TUTORIAL ARTICLE

The evolution of interactive graphical control interfaces for music applications

PETER D. MANNING

Department of Music, University of Durham, Palace Green, Durham DH1 3RL, UK
E-mail: p.d.manning@dur.ac.uk
Fax: 0191 374 3219

This article provides an insight into the history of interactive graphical control interfaces for computer-based and computer-assisted music applications and identifies a number of the challenges which currently face contemporary designers of new software and hardware tools in this context. By studying some of the key features of this rich and varied legacy, it is possible to establish some useful criteria to underpin the development of new and improved tools for manipulating visual representations of music data which can be as diverse as common music notation, acoustic spectra, and all manner of simulations modelling the physical control surfaces associated with devices such as commercial synthesizers and audio mixing consoles.

1. INTRODUCTION

The burgeoning of multimedia tools and applications in recent years has radically transformed the working environment for composers of electroacoustic music, not least in the context of visual resources to support these processes. The use of interactive tools to generate and manipulate graphical representations of all manner of music data, from conventional score information in Common Music Notation (CMN) to spectral analyses of acoustic material, is now commonplace. Indeed it would appear the scope of such applications is limitless.

One of the most far-reaching concepts has been that of the virtual audio studio, where a traditional control surface such as the mixer, with its familiar array of physical devices such as knobs, switches and sliders, is entirely replaced by visual display technology, accessed interactively using standard WIMPs (Windows, Icons, Mouse, Pointer) resources in conjunction with a conventional computer keyboard. Central to the design philosophy is the proposition that the functionality of such control surfaces can be accurately simulated in an environment which is based on iconic representations of the various components. This is an ambitious objective which demands closer scrutiny in due course, for it focuses attention on perhaps the most critical consideration in the current context, the criteria which determine the functional relationships between computer simulations and their physical analogues. Before embarking upon a study of advanced applications such as the above, it is necessary to examine some basic operational principles in a more general context, that of conventional music notation and the printing of scores via a computer.

2. COMPUTER-BASED SYSTEMS FOR NOTATING AND PRINTING MUSIC

Perhaps the most familiar application of computer graphics in the field of music is that of desktop music publishing. Although this particular activity can be uncoupled from the processes of sound generation, it directly engages with issues of music data representation and processing which are of crucial significance to the development of multimedia tools for electroacoustic composition. Composers and musicologists alike now have direct access to a variety of sophisticated music notation software programs, capable in many instances of producing a quality of output on a modern PC and laser printer which rivals that of commercial music publishers. Prior to the mid-1980s or thereabouts the situation was radically different, with the majority of contemporary composers entirely dependent on hand-produced scores, either using their own manuscript or the services of a copyist. Although some music publishing houses have been prepared to continue investing in the services of professional engravers, this practice had already gone into sharp decline by the late 1940s, and with the development of desktop music publishing such traditional skills are now all but redundant.

This considerable legacy of handwritten scores, albeit a product of necessity rather than choice, fostered its own special culture. In particular, the freedom to customise the appearance of both scores and parts in terms of the precise positioning and intrinsic characteristics of each and every element inserted on the musical stave encouraged a particular intimacy between composers and their works, not least in terms of their acquired craftsmanship and attention to detail. The automated world of desktop music publishing, with its WIMPs-based repertory of standard music icons and supporting display control

Organised Sound **4**(1): 45–59 © 1999 Cambridge University Press. Printed in the United Kingdom.

algorithms to control key aspects such as note positioning, presents a very different kind of working environment for the traditional composer. The ability to standardise the appearance of so many features now becomes a virtue, with a prevailing culture which militates against the crafting of score details other than those which can be readily selected and pasted from the basic repertory of icons and functions provided by the software.

As notation software increases in sophistication, so the repertory of automatic functions to control layout reduces further the need for the individual to intervene in the processes of assembling the score from its basic components. The consequences of this trend are thus twofold: On the one hand an increasing degree of consistency is achieved in the appearance of music scores generated by a specific music notation software program. On the other hand, the incentive for direct user intervention in customising the final appearance of the score is reduced, and indeed the functional characteristics of the software itself may preclude any real exercise of discretion in matters of finer detail.

3. THE USE OF MIDI-BASED TOOLS FOR GENERATING MUSIC SCORES

The construction of music scores for printing in the manner just described is a highly controlled activity insofar as the user visually determines the choice and disposition of each new component to be added to the score, checking each action step by step. It has to be recognised, however, that the input of commands and data via the computer mouse and alphanumeric keyboard is a relatively slow and time-consuming operation, and there will be occasions where a much faster alternative mode of input will be desired. The most obvious means is direct data entry via a conventional music keyboard, but this solution is not without its problems. Although experimental work linking music keyboards to computers via a suitable interface can be traced back to the 1960s, it was the introduction of the now universal digital protocol for communicating music performance information between devices, MIDI, in 1983 which revolutionised this mode of communication. Unfortunately this protocol (erroneously known as the Musical Instrument Digital Interface), which has basically remained unaltered to the present day, suffers from a number of operational deficiencies in the specific context of data capture for music printing, which compound further some very real difficulties associated with any methodology which seeks automatically to transcribe performance data into an equivalent CMN score.

Considerable progress has been made in recent years in developing data formats specifically for the representation of musical notation, most notably NIFF (Notation Interchange File Format), completed in 1995, which provides a standard protocol for transferring all the details of a CMN score between different music printing systems in a standard format. NIFF, however, has yet to be approved by all the leading software manufacturers and does not in itself provide any further solutions to the basic problems of performance practice which impede this mode of transcription. In truth, a significant proportion of the problems encountered in this context are as much to do with the limitations of CMN itself as a representational medium for music performance data than any intrinsic limitations of computer graphics as a means of displaying such information in a visual format.

The processes of encoding MIDI performance data are already reductive in the sense that vital details of the synthesis processes themselves are not directly captured, only the higher level control information necessary to reactivate the individual voice generators. At the most fundamental level, a basic MIDI message consists of three bytes of binary-coded information, the first being concerned with the status of the data to follow, in this context 'note on' at the start of a sounding event and 'note off' at its termination, the second with the tempered pitch value of the note in the range 0 to 127, where middle C corresponds to the value 60, and the third is a measure of volume in terms of the velocity of the key stroke which initiated the note message, also registered in the range 0 to 127. If the timing and duration of each note is also to be recorded, suitable time-stamps have to be generated for each 'note on' and corresponding 'note off' message as additional items of information.

It is clear from even such a cursory study of the format of a basic MIDI event message that there are a number of problems to be surmounted in converting this information into an equivalent score. Even such a basic consideration as the pitch of each event can present major problems, since common music notation requires the use of additional descriptors for the complete set of twelve pitch classes in the tempered scale, the choice and use of which is entirely context-dependent and influenced by the prevailing key signature. MIDI pitch code 61, for example, does not differentiate between C♯ and D♭, and indeed as far as any keyboard performer is concerned, these notes are associated with the same key. This provides a specific instance of a situation where music data captured via performance cannot provide sufficiently detailed information for the reliable extraction of equivalent music notation information.

In the case of the other features provided by this basic mode of MIDI data capture, the reverse is true. Although it is the case that the subtle variations in the timing and articulation of these note/events introduced as part of the art of performance transform an otherwise mechanical reproduction into one which is musical, these are matters of interpretation which are not fully quantified in a CMN score. Such features, both as regards timing and also other aspects of articulation, can only be indicated via a strictly limited set of additional markings which in many instances merely provide general clues as to what is actually required. Interpretation of phrasing slurs, for example, may involve all manner of subtle variations in such features, which will differ in execution from performer to performer. Dynamic marks are also merely indicative in the sense that they do not define the actual volume of the associated note/events. The interpretation of this descriptor is invariably context dependent, influenced by other considerations such as the timbre of the sound and its tessitura.

Given these representational restrictions, it becomes very clear that even the most sophisticated of MIDI data capture/CMN generation software programs can only achieve very limited and incomplete representations of performance data. If the primary objective is to produce a representative score in a conventional CMN format, then all timing variations in terms of the onset and release of each component note/event relative to the basic pulse and suitable subdivisions thereof have to be quantised to the nearest appropriate values. Although the quality of these supporting tools has improved considerably over the years, to the point where some quite intelligent decisions can be made automatically according to the speed and density of the incoming realtime MIDI data stream, performers will invariably choose to key in material as mechanically as possible in order to reduce the opportunities for transcription errors still further. This detachment of a visual representation of music data from its performance characteristics has led to what initially might seem a rather curious facility in some generally available music notation programs, whereby it is possible to retain all the performance control information embedded in a MIDI data flow for the purposes of audio-proofing, whilst at the same time displaying a quantised representation of the basic note/event information in CMN which can then be independently crafted using conventional WIMPs facilities.

This process of making a virtue out of necessity can be interpreted in different ways. For example, it allows the user to retain an unmodified aural map of the source information when using manual editing facilities to craft a definitive visual CMN score, ready for printing. Once such a score has been realised in a conventional hardcopy format, it becomes totally separated from its origin, the task of resynthesis passing to independent performers who will interpret the information in a variety of ways, using such electronic or acoustic instrumental resources as they may decide is appropriate. Rather different principles comes into play if, instead of printing out a conventional score, the displayed CMN data is directly converted back into MIDI data, in turn fed to the voice bank of an associated synthesizer. Even if precisely the same voices which were used for originally keying in the data are employed for this resynthesis process, the acoustic results will be significantly different, for the reasons discussed above.

Such operating conditions are all too familiar to electroacoustic composers who have made extensive use of the CMN data entry facilities provided by almost all sequencing software programs. They also extend beyond this manual entry mode of specification to the general use of quantising routines not simply to rationalise CMN displays of performance data, but to tidy up the performance itself. Partial solutions to these adverse effects of performance quantisation have been developed by the designers of sequencing software in the form of functions which will 'groove' quantised data by automatically reintroducing an element of expression in terms of superimposed variations in event timings and/or attack velocities. Some routines go a stage further with facilities which attempt to interpret additional details which may have been coded in CMN, for example articulation and dynamic markings. Such resources, however, can only provide generalised responses which will inevitably lack the refinement of human performance through interpretation, a practice which can freely and creatively take into account broader contextual considerations.

The very fact that such facilities are necessary to revitalise a CMN transcription reinforces the practical limitations of CMN as a visual tool for controlling the generation and processing of acoustic material. In particular, CMN tells us nothing of value about the sonic characteristics of the note/events themselves. We thus have no option but to come to terms with a situation where a skilled musician will instinctively know from training and experience how to interpret the elements of a conventional score and convert them into an acoustic performance, and yet many of these crucial characteristics cannot be identified, let alone automatically generated, from the visual information contained in the score itself. Looked at from another viewpoint, the fundamental principle of WYSIWYG (what you see is what you get) which underpins the visual dimension of a multimedia computer system cannot, at least in the case of CMN, be extended to the notion of WYSIWYH (what you see is what you hear).

The reasons why CMN-based graphical interfaces, for all their shortcomings, are so commonly encountered in computer-based or computer-assisted sound synthesis systems is largely due to a different, and ultimately superior factor. This concerns the requirement that any representational system should directly relate to the experiential knowledge of the user, and the closer this correlation, the more effective is its immediate value. It is precisely because experienced musicians without any further training or assistance can so readily interpret the significance of such music data as a definitive set of instructions for a generative acoustic process that its ultimate value as a tool for electroacoustic composition becomes so strongly assured. It also follows that any move away from this standard representation of music data, perhaps to include elements which more accurately describe the physical characteristics of the associated sounds, will inevitably undermine this long-established link.

4. THE DEVELOPMENT OF ALTERNATIVE MODES OF MUSIC DATA REPRESENTATION

The notion of modifying CMN to incorporate additional representational elements in the context of electroacoustic music is by no means new. Pierre Schaeffer, for example, investigated such possibilities in 1950 as part of his early research into a *solfège* for his *objets sonores*, but quickly realised that this methodology alone could not fully address important issues such as the detailed description of timbre. Such ideas were nevertheless revisited by the research team responsible for the Structured Sound Synthesis Project at Toronto University during the late 1970s (Buxton 1978, Buxton, Sniderman, Reeves, Patel and Baeker 1979). Perhaps more than any other pioneering venture of its time, the practical outcome of this project, the SSSP digital synthesizer, complete with a multidimensional control interface which integrated standard and custom-designed physical control devices with an interactive graphics toolbox, firmly established the basic operational framework for many of the multimedia sound synthesis facilities in use today. Given the strategic importance of this pioneering venture, it is thus instructive to consider further some of its primary design objectives.

In developing an iconic system of data representation, considerable thought was given at the initial design stage to how the basic principles of CMN might most usefully be adapted and modified to serve the needs of the electroacoustic composer. The resultant approach was quite radical in a number of respects. For example, many of the descriptive features of CMN were completely dispensed with, along with the expectation that notes should be quantised and mensurated as a matter of course. Instead, the music stave was simply regarded as a time line with a vertical grid indicative of pitch, onto which neumes indicating note events could be inserted at the point of onset, the choice of duration values in CMN being merely indicative of the approximate extent. Variations in basic pitch during the course of a note could then be superimposed in the form of a continuous pitch line extending from the neume for its actual duration as measured on the time axis.

These partial representations of data in CMN were supported by two other graphical representations of the same information, including a piano-roll type of display plotting the base frequencies of individual note/events against time. Such pitch maps are now routinely provided by modern MIDI sequencing software as an alternative to CMN data displays, sometimes augmented with parallel displays of the associated attack velocities plotted on the same horizontal time axis. The third mode of data display, an attempt to provide an extra dimension by way of a tree structure which identified the individual synthesis control structures, crossed an important boundary in terms of the level of detail which may be displayed and altered interactively, for it allowed direct access to the synthesis processes themselves via the nodes of the tree.

This latter level of integration has rarely been matched in the MIDI environment. Commercial, academic and shareware graphics-based voice-editing software has been developed for a number of MIDI synthesizers, notably earlier Yamaha synthesizers such as the DX7 and the SY77/SY99 and a number of drum/percussion instruments, but operational considerations, in particular the need to switch to MIDI ‘system exclusive’ mode to allow the transmission of voice data between the computer and the synthesizer, inevitably leads to an enforced separation of the two processes. In the case of the SSSP system, the freedom to move directly and logically between the various levels of parameter control was a fundamental design feature.

Four primary modes of synthesis are supported by the hardware: (i) additive synthesis, where notes or other sound complexes are assembled element by element from up to sixteen individual sinewave components, each of which may be controlled individually in terms of both frequency and amplitude; (ii) fixed-waveform synthesis using up to eight different representations of compound timbres stored as wavetables; (iii) frequency modulation or FM, where by modulating the frequency input value for one audio oscillator (the carrier) with the output waveform of a second oscillator (the modulator), a variety of complex spectra can be generated (Chowning 1985); and (iv) VOSIM (VOice SIMulation), a technique for generating quasi-speech sounds by means of pulsed trains of single sine-squared waves (Kaegi and Tempelaars 1978). Crucially, the entire repertory of synthesis resources could be directly accessed and edited using the same graphics toolbox as that used for performance data.

The above reference to a boundary between performance instructions for a particular synthesis

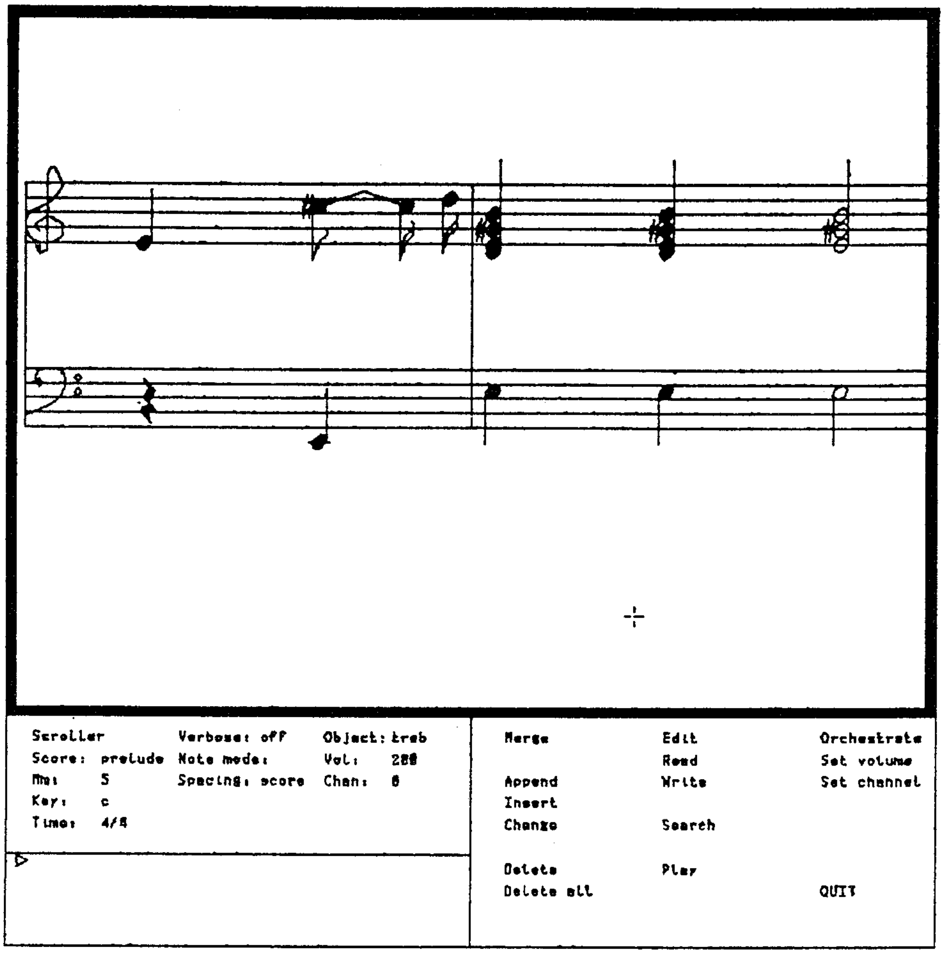

Figure 1. Representation of score in common music notation (Buxton 1978: 74).

engine and instructions which determine the functional characteristics of the engine itself is essentially an artificial one, more often than not determined by the functional limitations of the performance interface itself, hence the distinction which has just been made between realtime MIDI data transfer in a performance mode and the transmission of device-specific information for the voice generators in a purely editing mode. To the electroacoustic composer there is no valid basis for making such a firm distinction. What is more important, especially in a graphics-driven environment, is the nature of the control processes which are thus accessed, and how their functionality relates to the user's understanding of the particular synthesis methodology.

The SSSP designers fully recognised the problems which are encountered here, in particular the fact that it is often very hard to make a clear distinction between data and process, the most obvious example being the traditional notion of an orchestra, which identifies the available sound resources, and the score which determines how these resources are to be used. What follows from this is the need to recognise that data has to be realised at several conceptual levels, and that the user must be able to move between these different levels without difficulty and with a clear understanding of the significance of each mode of representation. When designing a graphics-driven control facility for an additive synthesis engine, for example, it is important to bear in mind that some functions will relate more readily to alternate modes of representation than others. Obvious candidates in this particular context are general characteristics such as the overall envelope of a single note/event, i.e. the nature of its attack, sustain period and decay, and its fundamental pitch.

As soon as the processes of specification proceed to a deeper level, for example the choice and disposition of individual frequency components and their associated amplitudes which collectively define the timbre for a single voice, then the opportunities for such an immediate correlation start to break down. What is required here is a conceptual level and an associated mode of information display which can more usefully link the components of sound spectra as functions of time with the perceived results. In the absence of any existing criteria which may be easily drawn from our general music experience, such tools

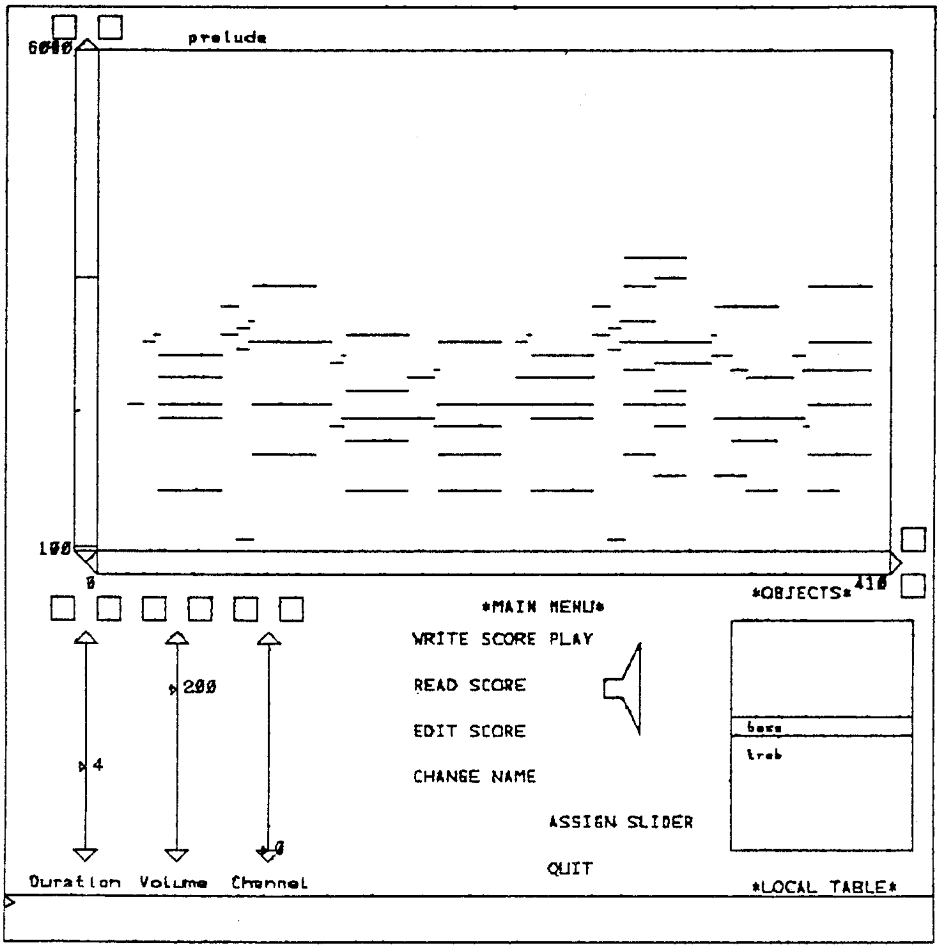

Figure 2. Representation of score using 'piano-roll' type notation (Buxton 1978: 74).

have to be invented, and the user suitably trained in their use. More often than not, however, in such circumstances composers are left to develop their own terms of reference entirely unaided.

The problems of correlation take on yet another dimension where the parameters controlling the synthesis process do not directly relate to the actual results. Perhaps the most widely known example of this is the earlier-mentioned technique of FM synthesis, where the underlying mathematics which determine the acoustic spectra resulting from a particular choice of carrier and modulating frequencies and associated modulation indices, even for a simple FM oscillator, are so complex that it becomes exceedingly difficult to construct any really useful reference facilities for the prospective user. Where interactive graphics display facilities do come into their own in this particular context is at a macro level, building the basic structure of more complex FM oscillators which involve a number of individual components. Such visual representations using icons for the various elements can of course be extended to all manner of synthesis and signal processing procedures, including those involving additive synthesis. It still has to be borne in mind, however, that no matter how sophisticated the graphics interface, the visual information provided in this context is essentially that of a control panel, and therefore only provides limited clues to what might be the true nature of the resultant sounds.

5. THE VISUAL REPRESENTATION OF ACOUSTIC DATA

In seeking visually based mechanisms for representing acoustic (as opposed to process) control material in a format which relates more closely to what we actually perceive, we are faced with an even deeper set of problems which concern the way in which we conceptualise sound information, and how this can be usefully related to the processes by which such phenomena are actually created. Our powers of visual and aural perception both outwit the current capabilities of multimedia computing systems and are also all too readily defeated by them. Skilled musicians can not only identify individual elements within a composite musical texture as discrete components, but also, at least in the case of Western

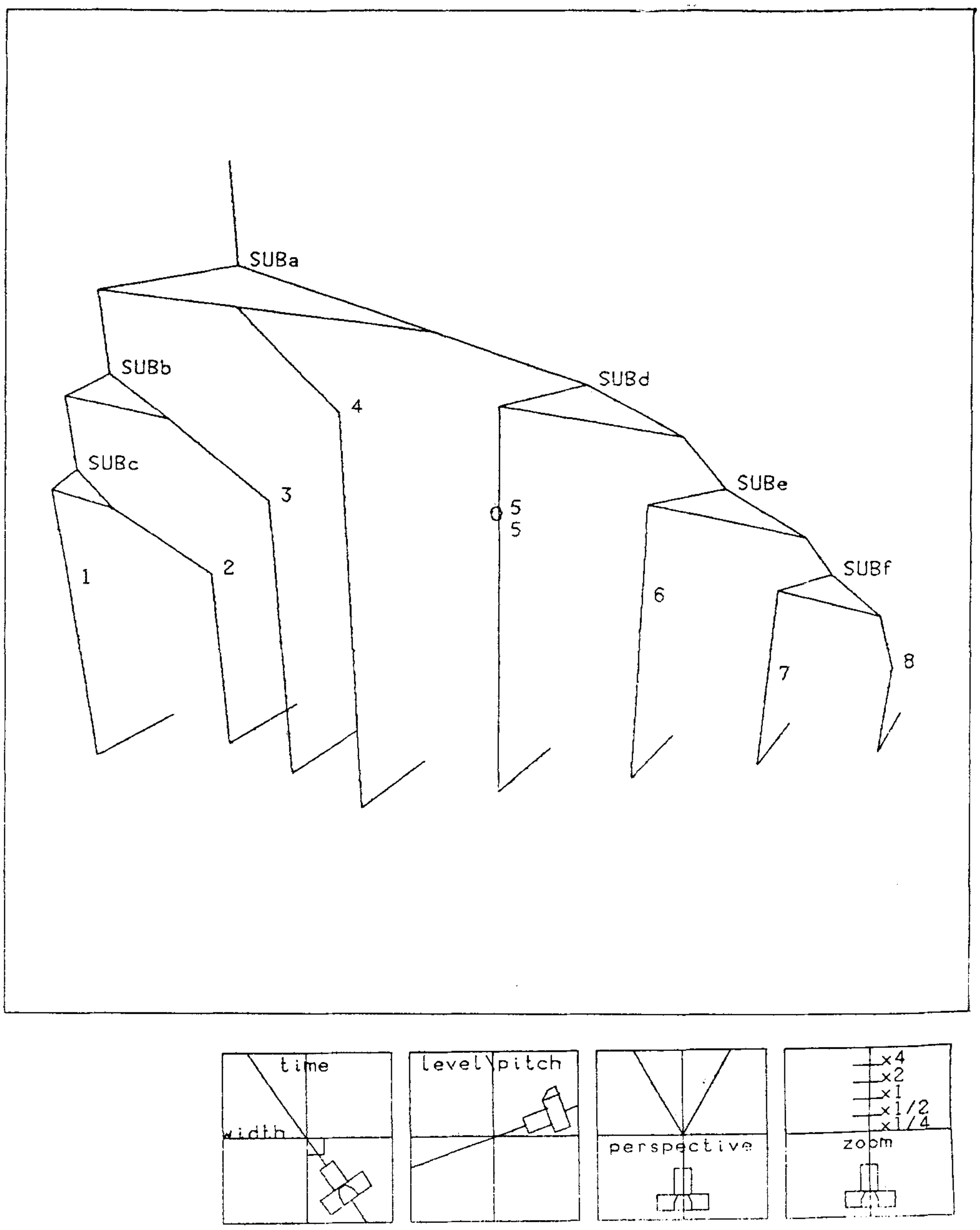

Figure 3. Three-dimensional tree representation of score (Buxton 1978: 75).

music, directly transcribe this information into a visual representation using CMN. As those researchers who have investigated various methods of acoustic pitch tracking over the years are only too aware, the problems encountered in realtime digital signal analysis of the pitch changes in the acoustic signal of a monophonic orchestral instrument source become close to impossible when such information is masked by a supporting polyphonic texture.

Against this may be set the powerful capabilities of the range of advanced software tools which are now available for the analysis of acoustic information, with a capacity, for example, to isolate the individual components of complex audio spectra in a manner and to a resolution which far exceeds the analysis capabilities of the human ear. Such information can be displayed in a variety of formats such as Fourier transforms in the time or frequency domain, and the data thus extracted used as source material for a growing range of powerful signal-processing algorithms now available for electroacoustic composition.

These spectrum analyses, however, produce a number of conceptual problems for the observer since they do not obviously relate to any normal musical experience. These problems increase the moment observation becomes interaction, using graphics tools to modify both the display and the associated data. Software tools which allow spectral displays of acoustic signals to be interrogated and modified in such a

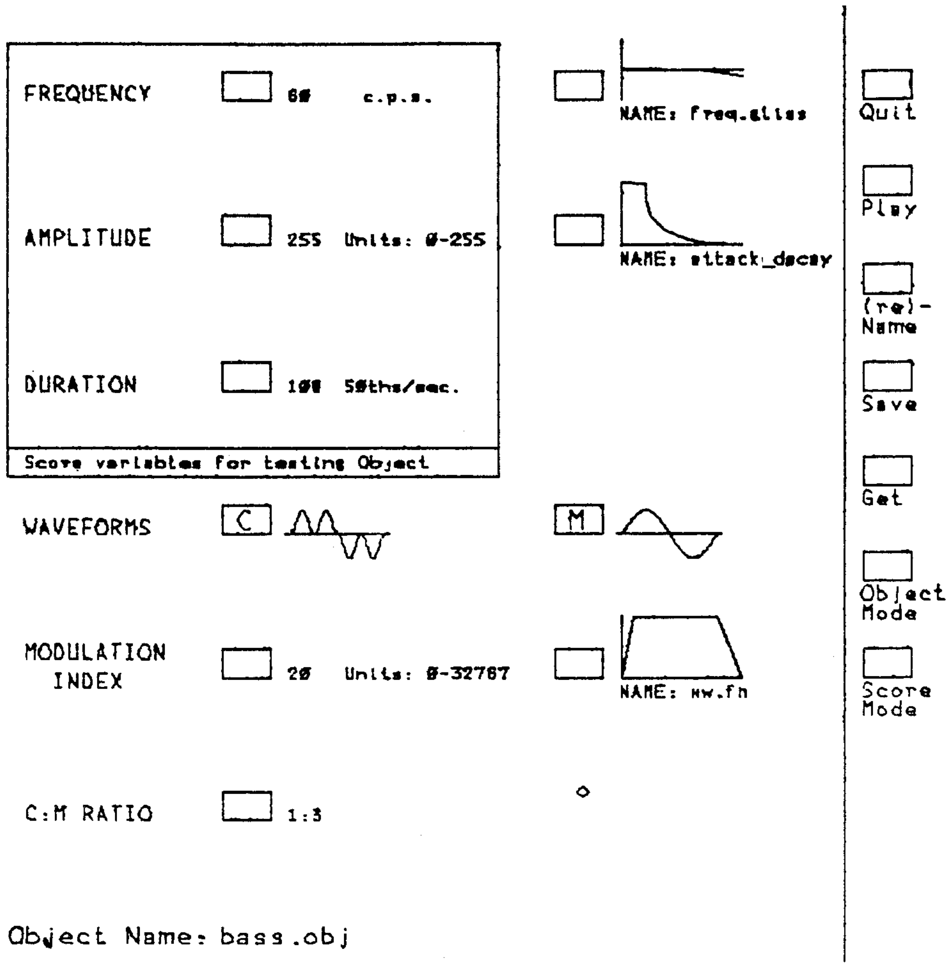

Figure 4. Representation of an FM object at the *acoustical* level (Buxton 1978: 76).

manner, for example Audiosculpt from IRCAM, can provide an exceedingly useful means of digital signal processing in the hands of a skilled user, but even resources as sophisticated as these still cannot facilitate the identification of the elements which constitute a single instrumental component within a complex orchestral texture, let alone their physical extraction. Here, however, another factor comes into play since it is the limitations of the analysis process which impedes such an objective, and not specifically any limitation of the graphics medium itself.

Replicating the processes of neural analyses which allow us to achieve such an acoustic separation in many instances requires computing resources which go way beyond the basic identification of component frequencies and amplitudes in a composite audio spectrum. Such research is still very much in its infancy. In an earlier example concerned with the capture of control data for MIDI instruments and its conversion into a visual mode of representation using CMN, the basic assumption was made that such transcriptions will generally provide sufficient information for the observer to establish a significant correlation between the score and the sound. In the current context, where access to underlying synthesis control information is simply not possible if the analysis method deals exclusively with acoustic information, the observer will have to rely on alternative criteria for constructing any meaningful visual representation of sound, bearing in mind what may be very real limitations imposed by the underlying processes of data analysis.

The need for alternative representations of sound information is reinforced further in the context of electroacoustic music by the fact that the synthesis of traditional instrumental sounds forms only part of the overall scope of activities. Many composers make extensive use of abstract sound images which possibly may only be quantifiable in spectral terms, and whether they are working indirectly at the control level or directly at the signal processing level, they therefore require special tools which will coherently relate both to their compositional ideas and also the resulting sound material.

Such a quest goes to the very heart of the issue identified at the outset regarding the functional relationships which may be profitably developed between sound and its visual representation. Matters become further complicated by considerations such as the relationship between the science of acoustics

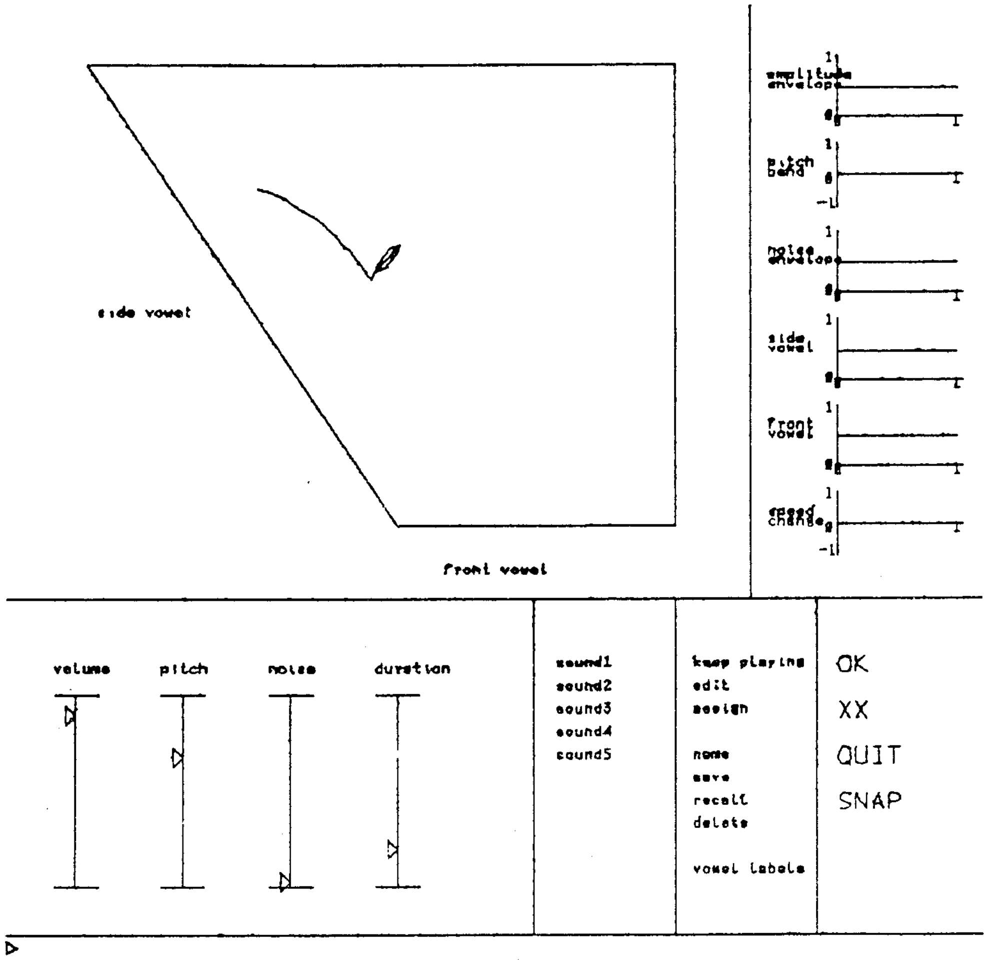

Figure 5. Representation of a VOSIM object at the *perceptual* level (Buxton 1978: 76).

and that of psychoacoustics, where various degrees of subjective interpretation come into play in an attempt to bridge the gap between our perception of sound and its physical representation.

6. PHYSICAL AND PSYCHOACOUSTICAL REPRESENTATIONS OF MUSIC DATA

Studies of these different modes of representation and the possibilities of linking them usefully together are far from new. Reference has already been made to the work of Pierre Schaeffer initiated some fifty years ago in his quest for an all-embracing typology and morphology of sound. His extensive investigations have laid important foundations for further research and development in the field of graphic-based tools for sound synthesis and processing. For example, his extensive writings on the subject, from *A la recherche d'une musique concrète* (Schaeffer 1952) to *Traité des objets musicaux* (Schaeffer 1966), provide a deep insight into some of the most intractable problems of sound classification and representation.

Those who set out to design such tools have much to learn from his appraisal of fundamental issues such as how physical analyses of acoustic material, measured in terms of pitch, frequency and time components, can be rationalised in terms of descriptors, both text-based and visual, which coherently link objective and subjective attributes in a musically meaningful manner. Most significantly, Schaeffer challenged the common notion that technology can somehow remain an entirely neutral and passive agent in the processes of sound analysis and resynthesis. In thus recognising its crucial role in shaping and controlling the ways in which we manipulate sound, he provides an important framework for those who seek to design new or improved tools for the electroacoustic composer.

A considerable debt is owed in this context to the subsequent work of François Bayle, who in his capacity as director of Schaeffer's Groupe de Recherches Musicales (GRM) from 1966 to 1996 attached a particular priority in terms of research to the development of multimedia tools for shaping the morphology of sound objects. As a further extension of this work, Denis Smalley has similarly spent many years refining and extending Schaeffer's theories to create his own toolbox for exploring sound shapes in

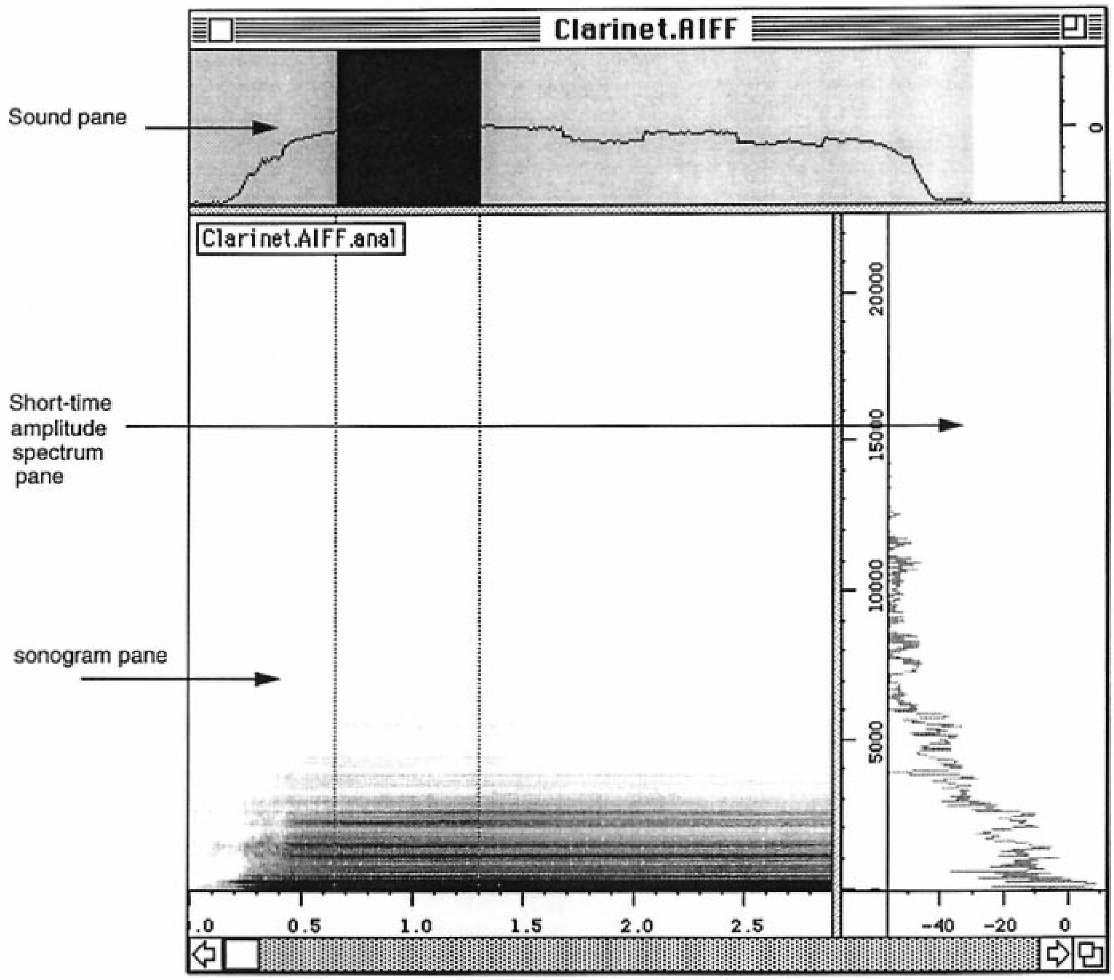

Figure 6. Sonogram of a clarinet note, using AudioSculpt. Reproduced with the authorisation of IRCAM – Centre Georges Pompidou. Image from AudioSculpt, ©IRCAM, registered trademark of IRCAM.

terms of what he has defined as spectromorphology (Smalley 1997), and there is much which can be usefully learnt from the work of others in this context, for example Trevor Wishart, who in the course of developing his repertoire of electroacoustic compositions has published an extensive treatise on sonic art and the principles which lie behind the construction of his own highly developed morphology (Wishart 1996).

7. THEORY INTO PRACTICE: THE DEVELOPMENT OF GRAPHICAL INTERFACES FOR AUDIO SYNTHESIS AND PROCESSING APPLICATIONS

Turning theory into practice in terms of developing more effective graphics tools for handling audio information in a multimedia environment poses some major challenges for contemporary system designers. There are some important issues which need to be highlighted by looking more closely at some features of the currently available facilities for managing digital sound synthesis and signal processing tasks, and the factors which have influenced their development.

The evolution of computing technology has occurred at a pace and in a manner which has radically altered the rules of engagement as regards its musical application. During the pioneering era of music computing, from the early experiments of Max Mathews in the 1950s at Bell Telephone Laboratories (Mathews 1969) to the birth of MIDI in the early 1980s, the pace of development was altogether more measured, and for the most part restricted to those academic institutions willing to support the costs of providing access to expensive computing resources and associated specialised digital peripherals such as digital to analogue converters for direct audio output or interfaces for the control of external synthesis equipment.

Accessibility was considerably improved during the 1970s with the development of much cheaper stand-alone minicomputers which could be dedicated to music applications, but such facilities were severely constrained by the lack of adequate processing power, both in terms of speed and also memory capacity. The scope for interactive working was thus very limited, and in the case of software audio synthesis and digital signal processing applications, almost nonexistent. Computer graphics facilities, if available at all, were very expensive, of limited capabilities and poorly supported in terms of software design tools. This factor alone makes the achievements of multimedia pioneers such as the members of the Structured Sound Synthesis Project all the more significant. This situation was to change dramatically with the birth and rapid development of the personal computer towards the end of the decade. This challenged the traditional alphanumeric culture of the mainstream computer industry with an increasing emphasis on the graphics-driven environment which is now universal for almost all general-purpose computer platforms.

What is sometimes forgotten by contemporary practitioners in the field of electroacoustic music is that the driving force for this particular development has been an overarching desire to provide visual imaging facilities for all manner of computer applications, from desktop publishing to computer-generated art. The development of iconic functions for music applications, be they in the context of music printing, MIDI or audio synthesis, thus forms part of a much larger field of multimedia activity. The modern high-level resolution colour monitor with its dedicated memory and extensive toolbox of graphics-generating algorithms is well matched to the power and processing capabilities of the associated computing engine, and the transition from an environment where technology has traditionally imposed significant constraints on what can be achieved in visual terms to one where such limits are for the most part determined by the skills of the software designer and the creative imagination of the user, is almost complete. Such restrictions which still exist are almost invariably attributable to the underlying computational process itself.

In highlighting a number of instances where the art and practice of music has benefited from graphical aids, the earlier discussion has made a number of assumptions which must now be re-evaluated, and in some instances challenged. One such assumption is that the development of an interactive WIMPs environment is the most natural and creatively productive means of linking an artist with a computer-based or computer-controlled set of tools. With the rapid development of ever more powerful and sophisticated tools for multimedia applications, there are risks that such an assumption becomes an imperative. In other words, it is the versatility of the technology which shapes the working environment for the musician, rather than the reverse.

In the case of electroacoustic composition, tensions will invariably arise between the creative aspirations of the individual and the functional characteristics of the available tools. In the past the very existence of operational limitations has often served to influence and focus the composition of electroacoustic music in distinctive and productive ways. Thus the characteristics of Pierre Schaeffer's early studio at the Club d'Essai in Paris are as unique as are those of a voltage-controlled synthesizer from the 1970s or an early digital synthesizer from the 1980s, and the repertory of works produced exclusively using such systems clearly reflect this. There is, however, an important, if often hard-to-define, boundary point in the development of technology where the notion of functional limitations changes to one of unknown possibilities which ultimately extend beyond the comprehension of even the most experienced electroacoustic composer, and it is here that the greatest dangers have arisen for those charged with the responsibility of developing new resources, or indeed substantially modifying the operational characteristics of established ones.

An interesting illustration of the latter phenomenon has been the evolution of the MusicN family of software synthesis programs, from the prototypes developed by Max Mathews at Bell Telephone Laboratories in the late 1950s to the various versions of CSOUND, originally developed by Barry Vercoe at MIT (Vercoe 1986) and now widely available to PC users as shareware over the Internet. Although the range and versatility of the functions has developed extensively over the years to embrace both synthesis and signal processing tasks, the fundamental principles of a set of descriptors for an orchestra of instruments, in turn activated by performance data from an associated score, have not changed at all.

When CSOUND was first released in 1986, the prospects of operating the program interactively and in real time, at least on conventional computing platforms, still seemed a remote possibility. Such has been the quantum leap in processing power since that date that today increasing numbers of CSOUND users are able to use the software in a realtime mode, in some instances supplying the score control data via MIDI, using a MIDI-to-CSOUND data converter. Such a working environment is in marked contrast to the early pioneering days when even a relatively modest set of synthesis instructions could take several minutes or even hours to compute. Such conditions firmly militated against any experimentation by trial and error, and the path to creative success lay in the disciplined study of the operational characteristics of each synthesis algorithm and an entirely predictive approach to their application as compositional tools.

In such circumstances, the requirement to specify every detail of both the orchestra and the score in alphanumeric code was both appropriate and also quite acceptable as part of the overall composition process. Indeed, the ability to exercise such a fine degree of control over every aspect of the synthesis task in ways which were simply not possible with commercial synthesizers was seen by many as more than adequate compensation for the inability to run the program interactively and in real time. Now that the primary impediment to these objectives (the basic lack of general computing power) has effectively been eliminated, attitudes have changed markedly, especially amongst those coming to use CSOUND for the first time. The continuing requirement for both the orchestra and also the score to be specified in conventional alphanumeric code is now seen as cumbersome and frustrating, hence the desire for front end multimedia editing tools, with a strong emphasis on graphics support.

In the quest for such enhancements, a number of factors need to be carefully considered. Amongst the most pertinent are the following: (i) Does the interpolation of an interactive graphics interface between the user and the command language for the synthesis engine distance the former in any material sense from the underlying functionality of the system? (ii) Is it really the case that the system being thus enhanced by such an interface is functionally suitable for such a mode of operation? If it is accepted that the answer to the first question is arguably yes, and to the second arguably no, then the implications are potentially far-reaching and contentious. Nevertheless, many developers and users of CSOUND, including the author, will not be deflected from their quest to enhance the accessibility of such software by means of such interactive tools. The ultimate challenge, as already noted, is to ensure that the tools themselves meet criteria which match the technology to the needs and aspirations of the composer, and not the reverse.

The starting point for any rational assessment of an interface design is the fundamental relationship to be established between the user and the system. The control of a creative environment, whether that of the composer or the visual artist, is very much bonded to the real, analogue world where objects are physical entities, and where physical movements, especially of the hands, provides the most natural means of interaction and expression. Amongst the most striking examples of spatial hand-operated sensors for controlling processes of sound synthesis are the succession of devices developed at STEIM in the Netherlands by Michel Waisviçz from the mid-1980s onwards, in particular the Hands (Krefeld 1990). These special hand gloves contain multiple electronic sensors which not only detect the movement of both hands in space in terms of their proximity to each other and their overall trajectory, but also the individual movements of each finger. Such a powerful conduit from the functions of the brain to the processes of synthesis in performance via multiple human-operated motor functions defies any meaningful replication in terms of an equivalent mouse-driven computer graphics interface, where individual functions can only be interrogated and modified one at a time by two-dimensional movements of a single pointer. The Radio Baton developed by Max Mathews (Mathews 1991) is functionally simpler than the Hands, but is based on much the same basic principles of registering hand movements in space, with all the practical advantages of a three-dimensional physical control environment.

The above examples of custom-designed performance controllers are perhaps extreme examples against which to benchmark the capabilities of a graphics-driven control interface. A fairer comparison, perhaps, would be a WIMPs simulation of the control panel of a modern digital synthesizer. With the latter device we encounter modelling of a physical interface which itself has already been subject to major rationalisation and hierarchical organisation in the digital domain. As far back as the first all-digital MIDI synthesizers such as the Yamaha DX7, launched in 1983 with an elementary graphics display and associated set of programming push buttons, this dramatic change of philosophy in terms of synthesizer interface design was all but complete. In the previous era of voltage-controlled synthesizers such as the Moog, Buchla or VCS3, all control functions were immediately accessible through an array of knobs, switches and patch chords or shorting pins. Users, therefore, can immediately assimilate the current settings of each function visually and effect changes immediately by appropriate movements of one or both hands simultaneously. In the case of most hardware digital synthesizers, their functions for the most part can only be accessed and adjusted one at a time, making it very hard for the operator to keep a mental track of other possibly interdependent settings whilst monitoring and adjusting what may be a very limited set of displayed parameters.

Improvements to the design of synthesizer control panels have not been universal or particularly extensive, and it is evident that market forces are far from helpful in this respect. Where the vast majority of synthesizer users are content to rely upon preset voice banks, and have neither the time or inclination to modify their performance characteristics in any material way, there is clearly no real incentive on the part of manufacturers to provide more sophisticated control interfaces. Such restrictions can only be effectively bypassed where the manufacturer allows remote access to most if not all the parameters which control the underlying synthesis algorithms themselves, and as noted earlier the scope for such remote modelling, even in MIDI 'system exclusive' mode, is often severely restricted. Nevertheless, the potential of an enhanced computer-graphics environment in this context, with the ability to display a number of functions simultaneously and move directly from one interactive process of data entry to another by the simple click of a mouse, is a powerful incentive for those prepared to write the necessary software.

One of the most complex issues to emerge in assessing the functionality of an interactive graphics interface is the extent to which the operation to be carried out is preparatory in terms of specifying and adjusting the initial architecture and parameter settings of a synthesis or signal processing engine, or operational in terms of changing one or a number of parameter settings as functions of time when the engine is actually in use. In practice, many operations fall some way in-between, for example the processes of adjusting the basic amplitude envelope for a synthesizer or sampler voice is essentially a preperformance initialisation operation, but the immediate feedback loop associated with monitoring and aurally verifying each parameter adjustment is arguably a performance function in its own right. The critical boundary in terms of any coordinated visual/aural/manual interaction is the point where the operator, in his or her judgement, is unable to assimilate the information provided by the input stimuli and produce an appropriate physical response via the control interface.

Such perceptual boundaries will vary from person to person, and also are to a large extent context dependent. There is, however, a pressing need for further research to be carried out in this area, not least in determining more useful benchmarks for those who design computer graphics control interfaces for music applications. One very simple and probably familiar example of the disparity which can arise between a mouse-driven control environment and its physical analogue is the shuttle or scrub facility offered by many software audio editing programs, for example Digidesign Soundtools, where a special marker is superimposed on a visual representation of the acoustic signal which is thus being accessed. When the mouse is used to drag this supposedly spring-loaded marker in either direction from its central position, the effect is to move the display image and the associated sound data in the direction indicated; the further the amount of drag, the faster the movement. The intention is to replicate the traditional shuttle control provided by a professional analogue tape recorder, a facility which has hitherto proved invaluable as a quick editing aid when determining the precise point at which a recording is to be cut and spliced. Simply moving a mouse left and right on a mouse pad, however, in no way replicates the feel of a spring-loaded mechanical shuttle, and such computer simulations are very poor substitutes for their physical counterparts.

The fact that the WIMPs environment allows acoustic signals to be displayed as a visual function provides only limited compensation in this particular context, but there are clearly other editing situations where such resources offer a functionality which far exceeds that associated with conventional analogue tape editing. The ability to isolate and freely juxtapose sound segments whilst at the same time using digital signal processing resources to modify their characteristics is a good example of the superior capabilities of digital sound editing tools when used in conjunction with a good interactive graphics interface.

8. THE CHARACTERISTICS OF THE WIMPS-DRIVEN VIRTUAL AUDIO STUDIO

This comparison of the merits of physical and graphical modes of interaction with audio equipment leads the discussion back to the opening observations concerning the ambitious nature of the proposition to design a virtual audio studio, thus eliminating the need for any analogue control devices external to the computer. It is now possible to draw together many of the various issues which have arisen during the course of this study and put this proposition formally to the test by considering in some depth the operational characteristics of a sophisticated professional graphics-driven multichannel audio mixing system such as Digidesign Protools. The fundamental concept behind a system such as this is the simulation of all the functions of a conventional analogue mixing and recording studio, within a digital computer, suitably enhanced by add-on processing and communications hardware and a number of additional digital signal processing functions such as filtering and reverberation. A key feature of the operating software is the use of a WIMPs interface as the primary means of monitoring and controlling the access and manipulation of sound information.

The craft of audio mixing is undoubtedly an actively managed process which operates, however, at a number of levels of user interaction. In setting up such a system for a recording there are a number of processes of initialisation to be carried out, for example the assigning and routing of input and output channels and the calibration of basic signal levels. As an intermediate step, initial adjustments may be made to features such as panning controls and channel equalisers, some of which may retain these settings whereas others may be subject to realtime adjustments during the recording itself. Since all the initial adjustments are usually carried out sequentially in a conventional mixing environment, the step-by-step procedures used to interact with a WIMPs interface would seem perfectly compatible. In operational terms, however, some compromises are necessary and inevitable. Rotary knob movements, for example, are extremely hard to replicate in a graphics environment. It is feasible to design a circular icon to represent a knob with an extending position indicator which can be rotated clockwise or anti-clockwise via the mouse pointer, but the associated circular movements of the mouse on its pad feel very unnatural and cumbersome to the operator, especially when such actions are compared with the simple twist of a physical knob, directly grasped between two fingers. Indeed, the ergonomic problems encountered here are in essence no different to those associated with a graphics simulation of a visual shuttle facility for sound editing. It is more usual, therefore, to encounter the alternative methodology used by Protools in this context, where supplementary screen windows are temporarily superimposed on the main display whenever adjustments are made to parameter settings for a particular channel, each variable being displayed as a movable slider rather than a rotary knob.

Such hierarchical arrangements, however, reintroduce the problems previously described in connection with the general use of miniature control display panels for hardware synthesizers in that once adjustments have been made and the supplementary screens closed, important parameter information may be hidden from the user. Moreover, an additional procedure has been introduced into the processes of control access and parameter adjustment. Although this situation may be tolerable for establishing initial parameter settings, the scope for interactive realtime modifications, if indeed the software is able to support such adjustments during a mixing operation, is now significantly impeded. It is indeed when programs such as Protools are used in performance mode that the earlier-mentioned drawbacks of a WIMPs control environment become all too clear. While it is feasible to display upwards of twenty-four audio channels on a high-resolution PC monitor, one cannot replicate the dexterity of two hands moving swiftly and accurately over an array of physical faders to make dynamic and possibly also simultaneous adjustments to the volume levels of individual tracks during a performance. The software will usually allow faders to be grouped such that the movement of one fader using the mouse and pointer will cause those others linked to it to move in tandem, but even a seemingly simple and very common operation such as cross-fading between two channels becomes all but impossible.

The inability to perform parallel control operations in real time highlights perhaps the most serious drawback which arises when graphic interfaces replace conventional control surfaces in the context of audio synthesis and signal processing. There are nevertheless special technologies now becoming available which may provide at least partial solutions to this particular problem, in particular contact-sensitive touch-screens which can detect the position and movement of individual fingers when in contact with the display surface itself. Exploration of such new technologies for the interactive control of audio synthesis and signal processing is still very much at the formative stage at the time of writing, and progress will be slow until such time as the necessary hardware enhancements to conventional graphic display screens become standard features of multimedia systems. Such improvements, however, may still not be sufficient to persuade many professional audio engineers to sacrifice the conventional mixer control interface in its entirety for a flat screen display, no matter how sophisticated the supporting functionality.

Digidesign's recent introduction (1998) of Procontrol, a 'hands-on' physical front end control surface for Protools, is especially significant in this context, since it would appear to recognise the functional limitations of an all-graphics mode of control and the mutual advantages of combining aspects of both technologies. At the most basic design level, architects of realtime WIMPs control interfaces repeatedly choose to ignore the simple fact that our ability to manipulate a number of typewriter keys both rapidly and simultaneously has considerable potential as a parallel function control mechanism in its own right. Such recognition of our physical skills does not in any way diminish the value of a linked graphics display facility to verify the changes as they occur, or indeed the importance of the mouse and pointer as an independent means of configuring the display and also selective data entry. What is clearly demanded, however, is a thorough reappraisal of the resources which are currently available for the control and monitoring of musical functions, and the development of strategies which will more appropriately bridge the gap between our physical, visual and aural communication skills and the functional characteristics of those music systems with which we wish to interact.

The term 'multimedia' thus needs to take on much broader terms of reference than are often currently used in this context, for it must not only include the resources used in sound production but also the design of the physical tools we use to control the resources themselves. The ultimate challenge is no longer the technology itself, but how we choose to apply it when combining so many different dimensions and possible modes of representation in a common goal, furthering the art and practice of music, not least in the context of electroacoustic music composition and performance. In such a fast-moving era of technical development, there is the constant promise of innovative and elegant solutions just around the corner, with the inference of immediate obsolescence and redundancy of all that has gone before. Experience has shown, however, that there are sometimes important lessons to be learnt from the past in seeking such objectives, and the field of interactive graphical control interfaces for music systems is no exception.

ACKNOWLEDGEMENTS

Figures 1–5 are reproduced with the authorisation of William Buxton, Department of Computer Science, University of Toronto (Buxton 1978). Figure 6 is reproduced with the authorisation of IRCAM, ©IRCAM – Centre Georges Pompidou – image from AudioScript.

REFERENCES

Buxton, W. 1978. *Design Issues in the Foundation of a Computer-Based Tool for Music Composition*. Technical Report CSRG-97. University of Toronto.

Buxton, W., Fogels, E. A., Fedorkow, G., and Sasaki, L. 1978. An introduction to the SSSP digital synthesizer. *Computer Music Journal* **2**(4): 28–38.

Buxton, W., Sniderman, R., Reeves, W., Patel, S., and Baeker, R. 1979. The evolution of the SSSP score editing tools. *Computer Music Journal* **3**(4): 14–25.

Chowning, J. 1985. The synthesis of complex audio spectra by means of frequency modulation. *Foundations of Computer Music*, pp. 1–29. Cambridge, MA: MIT Press.

Kaegi, W., and Tempelaars, S. 1978. VOSIM – a new sound synthesis system. *Journal of the Audio Engineering Society* **26**: 418–24.

Krefeld, V. 1990. The Hand in the Web: an interview with Michel Waisviçz. *Computer Music Journal* **14**(2): 28–33.

Mathews, M. V. 1969. *The Technology of Computer Music*. Cambridge, MA: MIT Press.

Mathews, M. V. 1991. The Radio Baton and Conductor Program, or: Pitch, the most important and least expressive part of music. *Computer Music Journal* **15**(4): 37–46.

Schaeffer, P. 1952. *A la recherche d'une musique concrète*. Paris: Seuil.

Schaeffer, P. 1966. *Traité des objets musicaux, essai interdisciplines*. Paris: Seuil.

Smalley, D. 1997. Spectromorphology: explaining sound-shapes. *Organised Sound* **2**(2): 107–26.

Vercoe, B. 1986. *CSOUND: A Manual for the Audio Processing System and Supporting Programs.* Cambridge, MA: MIT Media Lab.

Wishart, T. 1996. *On Sonic Art*, S. Emmerson (ed.). Amsterdam: Harwood Academic Publishers.