruments are used as the principle controllers, the traditional sense of interaction with an instrument remains essentially unchanged. However, this approach develops upon the sort of interaction we talk about between musical parts themselves. No longer is this the off-line domain of the composer, but the performer now has control via one, albeit augmented, instrument over an indefinite number of musical parts.

Perhaps the most familiar **virtual instruments** are those found in commercial software such as the Virtual Instrument plugins available for sequencers such as Cubase and Logic. These are typically software simulations of familiar analogue instruments such as synthesisers and samplers. But digital instruments are by no means restricted to simulation of existing analogue systems, and experimental research in this area extends far out into the art world: musical (and audio-visual) interfaces also appearing in exhibition contexts at festivals such as Ars Electronica.

That the design of technological interfaces has become worthy of critical appraisal by artists is perhaps testimony to the fact that this represents one of the major facets of virtual instrument design. If expanding the sound world was top of the agenda for early computer music, current research arguably focuses most heavily on approaches to constraining, controlling and expressively exploring the expanded horizons. One of the major problems facing designers and users of virtual instruments is their virtual nature itself, i.e. their lack of physical interface.

---

[2] `http://www.steim.org/`

[3] `http://www.nime.org/`

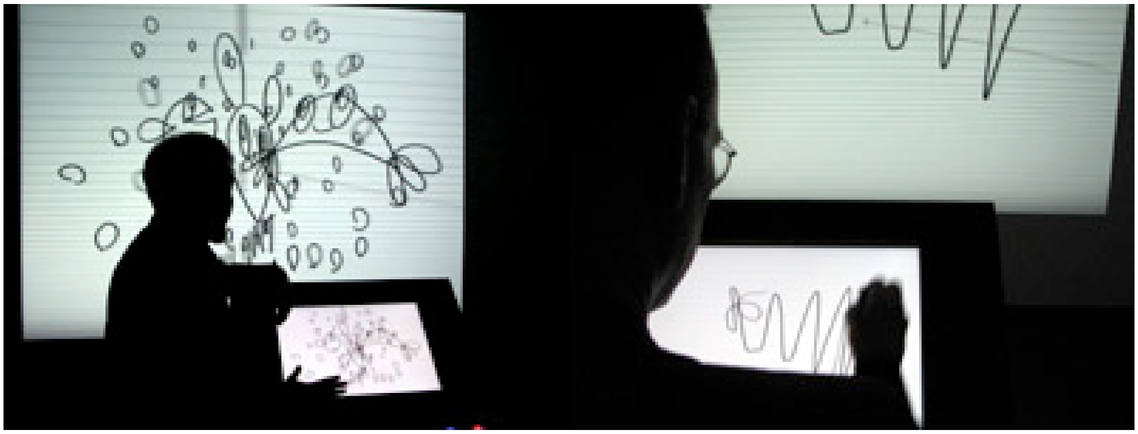

Figure 2.1: Sonic Wire sculpture by Amit Pitaru, an AV interface exhibited at Ars Electronica 2005. (photo: Thomas Petersen)

The physical form of acoustic instruments governs both the nature of the sound they can produce and the physical skills which must be developed in order to shape the sound. The minimal nature of most virtual instrument interfaces can evoke frustrations in the audience as well as the performer. The performance gestures associated with the physical existence of acoustic instruments provide not only sonic character and idiosyncrasies, but a visual correlate that is an integral part of performance from the audience's perspective. For many audiences (the eye-closed classical concert-goer being the exception), *seeing* the trombonist deftly alter his slide position, observing the finegrained coordination between the fingers of a violinist's left hand and right arm or watching the singer's bosom rise, head kick back and jaw drop are almost as important as hearing their sonic results. A look of studied concentration illuminated by the glow of a laptop screen just isn't the same.

For the performer the situation is perhaps even more severe. The physical form of many traditional instruments has evolved over hundreds of years to provide a very high number of degrees of freedom, each with fine-grained continuous control. The instrument is often the locus of integration of physical movements throughout the entire motor-system. Such interfaces take time to develop. I am not suggesting that the emulation of this physical congruence is a good model for digital instruments. As noted above, the decoupling of the interface and sound engine is a characteristic unique to digital and electronic instruments, and the dynamic reconfiguration of these connections is one of their powerful features which deserves exploration. Under such flexible conditions even if a comparably expressive physical interface was developed, the hours of mechanical practice endured by acoustic instrumentalists would be irrelevant: learned sensory-motor contingencies are powerless in such a dynamical setting. This represents an interesting line for future research.

There are a range of approaches that focus on the issues of control and expression which are relevant to both performer and audience. These are *hardware interfaces*, *gestural sensors*, *software interfaces* and *live coding*.

Much work is being done in the development of new *hardware Interfaces*, physical controllers which are used to control underlying Digital Signal Processing (DSP) engines. Many reappropriate existing instrument controllers, such as the MIDI guitar. Others explore idiosynchrasies of the digital or electronic medium such as Michel Waisivic's *Crackle Box*. Some also aim to capture expressive movements made by parts of the body not normally deployed in instrumental performance. Todd Machover's group at MIT for example developed the Sensor Laden Dancing Shoes (Paradiso and Hu (1999)). These shoes are fitted with sensors that send sixteen streams of control data relating to elevation, acceleration, orientation and pressure etc. Data gloves, made popular in the early days of Virtual Reality experiments have been extensively explored, most famously perhaps by Laetitia Sonami with her 'Lady's Glove' [4] (shown in Figure 2.2), the more adventurous or athletic deploying similar technology in full body suits. In recent years, more sophisticated sensors are being developed which provide haptic feedback such as Bert Bonger's force feedback and vibrotactile tools (Bongers (2006)).

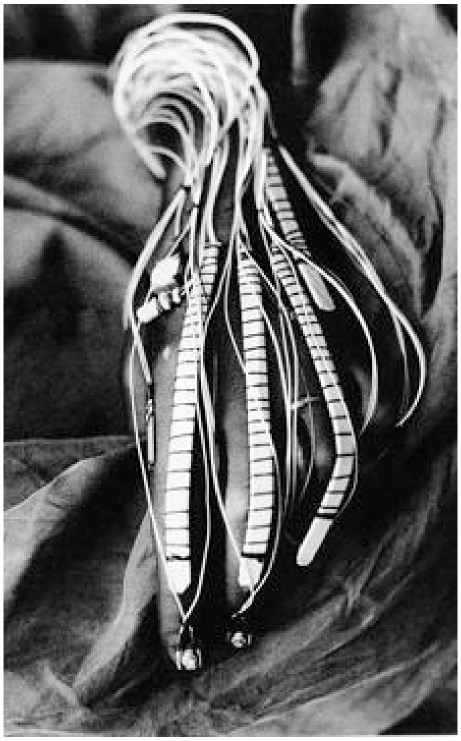

Figure 2.2: Laetitia Sonami's Lady's Glove

Gestural controllers are not only used in performance but also in the studio, one of the earliest examples is Paul de Marinis' use of a data glove to control a voice synthesiser in *Power of Suggestion*[5]

Gestural sensors, such as video analysis, infrared, or ultrasonics bypass the hardware interface and track the physical movements of performers, transforming aspects of their gestures directly into control signals. These might be continuous and high resolution as in the heterodyning oscillators of the theremin, which demands a refined technique comparable to that of a playing a string instrument. In other situations discrete sets of switches or longer distance sensors are used, triggered by larger movements such as that of dancers. In both areas physical movement is transduced with the aim of maximising the performer's expressive control and invariably creating observable correspondences between what the audience can see and hear.

Software interfaces, which add an extra layer of control between mouse-keyboard-screen and DSP engine, aim to put back some of the structural restraints of physical instruments. Interesting examples of work in this area comes from the ixi-software group who have developed a series of graphical front-ends to sound engines such as Pure Data or SuperCollider.

---

[4] `http://www.sonami.net/lady_glove2.htm`

[5] This is a track on de Marinis album *Music as a Second Language*. 1993. (Lovely Music).

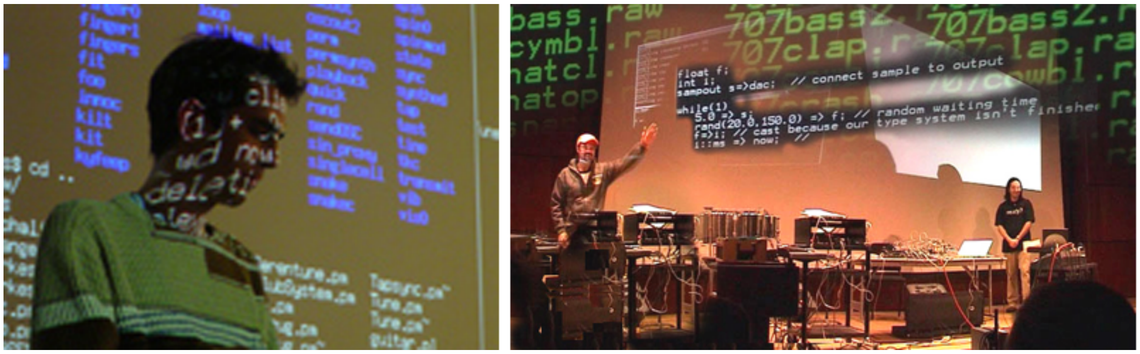

Figure 2.3: Alex McClean (left) and Perry Cook and Ge Wang (right) live coding

Finally *live coding* renounces all attempts to regain the physical, but instead embraces the power of dynamic interpreted languages, such as SuperCollider or ChucK, to write code 'on-the-fly' in performance situations. Rather than attempting to engage the audience with physical gestures, deftly typed lines of code are projected on a screen behind the laptopist, giving the audience an insight into the processes that create what they hear.

A major focus in this area then is in creating flexible and expressive interfaces between man and machine. The computer is employed as a powerful effects unit or DSP engine under the guidance of a human pilot: the main focus for interactive research parallels the concerns of the HCI community, i.e. the efficiency and effectiveness of the man-machine interface.

2.1.3 Responsive Accompanists

Another major aim of early research into interactive techniques was the exploration of methods that allowed a musican real-time control over the tempo of a predetermined computer music score (Winkler (2001) and see Dannenberg (1989) for review of early approaches). In this model, the musician acts as conductor (as in Winkler's conductor model presented in Section 2.1.1) defining the tempo of a programmed score. The interface may be made physically via a hardware controller or sonically using beat-tracking algorithms to follow the musician's performance. Classic examples of this score-following approach include Max Mathews' *Radio Baton*, which enables a musician to control tempo and other aspects of a score stored in computer memory, and Todd Machover's *Bug Mudra*, in which the conductor wears a specially designed glove to control the reverb, panning and mix of the digital score whilst simultaneously conducting three performers (Machover (1992)).

In more recent years there has been an at once more pragmatic and more sophisticated application of these techniques. Christopher Raphael's *Music Plus One* (MPO) (Raphael (1999)) is an 'intelligent' version of the more familiar *Music Minus One* system which provides recorded accompaniments for musicians to rehearse concertos and sonatas. In line with the standard score following approach, representations of both the solo part and the score are input into the system. A Listening process (based on a hidden markov model (HMM)) and Play process (which utilises a Baysian belief network) run concurrently to track and respond appropriately to nuances in the perfomer's interpretation.

> "Specifically, my goals are that the program must respond in real time to the soloist's tempo changes and expressive gestures . . . In this way MPO *adds* to the soloist's experience by providing a responsive and nuanced accompaniment rather than *subtracting* from it by imposing a rigid framework that stifles musical expression." - Raphael (2004)

Some of these features are available in the *Suivi* Max/MSP patches developed at IRCAM[6] (Orio et al. (2003)) which perform score following from audio input using two levels of HMM. One model tracks low level spectral features such as attack, sustain and decay of individual notes; the higher level implements transitions according to notes in the score.

This approach combines the active qualities of digital media with an automated instrument. Essentially we have a pianola which can adjust to the performer it is accompanying. The real development in this approach lies in the analyses devices necessary for the machine to track the performer's progress and respond accordingly. Interaction here then, also focuses on the interface between man and machine, where the interface itself is active.

2.1.4 Virtual Improvisers

At the far end of Rowe's instrument-performer spectrum, sit the 'virtual improvisers'. The aim in this area is to construct an 'artificial player', with a musical presence and personality of its own. It is in this area of interactive computer music then, that the type of 'mutual influence' discussed by Bongers and others is most explicit.

In an attempt to preserve a global musical context, some researchers take what Rowe describes as a transformative approach, modifying either the human player's improvisations directly, or transforming a predetermined database of musical fragments according to analyses of player's input. This approach underlines the current incarnation of Al Biles' *GenJam* (Biles (2002)) which improvises within a traditional Jazz framework by selecting and transforming phrases from a ready made database of Jazz licks or mimicing or modifying the human performer's phrases.

Presented as a practice or didactic tool, Francois Pachet's *Continuator* (Pachet (2002)) uses Markov techniques to build databases of sub-sequences that enable the creation of responses derived from the performer's improvisations. Pachet's system operates in real time to capture key structures of a performer's musical statements. The system operates in both 'Autarcy' mode where it progressively catalogues the input of the current performer, or 'Virtual Duo' mode, where a database built from another musician is used as the transition matrices for the system. The system's performance is impressive, capturing the idiosynchratic harmonic and gestural moves of professional Jazz pianists, and can also act as a structured learning environment for musical novices.

Others aim to increase the independence of the computer system by creating a generative engine that may be influenced by the performer whilst maintaining a certain independence. The best known, longest running (and least disclosed) project in this area is George Lewis' *Voyager* system which is designed to perform improvisations with a human instrumentalist. He regards the computer as "just another musician in the band" (Lewis (1999)), or more specifically "a multi-instrumental player with its own instrument" (p.103). The 'players in the orchestra' are controlled by global behaviour specifications, which are influenced by analysis of pitch and velocity data taken from the player's improvisations. The generative behaviour of the system is developed from white noise, which is shaped and filtered with a series of stochastic rule sets. The generative engine produces musical output regardless of whether or not a human performer is playing. When sonic input *is* detected, feature analysis of many aspects of pitch and velocity data is used to represent the state of the input at a given moment. These in turn influence the behavioural specifications, altering the musical behaviour of the system – or not.

Lewis' move away from the fixed idioms of many systems reflects his view of improvisation per say: "Musical improvisation is one domain among the various possible do-

---

[6] L'Institut de Recherche et Coordination Acoustique/Musique, a computer music research centre set up by Boulez in Paris.

mains of improvisation – an interaction within a multi-dimensional environment, where structure and meaning arise from the analysis, generation, manipulation and transformation of sonic symbols" (p.101).

This dedication to the independence of a digital generative process and a 'bottom up' conception of improvisation is explicit in Tim Blackwell's approach as embodied in his various *Swarm* systems (e.g. Blackwell (2003), Blackwell (2004)). On the premise that structure in free improvisation arises spontaneously from the interactions between players, Blackwell employs particle models of swarming phenomenon. In its simplest incarnation, the swarm is based on Craig Reynold's (1987) *boids* algorithm.

The boids algorithm is often used to illustrate the process of self-organisation, demonstrating how coordinated global behaviour can arise out of simple local interactions without the need for any supervisory control. The basic flocking model consists of three simple rules. These determine the movements of individual particles, or boids, according to the positions and velocities of their neighbours. The rules consider three factors: *separation*, which ensures that individuals do not bump into each other; *alignment*, which causes them to adopt the average heading of their nearby flockmates and *cohesion*, which causes them to move toward the average position of their local flockmates. Rules like this are sufficient to cause co-ordinated flocking in an initially randomly distributed cloud of particles.

Blackwell implements a similar system in an N-dimensional space which is then mapped into musical dimensions. In early incarnations the axes of 'music space' were loudness, pulse and pitch (Blackwell (2003)), later versions used the swarms to parameterise a granular synthesis engine (Blackwell (2004)). Interaction takes place by analysing the performer's input in the same dimensions of musical space in which swarms exist. Musical events in the outside world then become 'targets' in the swarm space to which the swarms are attracted.

Blackwell insists that interactivity, or 'strong interactivity' as he calls it depends upon 'instigation and surprise as well as response' (Blackwell (2006)). Using this swarming model, he creates a system which generates behaviours internally, giving a sense of independence from the performer and moving away from the one-way model of interaction. Lewis similarly eschews the instrument metaphor with its attendant notions of control, taking pains to stress that "the computer system is not an instrument, and therefore cannot be controlled by a performer . . . The computer's own musical behavior is the product of its own initiatives, and its response to outside input when the program has determined that such input is present." (Lewis (1999)).

2.1.5 Audible Ecosystems

Suggesting that the vast majority of computer music systems which are described as interactive should more properly be thought of as 'non-reciprocating reactions', Italian composer and sound artist Agostino Di Scipio submits a more systemic approach in what he describes as *ecosystemic signal processing* (Di Scipio (2003)). In projects such as *Audible Eco-systems Interfaces* (AESI) he broadens the traditional network of performance components from just humans and computers to explicitly include the performance space itself. The traditional roles of each are often inverted as a circular relationship between human performer(s), machine(s) and the surrounding environment is implemented (Anderson (2005) p.16).

Di Scipio's determination to express an alternative to the one-way model, which he sees as an expression of a common conception of interaction as a "determinate machine reaction to a planned human action", is clear. The design of the network of influences between performers, environment and DSP units reveals serious consideration of the ways in which mutual and continuous influence can be achieved: the computer's output may affect the instrumentalists, the instrumental sounds might affect the computer processes, the sound might elicit a resonant response from the room, and this response might feed-back to drive the computer processes or suggest changes in the instrumentalist's performance. In works such as *Texture-Multiple* (1993), and *5 difference sensitive circular interactions* (1998), he sets up relationships between the performers, machines and environment which cannot be understood in terms of simple input-output relationships[7].

As well as using instruments as passive sound-producing devices, Di Scipio develops methods by which they influence the digital transformations of the sounds they are delivering. For example a signal processing module might be set up to automatically alter its internal configuration according to changes in the input sound. As he puts it: "sound sets the conditions and boundaries for its own transformations". In this setting the production of sound is the by-product of a set of interactions, rather than their purpose.

2.1.6 Summary

In many areas of the field of computer music, the dominant model of interaction is a one-way street. Extended instruments focus on techniques for expressive control of DSP engines. Research in new controllers focuses on the interface between humans and computers, a consideration which is similarly central, if more complex, in score following approaches. Virtual performers that take a transformative approach employ some 'intelligent' methods of altering material but are driven cunningly by the performer.

The nature and abilities of digital instruments may have departed from those of traditional acoustic instruments, but the basic operational metaphor remains the same. At times aspects are automated, and at times the digital instrument unfolds ready made or cunningly transformed material, but in many instances of published research, there is little evidence of 'mutual influence': traffic down Winkler's street is essentially one-way. This is not to say that much exciting new music is not being made, and not meant to undermine the usefulness of the one-way model. However, its seems that some of the new interactive idioms which Winkler suggested lay ahead may not yet be being fully explored.

In systems such as Lewis' *Voyager* and Blackwell's *Swarm music* we see explicit exploraration of conversation style models. Di Scipio is not only closing the loop, but setting up circular, self-controlling processes in components throughout the performance system. The software takes on a decidedly active role in the performance. Notions of control have disappeared and are replaced by cooperation creating the possibility for more flexibility in improvised situations and arguably supporting a more spontaneous form of conversation. This system goes beyond any of the traditional performance practice models offered by Winkler. Others seeking novel approaches suggest looking to the broader field of interactive and generative arts. In the next section then, consideration will be given to interaction in the Alife installation arts which adopt a similarly inclusive, cooperative approach.

2.2 Interaction in Alife Installation Arts

Since the late 1990s, many artists interested in exploring this shift from control to cooperation and the creation of emergent behaviours in systems with a degree of autonomy have been exploring techniques drawn from the field of Alife. Alife is an interdisciplinary scientific field concerned with the creation and study of artificial systems that manifest low-level properties of living systems. In contrast to traditional AI which concerns itself primarily with high level cognitive competencies specific to human, Alife focuses

---

[7] A similar set up was explored in using analogue electronics by Gordon Mumma in *HornPipe* (1967)

on the basic adaptive abilities common to all living creatures: rather than the top-down approach of AI which specifies representational models of cognitive capacities directly, Alife is characterised by a bottom up approach in which adaptive behaviours emerge out of the organised interactions of low-level dynamical processes. The differences between these paradigms are considered in more detail in Section 2.3.

Many artists have looked to Alife techniques specifically because of the richer possibilities they afford for interaction. Ken Rinaldo, for example writes:

> "Perhaps the greatest potential for the arts and Artificial Life techniques is that they have presented opportunities for both artists and viewer/participants to develop true relationships with the computer that go beyond the hackneyed replicable paths of "interactivity" which have thus far been presented by the arts community." - Rinaldo (1998), p.374

For the visual arts, Alife techniques brought about a significant change in the nature of the artwork. As Sommerer and Mignonneau (1998) describe " the art work . . . is no longer a static object or a predefined multiple choice interaction but has become a process-like living system." (p.158). This change has brought many of the concerns and ambitions of visual installation arts much nearer to those in the performing arts. For example Rinaldo's (1998) vision of "a cybernetic ballet of experience, with the computer/machine and viewer/participant involved in a grand dance of one sensing and responding to the other." allies closely with the aims of those developing music performance systems.

This section looks at some of the ways interactivity has been developed within Alife installation arts. Consideration is given to the balance of influence between user and the system and how much freedom there is for 'spontaneous' conversation.

2.2.1 Breeding Artificial Forms: Interacting with Evolution

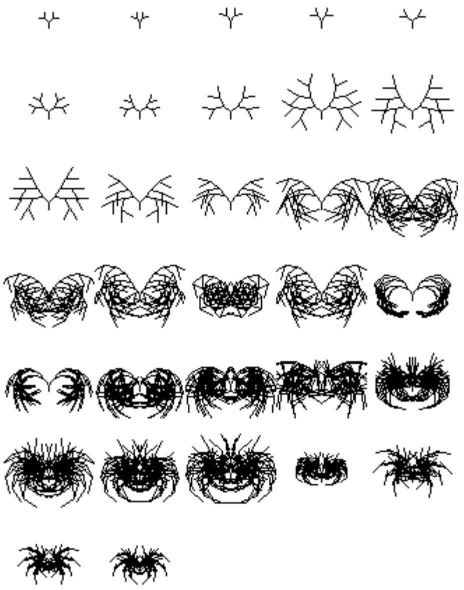

Figure 2.4: Screen shots of the 'stages of evolution' of one form in a biomorph environment. Each image (left to right, top to bottom) represents a mutation of the previous.

Some of the earliest explicit applications of Alife techniques in the interactive arts is seen in the work of William Latham and Karl Sims. This work has spawned a large and growing community of what has been dubbed 'breeder art' (Whitelaw (2004)). These original works were heavily inspired by *Biomporph*, a program which evolutionary theorist Richard Dawkins developed and discussed in his book *The Blind Watch Maker* (Dawkins (1986)). Dawkins used Biomorph to illustrate his arguments concerning the creative capacity of Darwinian evolution, namely that random variations created by chance mutations can be shaped into complex forms by natural selection.

The program presents the user with a selection of stick-built forms. The user selects an individual to become parent. The genes of the selected individual are copied and each copy is altered slightly creating a set of similar, but not identical, children. The 'genes' of each individual determine its visual appearance, specifying aspects such as colour, number of segmentations, depth of recursion, seperation of segments etc. From a starting point as simple as a five pixel cross, complex insect-like forms can be evolved. Some examples are given in Figure 2.4. Dawkins' premises and perhaps implementation are debated in the biological and evolutionary communities, but to the artistic community *biomporph* demonstrates that artificial genetics and artificial evolution, guided by human aesthetic preference, can give rise to complex visual objects (Whitelaw (2004)).

Karl Sims' (1991) *Genetic Images* was shown at Ars Electronica and the Pompidou Centre in 1993. The installation itself consisted of an arc of sixteen video screens, each displaying colourful abstract images. Users interact with the piece by pressing on the touch sensitive pads placed at the foot of each screen. Their selections form the basis of the next generation; over succesive generations they exert a steady influence over the nature of the generated images as they guide the process of graphic variation. The genome in this case is a mathematical equation. If one image is selected this single equation will be altered randomly to reproduce another sixteen forms. When two images are selected they are spliced together using a process analogous to cross-over in biological sexual reproduction to produce another sixteen images, each bearing hallmarks of the parents.

Figure 2.5: Karl Sims' Genetic images on display at the Pompidou Centre, Paris (1993).

William Latham's *Mutator* employs a similar process of aesthetic selection but operates on geometrical procedures rather than mathematical equations. The basic geometric building blocks and methods of transformation and accumulation are evident in the final forms which are reminiscent of mutant space age crustacaens (shown in Figure 2.6). Similar techniques have been widely adopted throughout the arts community. Although essentially a constraint satisfaction algorithm, or search tool, emulation of Darwinian processes raises discussion of the Promethean status of the artist. Just as electronic instruments blur the boundaries of instrument maker, composer, and performer, many working in this area discuss the dual roles that the artist takes on. Todd and Latham (1991) in particular discuss the way that 'evolutionism' changes the role of the artist. Rather than creating a work directly, the artist's task now is "the creation of generative systems and structures" on one level and "the selection of specific forms and animations" on the other. Introducing an analogy, partly perhaps inspired by the organic forms, Latham and Todd analogise these roles: "The artist first creates the virtual world ... then becomes a gardener within this world he has created" (p.12).

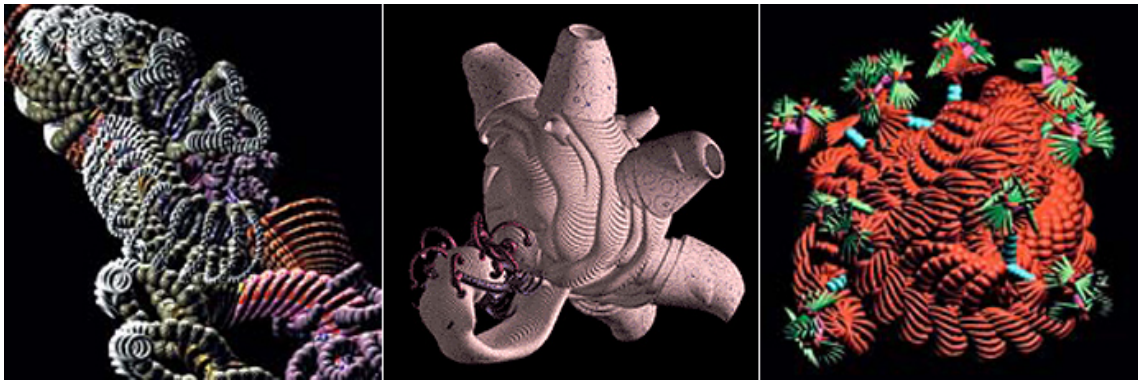

Figure 2.6: Examples of forms evolved in William Latham's Mutator.

But how far does this really take us from the one-way model of interaction? The role of the artist-creator allies with that of the instrument maker in defining a space of possibilities; the artist-gardener nurtures and explores these just as the player explores the bounds of their instrument. Part of the process is automated of course, giving some degree of agency which can surprise or upset the gardener player, but the gardener essentially remains in control of a collection of computational specifications. Existing musical applications of these techniques which will be examined more closely in Chapter 3 support this.

Others have developed more sophisticated interfaces. In Christa Sommerer and Laurent Mignonneau's *Interactive plant growing* (1993)[8] for example, users can influence the development of plants. As in both Sims' and Latham's program, these are virtual silicon graphic plants, but rather than pressing buttons to generate a series of static forms, a small garden of real plants acts as interface to a dynamic jungle of virtual plants. A user touching one of the potted plants may cause a three-dimensional fern like plant to start growing on the screen; another user brushing past a different potted plants triggers the growth of a vine, tree or moss. Each potted plant is fitted with electrodes that pick up the electrical potential of the plant. The signals differ according to the way the plants are touched. Voltage changes are then mapped to parameters which influence the growth patterns of the synthetic plants.

Although the technical processes do not depart wildly from the interactive GA, the juxtaposition of organic and synthetic plant life illustrates an interplay of natural and synthetic forms which resonates with considerations of integrating acoustic and digital musical practices.

---

[8] `http://www.mic.atr.co.jp/~christa/WORKS/CONCEPTS/PlantsConcept.html`, `http://www.medienkunstnetz.de/works/the-interactive-plant-growing/`

Figure 2.7: Christa Sommerer and Laurent Mignonneau's A-Volve.

2.2.2 Artificial Ecosystems: Interacting with an Environment

As the complexity of the simulated world grows, so the influence of the user's action diminishes. Many artists have developed systems that are based on the ecological concept of an ecosystem. These systems tend to model not just the genetic code of individuals, but the behaviours between individuals and the complex interactions between multiple species and their environment.

Many artists have explored these system in non-interactive situations where visitors are invited to observe the evolution of new forms and behaviours as the virtual entities themselves interact – mating and being born, eating, dying and competing for resources. Some forms of agent based modelling have been explored by composers, which will again be considered in Chapter 3, but they are used as closed compositional systems rather than in any interactive performance situations.

Those that have opened up these environments to human influence in an exhibition setting provide an interesting model of interaction that is subtle yet reminiscent of our interactions with aspects of the real world. In *A-volve*[9] Sommerer and Mignonneau (1997), develop the intuitive physical interface, or natural interface as they call them, of *Interactive Plant growing*. The piece consists of a virtual pond projected on the under side of a shallow water-filled glass tank. The pond is stocked with strange aquatic creatures: digital coelenterates slowly pulse along the edge, attracted to the visitors' hands that are dipped in the water.

A-volve not only simulates the basics of genetic evolution but incorporates rudimentary vision and a range of simple behaviours. These extend beyond predation and mating to deal with aspects of the physical and social worlds such as collision avoidance and parental protectiveness. The system also includes a model of basic jelly-fish-like

---

[9] `http://www.mic.atr.co.jp/~christa/WORKS/CONCEPTS/A-VolveConcept.html`, `http://www.medienkunstnetz.de/works/a-volve/`

propulsive locomotion which takes the radial shape of the creature into account, relating physical form to swimming ability and thus fitness.

Visitors can interact with the installation in two ways. By drawing on a pressure sensitive tablet on the side of the pond, they can introduce a new species into the pool. Their sketched outline is then rotated in three dimensions to create the body shape of their new species which is then released into the pool. Because radial form determines swimming ability, users can introduce a streamlined predator capable of wiping out the more sluggish members of the population, or a cursed defenceless blob which may provided refreshments for the faster fish to feed on. Motion tracking and shape-detection techniques bind the physical and virtual tanks, allowing visitors to interact with the creatures in an open and intuitive way: by dipping their hands in the tank, visitors can protect a creature from being eaten, or herd two together to encourage mating (or predation if they are that way inclined). The artists report that many become protective over their own creations, selecting its mates, or even gathering food for it.

Figure 2.8: Jon McCormack's *Eden* in installation.

Jon McCormack's *Eden* similarly allows visitors to influence the development of a virtual ecosystem. Derived from Holland's ECHO system (e.g. Forrest and Jones (1994)), Eden is a virtual projected space inhabited by rule-based, evolvable agents. These are represented visually as abstract forms that similarly predate, mate and evolve over time. Rather than the simulated sight of A-Volve populus, McCormack encodes a complex mechanism for the evolution of sonic communication. Unlike many models (some of which will be discussed in the next chapter), the ability to 'sing' and locate sound sources, does not come with a hardwired survival value. Never-the-less, in some runs McCormack (2001) reports that agents can evolve to utilise their sonic capabilities. Some creatures evolve altruistic behaviour, calling to invite others to share an abundance of food, others learn to exploit this altruism by developing siren-like tricks to lure their neighbours to their death.

Sadly the creatures of Eden (at least in current incarnation) do not respond to heckles or wolf whistles from exhibition visitors, however video analysis fuses the virtual and physical space, allowing visitors to influence the evolution of the ecosystem. The presence of people in the real environment increases the rate of biomass growth at the corresponding point in the virtual world. As this is a spatial world, the idea is that agents with the most interesting behaviour will attract more people to their local vicinity, increasing their potential food resource and so chances of survival. Conversely the level of movement in the physical world increases mutation rates of agents in the localised virtual world: increasing mutation rates means that new (although not necessarily better) behaviours or forms are more likely to evolve.

By McCormack's own admonition, the time scales involved in Eden's evolution, and the subtlety of influence means that many people are completely unaware that they exert any evolutionary pressure on the system. Indeed a major motivation for this twist to the straightforward interactive evolution of the gardening variety is to try and create an situation in which a more open-ended evolution may occur: interaction is a means to an end rather than the end in itself.

Both McCormack and Vorn and Demers, who work with robotic sculptures, describe their systems as reactive rather than interactive, emphasising the fact that users do not gain control of the self-steering system, but rather influence the unfolding of higher level events. Interaction in these systems is a far cry from determined control of the one-way model. The dynamic, adaptive nature opens possibility for spontaneous conversation, but the complexity of the system and indirect affect of the human visitor is such that the power of influence has swung the other way: the system is doing more talking than listening.

2.2.3 Single Synthetic Agents: Interacting with an Autonomous Other

Figure 2.9: Simon Penny's Petit Mal in action.

The sense of agency which emerges from these digital ecosystems is at once more minimal and more powerful when the visitor interacts with a digital creature on a one-to-one basis. One of the simplest, but perhaps most universally appealing examples of 'cybernature' is Richard Brown's *mimetic starfish*[10]. This is an almost photo-realistic, although over-sized, starfish which is projected onto a low circular white table. A transparent interface tracks the movements of visitors' hands and the image responds incredibly evocatively: tentacles stretch out languidly in response to kind, soft stroking and recoil sharply at fast, aggressive movements. Even inanimate objects placed in the vicinity of a tentacle cause it to reach out as if the starfish is inquisitively trying to examine the object.

Although perhaps not the most sophisticated example of Alife art, the starfish's basic reactive responses induce visitors to enter into a form of gestural turn taking, forming a prototypical communication of gesture. The physical reality of the system is nothing more than an array of projected light, but the starfish is attributed with a degree of agency and even personality.

This sense of autonomous agency is increased considerably when artworks escape the confines of the virtual and are palpably present in the form of physical robots. Artists such as Rinaldo, Yves Klein, and Vorn and Demers have all taken this leap into the real world which enables an interface-less open form of interaction. *Petit Mal* for instance is Simon Penny's 'autonomous robotic artwork'[11]. Petit Mal stands just over a metre tall and physically comprises a scrawny counterweighted column encircled with ultrasonic and pyroelectic sensors balanced precariously on two bicycle wheels.

> "The robot presents someone with the impression of a non-human, non-animal sentience, which then has to be dealt with in some way. If they run away, it will chase them. If they want to play, it will play. If they are aggressive and advance, it will back off. At some point, if you're boring, it gets bored and goes away." - Davis (1996), p.32

In contrast to the lavish graphical worlds of the virtual ecosystems described above, Penny's work presents the minimal requirements for a basic illusion of sentience. Visually he makes an effort to present the public with something that is neither a biological simulation, nor an automaton, but something that is 'substantially itself'. He describes his approach as under-engineering, and capitalises on mechanical or electronic quirks as the generators of emergent behaviour such as the dynamics of the double pendulum structure which is the central control system, or deficiencies in sensor readings. It is these quirks, he suggests, that give rise to its "personality". Petit Mal is an attempt to explore interactive machine behavior in a real world setting. The reflexive nature of interactivity is a focal issue: interactive behavior is defined by the cultural experience of the human visitor.

Pieces like these support a simple but powerful conversational interaction. Petit Mal in particular engages those it encounters, seemingly judging their moves and altering its behaviour accordingly. These basic behaviours support a strong sense of interaction. Much of this perhaps is associated with its embodiment, as Penny himself says, 'evaluation of interactivity is subjective', this raises the importance of considering not only the formal system, but how it is dressed and presented to the world. The alluring personalities exuded by some of these artificial forms inspire a possible style of conversation for our digital performance partners. The sense of agency that we attribute to some of these systems would be a very attractive property in a musical performance partner, offering a novel alternative to transformational approaches for generating musical material.

2.2.4 Summary

In many respects, the nature of interaction in these systems is much closer to the fluid models of communication sought by members of the interactive music community: no-

---

[10] `http://www.mimetics.com/starfish.html`

[11] `http://www.ace.uci.edu/penny/works/petitmal/petitcode.html`

tions of control and masterability central to the one-way model are replaced by indirect influence and cooperation. This represents an attractive shift for those aiming to create a sense of distinct musical personality in man-machine improvisation. But in many cases, the systems are perhaps *too* independant. McCormack and Vorn and Demmers describe their systems as reactive, stressing that visitors can only indirectly and inconsistently influence their path. This is an issue which Rinaldo sees as an obstacle:

> "One difficulty with some Artificial Life artworks is that the systems may not seem to be responsive to the changing environment, as the work demonstrates its own internal desires. This can make the work seem unresponsive or uncaring." Rinaldo (1998), p.373

Control of the dialogue has switched: the machine has gained control and is not listening. How then could we gain the balance desirable for man-machine improvisation? Simply reducing the complexity of the system so that the human visitor engages with a single virtual entity rather than an entire ecosystem as in Richard Brown's starfish and Penny's Petit Mal seems to begin to readdress this balance and evoke an attractive form of conversational interaction. Not only is there a proto-conversation, but the aesthetic achievement of a sense of artificial agency. In contrast to the transformative approaches used in artificial improvisation software, the system demands to be encountered on its own terms. How can we understand the nature of these differing forms of interaction so that we can preserve the desirable qualities of the fluid conversation but temper the balance such that we can employ these systems on stage for musical performance? How does the framework adopted by Alife artists differ from that of traditional interactive music?

2.3 Frameworks for Understanding and Implementing Interactive Systems

This first part of this section considers the dominant design protocol in current interactive computer music research, and contrasts this with the scheme of those promoting the conversation model. These two approaches ally closely with two different philosophical perspectives on interaction which are introduced in Section 2.3.2. It is suggested that existing approaches tend to operate within a computationalist paradigm and that the adoption of a dynamical perspective may support the development of software that is capable of engendering a more conversational style of interaction.

2.3.1 Design Schemes: Pipeline vs Circular Causality

Traditionally, interactive software design is split into three principle parts: sensing, processing and responding (Rowe (1992)). Shown schematically in Figure 2.10 the first two steps can be divided into sub-tasks as in Winkler's description:

- Listening (input). Human activity is translated into digital information and sent to the computer.
- Listening (analysis). The computer receives human input and analyses the performance information for timing, pitch, dynamics etc.
- Processing (interpretation). The software interprets the computer listening information, generating data that will influence the composition.
- Processing (composition). Interpreted information is used to guide the generation of composition data.

- Responding. The computer plays the music, using sounds generated internally or by sending musical information to devices.

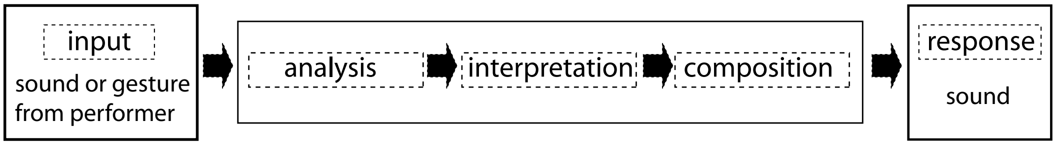

Figure 2.10: Schematic of system design protocol in interactive computer music.

This basic protocol has remained unchanged, appearing in recent reviews (Schnell and Battier (2002)), and is virtually ubiquitous throughout the community. Although we may assume that most performers think in terms of a conversational model when playing music, a notion that presumably carries over to performance in digital systems, this schema is used to describe, and thus presumably influence, the design of performance systems. As is evident, the basic ontology is linear, what we might call a pipe-line model of information processing. The performer exerts control over the system by providing sonic input. The system analyses it and computes a response. We may assume that the future actions of the performer in most circumstances will be affected by the sound, so affecting the state of the computer system and closing the loop, but this is only implicit. Although this scheme is presented again and again in the context of performance systems, it sidelines the fundamental feedback from the system output to the performer.

The other noteable feature of this design scheme is the decomposition of the task into functional components: input, analysis, interpretation, composition, response. Naturally, many of the steps in this process are based on cognitive models of the comparable processes in humans. For example improvisation systems designed by Wessel in the late 1980's, for performance with Roscoe Mitchell and later with Ushio Torikai, consisted of a collection of objects divided into three modules: *listening assistants*, which extracted musically meaningfull information, i.e. DSP objects for pitch extraction, parsing, and tempo extractors and objects for musical analysis; *composing assistants*, which helped the improvisers manage the construction and set-up of their performance materials on the fly; and *performance assistants*, which supported gesture mapping, phrasing and articulation. The listening assistants were based on a model of memory influenced by cognitive psychology, comprising both direct short and more abstract long term memory. The design of the phrase boundary was based on the elementary grouping mechanisms taken from Lerdahl and Jackendoff's Generative Theory of Tonal music (Lerdahl and Jackendoff (1983)), whilst the tonal field estimation scheme was based on a method developed by Krumhansl (1990). This approach has proved very powerful in certain areas, and the development of sophisticated and powerful audio analysis tools are undoubtedly central to the successful development of digital performance systems.

A potential problem though, is that this pipeline model reinforces the metaphor of one-way interaction. Coupled with the power of these intelligent analysis modules, the motivation for considering other forms of interaction is obviated. As noted above, electronic and digital instruments are distinct from acoustic instruments in that the interface is decoupled from the sound producing mechanism. It is very natural then to approach the design of new instruments by focusing on the input device and response mechanism as distinct design problems. Layers of complexity are then added as interactive systems become more sophisticated and this model is applied not only to instrument design, but becomes the de facto approach to 'artificial players'. In a recent review and project proposal, Robertson and Plumbley (2006) describe their aims of creating an 'autonomous player' and break down the task in exactly this way. "The first stage is the development of a system that is capable of predictive score following and sequencing. We consider that this is a sub-problem of our eventual aim of an interactive system capable of autonomous generation of output." (p.3).

Many researchers wanting to push the boundaries and explore what they feel are richer interactive relationships invariably invoke discussion of feedback and circular causality within a systemic view. Garth Paine for example takes inspiration from Cybernetician Norbert Wiener in presenting a more inclusive consideration of the elements of an interactive system:

> "In a sense, the ... interactive musical environments creates an ecosystem formed by the human presence and nature of behaviour, the response of the technology (the aural or visual response as experienced by the inhabitant of the installation) and the space itself. The process of understanding this dynamic relationship between the human condition and the physical space is supported by the study of cybernetics, and in particular the closed causal loop." - Paine (2002), p.301

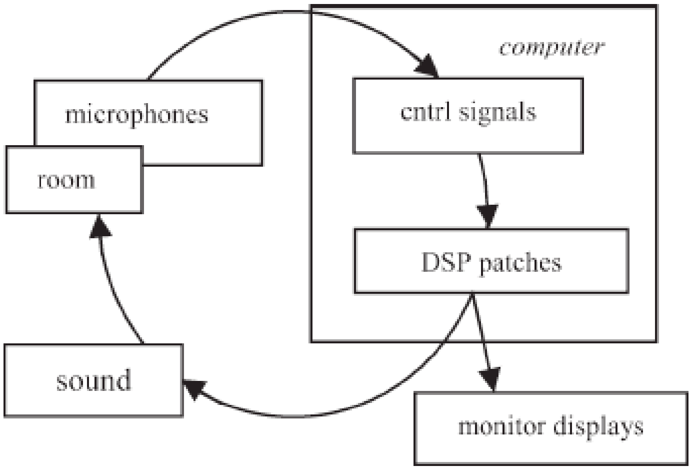

Figure 2.11: Basic design of Di Scipio's Audible Eco-Systemic Interface.

This is almost precisely the approach taken by Di Scipio in AESI, the basic design scheme for which is shown in Figure 2.11. This closed causal loops captures much more accurately the ongoing negotiations that can occur simultaneously between individuals in a group of performing musicians. In aiming to build systems with enough freedom to improvise with, the basic design scheme should ideally facilitate the possibility for this continuous and simultaneous influence throughout all parts of the system. The switch from a linear to a circular scheme intuitively supports this more open, dynamic relationship between the system and performer. Closing the loop infers a continuous flow of influence. Establishing a continuous flow, rather than set of determinate feed-forward commands in turn increases the flexibility and so the possibility for more spontaneous communication.

2.3.2 Heteronomy vs Autonomy

The distinction between the determinate control of the one-way model and the mutual influence of the conversation model are well recognised within certain philosophical circles where they represent the paradigms for interacting with *heteronomous* and *autonomous* systems respectively. Autonomy and heteronomy literally mean self-governed and other-governed: an autonomous system is described as a self-determining and self-maintaining system; a heteronemous systems is determined and controlled externally.

Biological entities across many scales – a living cell, a jellyfish, an ant colony, or a human being – behave as a coherent, self-determining unity. On the other hand, most human-designed technology such as an automatic bank machine is determined and controlled from the outside (Thompson (2007)). In terms of organisational structure, a heteronomous system is defined by an input-output information processing flow which is controlled externally. By contrast an autonomous system can be characterised by highly recursive network of dynamic processes that generate and maintain internal variants in the face of external (and internal) disruptions (Varela (1979)).

Traditional computationalist systems – both cognitivist and connectionist – typify the heteronomous perspective. For example a typical connectionist network has an input layer and an output layer. The inputs are assigned by the observer of the system, and the output performance is evaluated in relation to an externally imposed task. Rule-based or statistical models such as the Markov chains used in the improvisation systems of Pachet and others work in a similar way, albeit with one layer of abstraction. The observer (composer or performer) selects or plays a set of notes (inputs) into the system, the system creates an abstracted representation of this input data. This is then used to define the output according to subsequent given inputs. These systems are typically constructed from the 'top down', comprising independent processing modules which pass information to each other.

In contrast, an autonomous perspective assumes a dynamical approach in which there are no inputs or outputs in the usual sense, but rather a closed loop of circular causality. The dynamical processes of the performance networks composed by Di Scipio, or the evolving Alife ecosystem models lie closer to this notion of autonomy by virtue of the recursive loops which bolster endogenous, self-organising and self controlling dynamics. Within this framework, design is attacked from the bottom up. As Di Scipio notes, in a musical context the final sonic output is the 'by product' of sets of composed interactions.

In these next sections, a brief outline of the computationalist approach is given, followed by a closer consideration of autonomy and an outline of how the attendant dynamical approach differs.

2.3.3 Cognitive Science and the Computationalist Approach

Mainstream cognitive science can be characterised by two central pillars. Firstly the computationalist theory of mind is upheld, which supports the 'physical symbol system hypothesis'. This states essentially that a physical symbol system like a computer is both necessary and sufficient for general intelligent action (eg Newell and Simon (1976)). Secondly, internal activity is divided into a sense-think-act (or sense-model-plan-act) sequence as in David Marr's description of how a three-dimensional world model could be constructed from a two-dimensional image in order to generate appropriate action.

These two principles were brought together in a classic paper by Lachman et al. (1979) in the late seventies where they define cognitive science as "how people take in information, how they recode and remember it, how they make decisions, how they transform their internal knowledge state, and how they translate these states into behavioural outputs" (p.99). If we compare this to the dominant scheme in interactive computer music we see just how strongly the two are allied: 1) translate human activity into digital information, 2) analyze the performance information and extract pitch, timings, dynamics etc., 3) interpret information for use in influencing the composition, 4) process this information to generate a composition and 5) send information to sound generating device and output. (Winkler (2001) p.6.) The approach, both in terms of functional decomposition and the linear information processing flow represents the classical cognitivist approach to cognition as practised in classical AI.

2.3.4 Autonomy

Definitions of autonomy are couched in many different ways, from practical sets of engineering constraints in robotics research, to extreme theoretical accounts of the fundamental organisational principles of biological life (Varela (1979)). Since the 1990s, the term autonomous robotics (Maes (1991)) has been used to refer to a set of engineering constraints on the design and evaluation of robots in both cognitive science and engineering. These include conditions like mobility, and real-time response in real-world environments, no remote control, no external energy supply and no human intervention in task solving. Brooks' situated robotics (e.g. Brooks (1991)) develops upon these ideas. Many Alife artists working in hardware acknowledge the work of these researchers as a major influence, and it is perhaps the achievement of these engineering constraints which Rinaldo and Penny refer to when they talk of autonomy.

Practical research in autonomous robotics has driven a deeper consideration of the notion of autonomy. Increasingly consideration is given to the types of interactive processes that are established between the robot and its physical environment, as well as the properties and dynamic structure of its control mechanisms. Engineers such as Tim Smithers (1997) and Randall Beer (1995) strongly criticise the classical computational information processing approaches which provide little room for considering these aspects, putting forward dynamicism, embodiment and situatedness in place of the virtual, formal approach of traditional AI.

Many working in this field have been greatly influenced by the biologist and philosopher Francisco Varela. He approaches autonomy in terms of the organisational structure of an entity. For Varela, autonomy comes about by virtue of a systemic organisation which defines its own identity: a kind of self maintaining, self-reinforcing and self-regulating system subserved by a highly recursive network of dynamic processes, capable of generating and maintaining internal variants in the face of disruptions both internal and external (Varela (1979) p.55). Varela defines an autonomous system as one which has organisational and operational closure. Closure doesnt mean that the system is cut off materially and energetically from the outside world (that would be impossible) but refers to a system whose organisation is constituted by a network of internal processes. The operation of the network is sufficient for those constituting processes to be generated and sustained without any of them being driven from outside the system.

Ruiz-Mirazo and Moreno (2004) give an account of *basic autonomy* in terms of the energetic and thermodynamic requirements in the physical world. This brings with it specific and demanding physical-implementation requirements: "the system must be made up of certain types of components, specifically a semipermeable active boundary (a membrane), an energy transduction/conversion apparatus (an energy currency like ATP in living cells, which transfers energy from chemical bonds to energy-absorbing reactions within the cell), and at least one type of component that controls and facilitates the self-construction processes (catalysts)". (p.252).

In the biological domain, this form of autonomy is exemplified by a living cell. The recursive constituent processes in this case are chemical. Their recursive interdependence takes the form of a self-producing metabolic network which also produces its own membrane. This network constitutes the system as a unity in the biochemical domain and determines a domain of possible interactions with the environment. This kind of autonomy in the biochemical domain is referred to as *autopoiesis* (Maturana and Varela 1980).

A system does not have to be autopoetic in the strict sense of maintaining a self-producing, bounded molecular system, but it remains unclear exactly how the autonomy of an artificial agent can measured or implemented. Aiming to develop a more operational definition for use in Alife simulation, Barandiaran (2004) has recently developed a definition of autonomy in the behavioural domain. Working within a dynamical systems framework, Barandiaran models the metabolic constructive processes of basic autonomy as a set of essential variables which tend to stay away from equilibrium. He describes a *behavioural adaptive autonomy* which is defined as "homeostatic maintenance of essential variables under viability constraints through self-modulating behavioral coupling with the environment, hierarchically decoupled from metabolic (constructive) processes." (see Barandiaran (2004) for explication).

Now none of the art works or music systems we have looked at are autonomous on these terms. Indeed we may not really want a truly autonomous system as a partner in improvisation. However, the framework adopted by researchers taking this autonomous perspective on cognition provides much inspiration for the current project. Unpacking some of the tenants central to those adopting autonomous approaches to cognition and Alife helps contextualise the instrument and conversation models of interaction, clarifying the differences between them and providing conceptual and practical tools for the current project.

2.3.5 Dynamical Approaches

Proponents of autonomous robotics and Alife are a sub-population of a large community of research adopting a dynamical approach to mind science (e.g. Kelso (1995), Lewis and Granic (2000), Port and Van Gelder (1995), Thelen and Smith (1994). In contrast to the cognitivist hypothesis mentioned above – that cognitive agents (natural and artificial) are digital computer or physical symbol system and that cognition should accordingly be explained in symbol processing terms – the dynamical hypothesis postulates that the cognitive systems instantiated in natural agents are dynamic systems and therefore that action, perception and cognition should be explained in dynamic terms.

The cornerstone of the dynamical approach is the emphasis it places on time. Traditional computational models are static, in that they specify a sequence of discrete states which a system passes through. Dynamic-systems models specify how a process unfolds in real time. As Tim Van Gelder (999a) states, "Although all cognitive scientists understand cognition as something that happens *over* time, dynamicists see cognition as being *in* time, that is, as an essentially temporal phenomenon." (p.244). Elsewhere he compares the approaches with a series of oppositions: change versus state; geometry versus structure; structure in time versus static structure; time versus order; parallel versus serial; and ongoing versus input/output (Van Gelder (1998)). At the lowest level then it should become apparent that the adoption of a one-way instrument metaphor implemented within a pipe-line model of information processing allies with a computationalist approach to cognition, whereas the conversation model based on circular feedbacks sits within a dynamicist understanding.

Dynamicists conceive of state changes in terms of their position and trajectory in phase space, i.e. geometrically: computationalists focus on the internal formal or syntactic structure of combinatorial entities. Computationalists think of cognition as the rule governed transformation of one formal static structure into another, whereas for dynamicists cognitive structures are laid out as temporally extended patterns of activity. Cognition is seen as the flow of complex temporal structures mutually and simultaneously influencing each other. Dynamicists are therefore interested in the timing (rates, periods, durations, synchronies) of processes, whereas computationalists have traditionally not been interested in these temporal details, but only in the order in which cognitive states occur (Thompson (2007)).

The serial progression of listen-process-respond is the musical equivalent of the computationalists' sequential ordering of cognitive subtasks into sense, plan, think, act. This contrasts with the dynamicists' conception of cognition as the unfolding of a continuous coevolution of acting, perceiving, imagining, feeling, and thinking. These basic comparisons promote two very different understandings of the relationship which an entity has with its environment, of interaction. Let's assume that the entity is a human musician and the environment is the piece of performance software. Under the one-way model, the human sends a trigger which sets off a series of events that flow down the processing chain. The response is fully determinate so it can be controlled and mastered. The conversation model assumes that both entities are autonomous systems each maintaining their identity and thus independence but influencing each other through a process of structural coupling. Coupling refers to the fact that the conduct (dynamics or behaviour) of one system is a function of the conduct of the other. In dynamic systems language, the state variables of one system are parameters of the other system and vice-versa (where a variable determines the state of the system along a trajectory within a particular field, and parameter determines the field in which it currently exists). In a truly autonomous system, the domain of interaction is determined internally, thus we cannot absolutely predict what its response will be. We cannot control it. A completely autonomous system in this sense is neither desireable nor achieveable as a performance partner. But *some* degree of internally generated state is the first step toward creating a sense of distinct musical personality.

The dynamical perspective provides a framework within which we can understand how our action can have differing levels of influence on a system: why some Alife systems are 'uncaring', whilst others such as starfish are so attendant to our moves.

Agent-based evolutionary ecosystem models such as that used in McCormack's Eden are described by a large set of interdependent processes operating across different time frames. The global system is constituted by a complex set of evolving processes operating on multiple time scales. The behaviour of any one agent is determined by the state of its local environment (which could include other agents) and the state of its internal system. This internal system mediates the sensors and actuators, so determining its interaction with the environment and is subject to lifetime learning as well as evolution. Population dynamics are influenced by the inbuilt seasonal variation in biomass density, which will presumably cause population levels to fluctuate even in the absence of any visitors. The affect we have on the system then, in influencing either biomass growth or mutation rate is only one of many factors that determine any observable changes at the level of agent behaviour, sub-population or global population. Additionally any affects will firstly be extended in time, and possibly space. This makes our influence perceptually inseparable on a micro level from the effects of default seasonal variation or the effects induced by others in the exhibition.

In Brown's starfish however, there is a much stronger and more direct coupling between our behaviour and the behaviour of the agent. Firstly there is only one agent, secondly the interaction occurs in the same time scale and is local, and finally although there may be a small amount of noise or minor oscillation injected into the process which controls movement, the principle variable is the motion detection system which directly initiates reaching or recoiling behaviour.

2.3.6 Bottom Up Design

These examples also illustrate the impact of adopting a dynamical approach in terms of the conceptualisation and design of a system. As noted above, system design under a computationalist approach typically proceeds in a modular fashion, where for example the interactive interface is designed and implemented separately from the composition module. As becomes apparent in considering Eden, what the audience interacts with isn't some separate interface module, but the resource levels in the environment which are intrinsic to the fundamental processes by which the whole system is constructed, maintained and evolves.

These opposing perspectives ally closely with the top down versus bottom up approaches taken by cognitivist and dynamical approaches to understanding cognition: whereas cognitivists focus on some kind of central processor or homunculi that controls behaviour, a dynamicist considers the distributed and functionally integrated network of recursive processes from which a coherent behaviour emerges as a global product of the system. The switch in perspective is neatly summarised by Di Scipio's suggestion that his ecosystemic approach represents " a shift from creating wanted sounds via interactive means, towards creating wanted interactions having audible traces." (Di Scipio (2003) p.271).

The notion of emergence is central to dynamical and Alife approaches to life and mind, and is an equally close to the hearts of Alife artists. Within Alife research and amongst those concerned with autonomous systems the concept of emergence as a process underpins the methodological approach to understanding, and attempting to simulate, life. As Langton put it in his inaugural speech:

> "The "key" concept in Alife is emergent behaviour. Natural life emerges out of an organised interaction of a great number of nonliving molecules, with no global control responsible for the behaviour of every part. Rather, every part is a behaviour itself, and life is the behaviour that emerges from out of all of the local interactions among individual behaviours. It is this bottom-up, distributed, local determination of behaviour that Alife employs as its primary methodological approach to the generation of life like behaviours" - Langton (1989), pp. 2-3

This bottom up approach deviates substantially from the standard design process of interactive music. As noted above, the channels of interaction are no longer constructed front-end interfaces, but slip streams into the internal dynamics which constitute the larger model. Recognition of this is apparent in Di Scipio's description of his construction of a performance network. "System interactions, then, would be only indirectly implemented, the by-product of carefully planned-out interdependencies among system components, and would allow in their turn to establish the overall system dynamics, upon contact with the external conditions."

2.4 Summary and Implications for Design

Consideration of these different frameworks provides a set of conceptual tools for thinking about and realising a more conversational style of interaction. Strong autonomy may not be achievable or desirable for the current project but provides a very useful set of metaphors.

On a conceptual level, the autonomy/heteronomy distinction provides a useful framework for understanding how a musician engaged in a musical dialogue with others can retain their individual identity, whilst being a part of a larger musical unit. We can conceptualise a cell as either an autonomous entity, structurally coupled to the biological environment of the body, or as a component functionally defined in relation to the larger organism. By extension we can see the individual musician as both a distinct musical personality and as an instrumentalist with a defined role relative to the ensemble of which they are a part. This points to the possibility of developing software processes that have their own musical agenda yet are able to collaborate with a human musician in an improvised performance. Rather than being a unit in the pipe-line model, the software algorithm and human musician are elements in a closed causal loop.

In very simple terms, changing the system boundary – separating human performer from computer system, or encircling both as a unified system – assists in matters of assessment. For some, the important thing is that the audience get a sense that the machine has its own musical agenda. For example Mari Kimura stated at a recent NIME workshop ″my job as a performer is to give the audience the impression that we are equal partners″. Talking specifically about Eric Singer's robotic guitar GuitarBot, with which she performed at NIME 06, she described some of the rules that defined the relationship between what she did on the violin and what the robot did. These consisted of things like: 'if the note is higher than E2, then play, else don't play'. The simplicity of such rule sets, she suggested allowed her as a performer to learn to play the system, and to create a sense of intelligence and intent on the behalf of the robot.

Figure 2.12: Mari Kimura performing with Eric Singer's Guitar Bot at NIME 2005

Other researchers feel that the real litmus test, and therefore the aim of the enterprise, must be that the performing musicians themselves gain the sense that the system has a musical voice and initiative of its own, that it instigates as well as responds to musical suggestions. Achieving this will often, although perhaps not inevitably, equate with the audience's engagement with the spectacle on stage.

In either case it seems crucial to consider the effects of the overall performance network as in the systemic perspective which is well illustrated by Di Scipio's AESI project. If we consider system design solely in terms of what happens between the input and the output of the digital system, we may fail to take into account the potentially rich effects of the real-world environment. These considerations are key in the embodied situated approaches to robotics mentioned in Section 2.2.3. In these areas, *behaviour* is defined as the observed agent-environment interactivity (in line with our every-day understanding), and *mechanism* is defined as the structure inside the agent which subserves this interactivity.

In biological systems, the important message is that behaviour is a product of the joint activity of agent, environment, and observer, so the (agent-side) mechanisms underlying the generation of any behaviour should not be assumed to be identical to the behaviour itself. An important consequence is that if a behaviour appears complex to an external observer, this does not imply that the underlying mechanisms are also complex. The classic illustration of this is the description Herbert Simon (1969) gave of an ant walking across a beach. The internal mechanisms of the (hypothetical) insect comprise simple obstacle-avoidance rules such as, if there is a rock or clump of sand to the left, go right and vice versa. The ant responds to every miniscule lump of sand, flotsum, jetsum and pebble, turning left and right and right and left as it negotiates the rough terrain. Simons points out that from the perspective of an external observer the trajectory traced by the ant is strikingly, and perhaps irretrievably, complex. This classic example serves to illustrate the possibility of achieving complex behaviours from simple mechanisms, something which is an attractive possibility for any designer of creative software. In a musical context, where the environment might be sonic, we can potentially generate behaviours that are not only apparently complex, but that are also contingent on current events in the sound world.

The situation where both the agent and the ant are capable of dynamically adapting is captured by the concept of structural coupling and puts forward an understanding of interaction that describes our experience of musical interaction with other human musicians in a much richer way than any models currently used within interactive computer music. The notion of coupling itself (where-by each system is a function of the conduct of the other) captures the sensation of togetherness experienced when playing. In free improvisation, and more subtly in an ensemble of scored parts, there is often no 'leader'. Whether subtle changes occur in expression, or dramatic changes in pace or texture, it is often impossible to pin down their origin.

The linear notion of cause and effect implicit in the predominant sense-plan-act software design fails to capture these dynamics which are central to the coherence of the group, and would be valuable characteristics of musical performance to move toward in artificial systems. This sense of ensemble can be achieved by attentive musicians who have never met, but we have all witnessed the phenomenon of a group who have played together over a long period of time: whether a string quartet, a rock band or a free improv group, there is something about the co-ordination of an established ensemble which belies their existence as individuals, it is as if their musical selves have somehow aligned, dragging their arms, fingers and minds with them. This type of structural congruity between autonomous entities is understood from an autonomous systems perspective as precisely the result of a history of interactions between autonomous systems, i.e. structural coupling.

Forms of these dynamic interdependencies can already be seen in Di Scipio's AESI , and perhaps in Blackwell's Swarms system. Di Scipio's performance network can itself be conceived as a dynamical system instantiated across digital and acoustic media. The closed causal loop defines an interdependency between each element and each can dynamically adjust according to reciprocal influences. Similarly, the low level rules of the boids algorithm underlying Blackwell's Swarm system defines every element in relation to each other, such that each adjusts to varying local conditions. In practical terms then, a dynamical bottom up approach can be used to place interdependent adaptive processes at the very core of the system. This facilitates the achievement of flexible and spontaneous form of interaction in a continuous, adaptive circuit that goes beyond the explicit switching of control implemented in systems such as Winkler's *Snake Charmer* and offers an alternative form of man-machine interaction.

The final important implication in adopting a dynamical perspective is the suggestion of an alternative to functional modularity. Recall that for dynamicists, cognition isn't the transformation of formal structures by distinct modules, but the temporally extended pattern of activity across the brain. Applied to the design of performance systems, this suggests that rather than designing separate interaction and composition modules, one process may be used to both subserve a response *and* generate musical material. The next chapter therefore presents an overview of the many different approaches to algorithmic composition, and considers the potential for adaptive dynamical techniques for generating musical material.